

ACTIONSCRIPT® 3.0-componenten gebruiken

Juridische kennisgeving

Zie http://help.adobe.com/nl_NL/legalnotices/index.html voor de juridische kennisgeving.

Inhoud

Hoofdstuk 1: Inleiding

Beoogd publiek	1
Systeemvereisten	1
Informatie over de documentatie	2
Typografische conventies	2
Termen die in deze handleiding worden gebruikt	2
Aanvullende bronnen	2

Hoofdstuk 2: Informatie over ActionScript 3.0-componenten

Voordelen van het gebruik van componenten	4
Componenttypen	6
Toevoegen aan en verwijderen uit een document	8
De versie van de component zoeken	9
Gebeurtenisafhandelingmodel van ActionScript 3.0	10
Een eenvoudige toepassing	11

Hoofdstuk 3: Werken met componenten

Componentarchitectuur	18
Werken met componentbestanden	20
Fouten opsporen in componenttoepassingen	22
Parameters en eigenschappen instellen	23
Het deelvenster Bibliotheek	24
Grootte van componenten wijzigen	25
Live voorvertoning	25
Gebeurtenissen afhandelen	26
Werken met het weergaveoverzicht	27
Werken met FocusManager	30
Werken met op List gebaseerde componenten	31
Werken met een DataProvider	32
Werken met een CellRenderer	40
Componenten toegankelijk maken	47

Hoofdstuk 4: UI-componenten gebruiken

De component Button gebruiken	48
De component CheckBox gebruiken	51
De component ColorPicker gebruiken	54
De component ComboBox gebruiken	56
De component DataGrid gebruiken	59
De component Label gebruiken	65
De component List gebruiken	67
De component NumericStepper gebruiken	72
De component ProgressBar gebruiken	75
De component RadioButton gebruiken	80

Inhoud

De component ScrollPane gebruiken	83
De component Slider gebruiken	86
De component TextArea gebruiken	89
De component TextInput gebruiken	92
De component TileList gebruiken	95
De component ULoader gebruiken	98
De component UScrollBar gebruiken	100

Hoofdstuk 5: UI-componenten aanpassen

UI-component aanpassen	103
Stijlen instellen	103
Skins	106
De component Button aanpassen	109
De component CheckBox aanpassen	111
De component ColorPicker aanpassen	112
De component ComboBox aanpassen	114
De component DataGrid aanpassen	116
De component Label aanpassen	121
De component List aanpassen	122
De component NumericStepper aanpassen	125
De component ProgressBar aanpassen	126
De component RadioButton aanpassen	128
De component ScrollPane aanpassen	130
De component Slider aanpassen	131
De component TextArea aanpassen	132
De component TextInput aanpassen	134
De component TileList aanpassen	136
De component ULoader aanpassen	138
De component UScrollBar aanpassen	138

Hoofdstuk 6: De component FLVPlayback gebruiken

Gebruik de component FLVPlayback	141
De component FLVPlayback aanpassen	160
Een SMIL-bestand gebruiken	172

Hoofdstuk 7: De component FLVPlaybackCaptioning gebruiken

De component FLVPlaybackCaptioning gebruiken	180
Ondertiteling met getimedede tekst gebruiken	182
Actiepunten met ondertiteling gebruiken	188
Meerdere FLV-bestanden met ondertiteling afspelen	190
De component FLVPlaybackCaptioning aanpassen	191

Hoofdstuk 1: Inleiding

Adobe® Flash® CS5 Professional is het professionele ontwerpgereedschap voor het maken van geavanceerde webtoepassingen. Componenten vormen de bouwstenen voor de Rich Internet Applications (RIA's) die deze toepassingen bevatten. Een *component* is een filmclip met parameters waarmee u de component tijdens het ontwerpen in Flash of bij uitvoering met methoden, eigenschappen en gebeurtenissen van Adobe® ActionScript® kunt aanpassen. Componenten zijn ontworpen zodat ontwikkelaars code opnieuw kunnen gebruiken en met elkaar kunnen delen en om complexe functionaliteit in te kapselen die ontwerpers zonder ActionScript kunnen gebruiken en aanpassen.

Met componenten kunt u gemakkelijk en snel krachtige toepassingen met een consistent uiterlijk en gedrag maken. In deze handleiding wordt beschreven hoe u toepassingen met Adobe ActionScript 3.0-componenten kunt maken. In de *Naslaggids voor Adobe® ActionScript® en ActionScript 3.0-componenten* wordt de API (programmeerinterface voor toepassingen) van elke component beschreven.

U kunt componenten gebruiken die door Adobe® zijn gemaakt, componenten downloaden die door andere ontwikkelaars zijn gemaakt of uw eigen componenten maken.

Beoogd publiek

Deze handleiding is bedoeld voor ontwikkelaars die Flash-toepassingen maken en componenten willen gebruiken die het ontwikkelen versnellen. Het is belangrijk dat u al bekend bent met het ontwikkelen van toepassingen in Flash en met het schrijven van ActionScript.

Als u minder ervaring hebt met het schrijven van ActionScript, kunt u componenten aan een document toevoegen, de parameters ervan in Eigenschapcontrole of Componentcontrole instellen en het deelvenster Gedragingen gebruiken om de gebeurtenissen ervan te verwerken. U kunt bijvoorbeeld een gedrag 'Ga naar webpagina' aan een component Button koppelen, waardoor een URL in een webbrowser wordt geopend wanneer op de knop wordt geklikt, zonder dat u ActionScript-code hoeft te schrijven.

Wanneer u een programmeur bent die krachtigere toepassingen wilt maken, kunt u componenten dynamisch maken, ActionScript gebruiken om eigenschappen in te stellen en methoden bij uitvoering aan te roepen en het gebeurtenislistenermodel gebruiken om gebeurtenissen te verwerken.

Zie "[Werken met componenten](#)" op pagina 18 voor meer informatie.

Systeemvereisten

Flash-componenten hebben dezelfde systeemvereisten als Flash zelf.

Elk SWF-bestand waarin componenten van Flash CS3 of hoger worden gebruikt, moet worden weergegeven in Adobe® Flash® Player 9.0.28.0 of hoger en moet worden gepubliceerd voor ActionScript 3.0 (u kunt dit op het tabblad Flash instellen via Bestand > Publicatie-instellingen).

Informatie over de documentatie

In dit document worden de details beschreven van het gebruik van componenten om Flash-toepassingen te ontwikkelen. Het is wel vereist dat u algemene kennis hebt van Flash en ActionScript 3.0. Specifieke documentatie over Flash en verwante producten is apart verkrijgbaar.

Dit document is als een PDF-bestand en als online Help beschikbaar. Start Flash en selecteer Help > Flash Help > Adobe ActionScript 3.0-componenten gebruiken om de online Help weer te geven.

Raadpleeg de volgende documenten voor informatie over Flash:

- *Flash gebruiken*
- *ActionScript 3.0-ontwikkelaarsgids*
- *Naslaggids voor ActionScript® 3.0 voor Adobe® Flash® Professional CS5*

Typografische conventies

In deze handleiding worden de volgende typografische conventies gebruikt:

- *Cursieve tekst geeft een waarde aan die moet worden vervangen (bijvoorbeeld in een mappad).*
- Broncode geeft ActionScript-code aan, waaronder methode- en eigenschapnamen.
- *Cursieve broncode geeft een code-item aan dat moet worden vervangen (bijvoorbeeld een ActionScript-parameter).*
- **Vetgedrukte tekst geeft een waarde aan die u invoert.**

Termen die in deze handleiding worden gebruikt

De volgende termen worden in deze handleiding gebruikt:

bij uitvoering Wanneer de code in Flash Player wordt uitgevoerd.

tijdens het ontwerpen Wanneer u in de Flash-ontwerpomgeving werkt.

Aanvullende bronnen

Naast de inhoud van deze handleidingen, biedt Adobe regelmatig bijgewerkte artikelen, ontwerpideeën en voorbeelden in Adobe Developer Center en Adobe Design Center.

Extra componentvoorbeelden zijn te vinden op www.adobe.com/go/learn_fl_samples.

Adobe Developer Center

Adobe Developer Center is uw bron voor de nieuwste informatie over ActionScript, artikelen over de ontwikkeling van toepassingen en informatie over belangrijke nieuwe kwesties. Bekijk het Developer Center op www.adobe.com/go/flash_devcenter.

Adobe Design Center

Leer de nieuwste feiten over digitaal ontwerp en bewegende afbeeldingen. Blader door het werk van de beste kunstenaars, ontdek nieuwe ontwerptrends en verbeter uw vaardigheden aan de hand van zelfstudies, hoofdwerkschema's en geavanceerde technieken. Controleer twee keer per maand voor nieuwe zelfstudies en artikelen en voor galeriestukken die inspireren. Bekijk het Design Center op www.adobe.com/go/fl_designcenter.

Hoofdstuk 2: Informatie over ActionScript 3.0-componenten

Componenten van Adobe® Flash® Professional CS5 zijn filmclips met parameters waarmee u het uiterlijk en gedrag van de filmclips kunt aanpassen. Een component kan een eenvoudig besturingselement voor de gebruikersinterface zijn, zoals een RadioButton of een CheckBox, maar een component kan ook inhoud bevatten, zoals een List of DataGrid.

Met componenten kunt u snel en gemakkelijk krachtige Flash-toepassingen met een consistent gedrag en uiterlijk bouwen. U kunt in plaats van aangepaste knoppen, keuzelijsten en lijsten ook Flash-componenten gebruiken die deze besturingselementen implementeren. U hoeft ze alleen maar van het deelvenster Componenten naar uw toepassingsdocument te slepen. U kunt het grafische uiterlijk van deze componenten ook gemakkelijk aanpassen zodat ze in het ontwerp van uw toepassing passen.

Dit is allemaal mogelijk zonder diepgaande kennis van ActionScript, maar u kunt ActionScript 3.0 ook gebruiken om het gedrag van een component aan te passen of nieuw gedrag te implementeren. Elke component heeft een unieke verzameling methoden, eigenschappen en gebeurtenissen van ActionScript die de *programmeerinterface voor toepassingen* (API) vormen. Met de API kunt u componenten maken en manipuleren terwijl de toepassing wordt uitgevoerd.

Met de API kunt u ook nieuwe, aangepaste componenten maken. U kunt componenten die door leden van de Flash-gemeenschap zijn gebouwd downloaden van Adobe Exchange op www.adobe.com/go/flash_exchange_nl. Ga naar www.adobe.com/go/learn_fl_creating_components_nl voor informatie over het maken van componenten.

De componentarchitectuur van ActionScript 3.0 omvat klassen waarop alle componenten zijn gebaseerd, skins en stijlen waarmee u het uiterlijk kunt aanpassen, een gebeurtenisafhandelingsmodel, focusbeheer, een toegankelijkheidsinterface en meer.

Opmerking: Adobe Flash CS5 omvat ActionScript 2.0-componenten en ActionScript 3.0-componenten. U kunt deze twee verzamelingen componenten niet door elkaar gebruiken. U moet de ene of de andere verzameling gebruiken voor een bepaalde toepassing. Flash CS5 biedt ActionScript 2.0-componenten of ActionScript 3.0-componenten, afhankelijk van het bestand dat u opent (ActionScript 2.0 of ActionScript 3.0). Wanneer u een nieuw Flash -document maakt, moet u Flash-bestand (ActionScript 3.0) of Flash-bestand (ActionScript 2.0) opgeven. Wanneer u een bestaand document opent, controleert Flash de publicatie-instellingen om te bepalen welke verzameling componenten moet worden gebruikt. Zie Adobe® ActionScript® 2.0-componenten gebruiken voor meer informatie over ActionScript 2.0-componenten.

Zie “[Componenttypen](#)” op pagina 6 voor een complete lijst met Flash ActionScript 3.0-componenten.

Voordelen van het gebruik van componenten

Met componenten kunt u het ontwerpen van uw toepassing scheiden van het schrijven van code. Ontwikkelaars kunnen zo functionaliteit maken die ontwerpers in toepassingen kunnen gebruiken. Ontwikkelaars kunnen veelgebruikte functionaliteit onderbrengen in componenten en ontwerpers kunnen de grootte, de locatie en het gedrag van componenten aanpassen door de parameters te wijzigen. Ze kunnen ook het uiterlijk van een component wijzigen door de grafische elementen of skins te bewerken.

Componenten delen kernfunctionaliteit zoals stijlen, skins en focusbeheer. Wanneer u de eerste component aan een toepassing toevoegt, vergt deze kernfunctionaliteit ongeveer 20 kilobytes. Wanneer u andere componenten toevoegt, wordt de eerste geheugentoe wijzing gedeeld door de toegevoegde componenten, waardoor uw toepassing minder snel groeit.

In deze sectie worden enkele voordelen van ActionScript 3.0-componenten beschreven.

De kracht van ActionScript 3.0 biedt een robuuste, objectgeoriënteerde programmeertaal die een belangrijke stap vormt in de ontwikkeling van de mogelijkheden van de Flash Player. De taal is ontwikkeld voor het bouwen van RIA's (Rich Internet Applications) op basis van een herbruikbare code. ActionScript 3.0 is gebaseerd op ECMAScript, de internationale gestandaardiseerde scripttaal. ActionScript 3.0 is compatibel met de taalspecificaties van ECMAScript (ECMA-262), editie 3. Zie de *ActionScript 3.0-ontwikkelaarsgids* voor uitgebreide informatie over ActionScript 3.0. Zie de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor verwijzingsinformatie over de taal.

De op FLA gebaseerde UI-componenten bieden eenvoudig toegang tot skins om deze eenvoudig te kunnen aanpassen tijdens het ontwerpen. Deze componenten bieden ook stijlen, waaronder skinstijlen, waarmee u aspecten van het uiterlijk van componenten kunt aanpassen en skins bij uitvoering kunt laden. Zie "[UI-componenten aanpassen](#)" op pagina 103 en de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

Nieuw component FVLPlayback voegt FLVPlaybackCaptioning toe samen met ondersteuning voor volledig scherm, verbeterde live voorvertoning, skins waarmee u kleur- en alpha-instellingen kunt toevoegen en verbeterde download- en lay-outfuncties voor FLV-bestanden.

Met Eigenschapscontrole en Componentcontrole kunt u de componentparameters wijzigen wanneer u in Flash ontwerpt. Zie "[Werken met componentbestanden](#)" op pagina 20 en "[Parameters en eigenschappen instellen](#)" op pagina 23 voor meer informatie.

In het nieuwe verzameldialoogvenster voor de componenten ComboBox, List en TileList kunt u de eigenschap `dataProvider` voor deze componenten via de gebruikersinterface vullen. Zie "[Een DataProvider maken](#)" op pagina 32 voor meer informatie.

Met het ActionScript-gebeurtenismodel kan uw toepassing luisteren naar gebeurtenissen en reageren door gebeurtenishandlers aan te roepen. Zie "[Gebeurtenisafhandelingsmodel van ActionScript 3.0](#)" op pagina 10 en "[Gebeurtenissen afhandelen](#)" op pagina 26 voor meer informatie.

Managerklassen bieden een gemakkelijke manier voor focus- en stijlbeheer in een toepassing. Zie de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

De basisklasse UIComponent biedt componenten kernmethodes, eigenschappen en gebeurtenissen waardoor ze worden uitgebreid. Alle componenten van de ActionScript 3.0-gebruikersinterface overerven elementen van de klasse UIComponent. Zie de klasse UIComponent in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

Het gebruik van een SWC in de UI-componenten op basis van FLA-bestanden biedt ActionScript-definitie als een element binnen de tijdlijn van een component om het compileren te versnellen.

Met een eenvoudig uitbreidbare klassenhiërarchie op basis van ActionScript 3.0 kunt u unieke naamruimten maken, klassen importeren waar nodig en eenvoudig subklassen maken waarmee componenten kunnen worden uitgebreid.

Zie de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

Opmerking: *Flash CS5 ondersteunt FLA-componenten en SWC-componenten. Zie "[Componentarchitectuur](#)" op pagina 18 voor meer informatie.*

Componenttypen

De Flash-componenten worden geïnstalleerd wanneer u Flash CS5 installeert.

ActionScript 3.0-componenten omvatten de volgende componenten voor de gebruikersinterface (UI):

Button	List	TextArea
CheckBox	NumericStepper	TextInput
ColorPicker	RadioButton	TileList
ComboBox	ProgressBar	UILoader
DataGrid	ScrollPane	UIScrollBar
Label	Slider	

Naast de UI-componenten omvatten de ActionScript 3.0-componenten van Flash de volgende componenten en ondersteunende klassen:

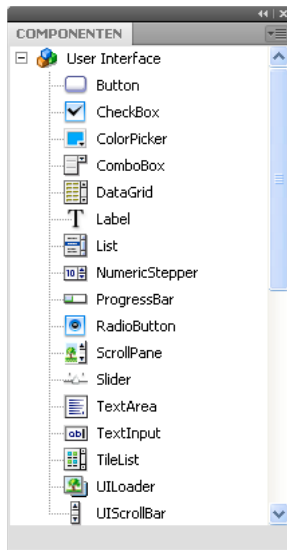
- De component FLVPlayback (fl.video.FLVPlayback), een op een SWC-bestand gebaseerde component.
Met de component FLVPlayback kunt u een videospeler opnemen in uw Flash-toepassing die progressieve streaming video via HTTP kan afspelen, van een Adobe® Flash® Video Streaming Service (FVSS) of van een Macromedia® Flash® Media Server (FMS) van Adobe. Zie “[De component FLVPlayback gebruiken](#)” op pagina 141 voor meer informatie.
- De aangepaste UI-componenten voor het afspelen van FLV, die op FLA-bestanden zijn gebaseerd en met zowel de ActionScript 2.0- als ActionScript 3.0-versie van de component FLVPlayback werken. Zie “[De component FLVPlayback gebruiken](#)” op pagina 141 voor meer informatie.
- De component FLVPlaybackCaptioning, die ondertiteling voor FLVPlayback biedt. Zie “[De component FLVPlaybackCaptioning gebruiken](#)” op pagina 180.

Zie de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor een uitvoerige lijst met ActionScript 3.0-componenten en de ondersteunende klassen.

Flash-componenten weergeven:

U kunt de Flash ActionScript 3.0-componenten in het deelvenster Componenten weergeven door deze stappen te volgen.

- 1 Start Flash.
- 2 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0) of open een bestaand Flash-document waarin de publicatie-instellingen ActionScript 3.0 opgeven.
- 3 Selecteer Venster > Componenten om het deelvenster Componenten te openen, als dat nog niet het geval is.



deelvenster Componenten met gebruikersinterfacecomponenten

U kunt ook extra componenten downloaden van Adobe Exchange op www.adobe.com/go/flash_exchange_nl. U kunt componenten die u van Adobe Exchange hebt gedownload installeren door Adobe® Extension Manager te downloaden van www.adobe.com/go/exchange_nl en deze toepassing te installeren. Klik op de koppeling Adobe Exchange home en zoek naar de koppeling Extension Manager.

Elke component kan in het deelvenster Componenten in Flash worden weergegeven. Volg deze stappen om componenten op een Windows®- of Macintosh®-computer te installeren.

Componenten op een Windows- of Mac-computer installeren:

- 1 Sluit Flash af.
- 2 Plaats het SWC- of FLA-bestand met de component in de volgende map op uw vaste schijf:
 - Windows:
C:\Program Files\Adobe\Adobe Flash CS5\taal\Configuration\Components
 - Macintosh:
Macintosh HD/Programma's/Adobe Flash CS5/Configuration/Components
- 3 Start Flash.
- 4 Selecteer Venster > Componenten om de component in het deelvenster Componenten weer te geven, als dat nog niet het geval is.

Zie “[Werken met componentbestanden](#)” op pagina 20 voor meer informatie over componentbestanden.

Toevoegen aan en verwijderen uit een document

Wanneer u een op een FLA-bestand gebaseerde component van het deelvenster Componenten naar het werkgebied sleept, importeert Flash een bewerkbare filmclip in de bibliotheek. Wanneer u een op een SWC-bestand gebaseerde component naar het werkgebied sleept, importeert Flash een gecompileerde clip in de bibliotheek. Nadat een component in de bibliotheek is geïmporteerd, kunt u instanties ervan naar het werkgebied slepen vanuit het deelvenster Bibliotheek of vanuit het deelvenster Componenten.

Componenten toevoegen tijdens het ontwerpen

U kunt een component aan een document toevoegen door deze vanuit het deelvenster Componenten te slepen. U kunt eigenschappen instellen voor elke instantie van een component in Eigenschapcontrole of op het tabblad Parameters in Componentcontrole.

- 1 Selecteer Venster > Componenten.
- 2 Dubbelklik op de component in het deelvenster Componenten of sleep de component naar het werkgebied.
- 3 Selecteer de component in het werkgebied.
- 4 Wanneer Eigenschapcontrole niet zichtbaar is, selecteert u Venster > Eigenschappen > Eigenschappen.
- 5 Voer in Eigenschapcontrole een instantienaam in voor de componentinstantie.
- 6 Selecteer Venster > Eigenschapcontrole en Component inspector en selecteer het tabblad Parameters tab om parameters voor de instantie op te geven.
Zie “[Parameters en eigenschappen instellen](#)” op pagina 23 voor meer informatie.
- 7 Wijzig de grootte van de component naar wens door de waarden voor de breedte (W) en hoogte (H) te bewerken.
Zie “[UI-componenten aanpassen](#)” op pagina 103 voor meer informatie over het instellen van de grootte van specifieke componenttypen.
- 8 Selecteer Besturing > Film testen of druk op Ctrl+Enter om het document te compileren en de resultaten van uw instellingen weer te geven.

U kunt ook de kleur en tekstopmaak van een component wijzigen door de stijleigenschappen van de component in te stellen of het uiterlijk aan te passen door de skins van de component te bewerken. Zie “[UI-componenten aanpassen](#)” op pagina 103 voor meer informatie over deze onderwerpen.

Wanneer u tijdens het ontwerpen een component naar het werkgebied sleept, kunt u naar de component verwijzen door de instantienaam te gebruiken (bijvoorbeeld `myButton`).

Componenten bij uitvoering toevoegen met componenten ActionScript

Wanneer u met ActionScript een component bij uitvoering aan een document wilt toevoegen, moet de component in de bibliotheek van de toepassing staan (Venster > Bibliotheek) wanneer het SWF-bestand wordt gecompileerd.

Wanneer u een component aan de bibliotheek wilt toevoegen, sleept u de component van het deelvenster Componenten naar het deelvenster Bibliotheek. Zie “[Het deelvenster Bibliotheek](#)” op pagina 24 voor meer informatie over de bibliotheek.

U moet ook het klassebestand van de component importeren zodat de API ervan beschikbaar is voor uw toepassing. Klassebestanden van componenten zijn in *pakketten* geïnstalleerd die één of meer klassen bevatten. Wanneer u een componentklasse wilt importeren, gebruikt u de aanwijzing `import` en geeft u de pakketnaam en klassenaam op. U kunt bijvoorbeeld de klasse `Button` importeren met de volgende aanwijzing `import`:

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

Zie de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) als u wilt weten in welk pakket een component is opgenomen. Zie “[Werken met componentbestanden](#)” op pagina 20 voor meer informatie over de locatie van componentbronbestanden.

Wanneer u een instantie van de component wilt maken, moet u de ActionScript-constructormethode van de component aanroepen. Met de volgende instructie maakt u bijvoorbeeld een instantie van een Button met de naam `aButton`:

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

In de laatste stap wordt de statische methode `addChild()` aangeroepen om de componentinstantie aan het werkgebied of de toepassingscontainer toe te voegen. In de volgende instructie wordt bijvoorbeeld de instantie `aButton` toegevoegd:

```
addChild(aButton);
```

U kunt nu de API van de component gebruiken om dynamisch de grootte en positie in het werkgebied op te geven, naar gebeurtenissen te luisteren en eigenschappen in te stellen om het gedrag aan te passen. Zie de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie over de API voor een bepaalde component.

Voor meer informatie over de methode `addChild()` raadpleegt u “[Werken met het weergaveoverzicht](#)” op pagina 27.

Een component verwijderen

U kunt een componentinstantie uit het werkgebied verwijderen tijdens het ontwerpen door deze te selecteren en op Delete te drukken. De instantie wordt uit het werkgebied verwijderd, maar de component wordt niet uit uw toepassing verwijderd.

U kunt een component uit uw Flash-document verwijderen nadat u deze in het werkgebied of de bibliotheek hebt geplaatst. Verwijder de component en de bijbehorende elementen uit de bibliotheek. Het is niet voldoende om de component uit het werkgebied te verwijderen. Wanneer u de component niet uit de bibliotheek verwijdert, wordt deze tijdens het compileren in uw toepassing opgenomen.

- 1 Selecteer in het deelvenster Bibliotheek het symbool voor de component.
- 2 Klik op de knop Verwijderen onder in het deelvenster Bibliotheek of selecteer Verwijderen in het optiemenu van het deelvenster Bibliotheek.

Herhaal deze stappen om alle elementen die aan de component zijn gekoppeld te verwijderen.

Voor meer informatie over hoe u een component uit de container verwijdert terwijl de toepassing actief is, raadpleegt u “[Een component verwijderen uit het weergaveoverzicht](#)” op pagina 29.

De versie van de component zoeken

ActionScript 3.0-componenten van Flash hebben een versie-eigenschap die u kunt weergeven wanneer u deze aan Technische ondersteuning van Adobe moet doorgeven of wanneer u wilt weten welke versie van de component u gebruikt.

U geeft als volgt het versienummer voor een gebruikersinterfacecomponent weer:

- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep de component naar het werkgebied en geef de component een instantienaam. Sleep bijvoorbeeld een ComboBox naar het werkgebied en noem deze `aCb`.

- 3 Druk op **F9** of selecteer Venster > Handelingen om het deelvenster Handelingen te openen.
- 4 Klik op frame 1 van de hoofdtijdlijn en voeg de volgende code toe in het deelvenster Handelingen:

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

Het versienummer, vergelijkbaar met het versienummer in de volgende afbeelding, wordt in het deelvenster Uitvoer weergegeven.

Voor de componenten FLVPlayback en FLVPlaybackCaptioning moet u verwijzen naar de klassenaam in plaats van de instantienaam omdat het versienummer is opgeslagen in een klasseconstante.

U geeft als volgt het versienummer voor de componenten FLVPlayback en FLVPlaybackCaptioning weer:

- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep de componenten FLVPlayback en FLVPlaybackCaptioning naar het deelvenster Bibliotheek.
- 3 Druk op **F9** of selecteer Venster > Handelingen om het deelvenster Handelingen te openen.
- 4 Klik op frame 1 van de hoofdtijdlijn en voeg de volgende code toe in het deelvenster Handelingen.

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

De versienummers verschijnen in het deelvenster Output.

Gebeurtenisafhandelingsmodel van ActionScript 3.0

ActionScript 3.0 introduceert een enkel gebeurtenisafhandelingsmodel ter vervanging van de verschillende gebeurtenisafhandelingsmechanismen die in vorige versies van ActionScript bestonden. Het nieuwe gebeurtenismodel is gebaseerd op de Document Object Model (DOM) Level 3 Events Specification.

Voor ervaren ontwikkelaars die de methode `addListener()` in ActionScript 2.0 gebruiken, kan het handig zijn de verschillen aan te geven tussen het gebeurtenislistenermodel van ActionScript 2.0 en het gebeurtenismodel van ActionScript 3.0. In de volgende lijst worden enkele belangrijke verschillen tussen de twee gebeurtenismodellen beschreven:

- U kunt gebeurtenislisteners in ActionScript 2.0 in sommige gevallen met `addListener()` en in andere gevallen met `addEventListener()` toevoegen, terwijl u in ActionScript 3.0 in alle gevallen `addEventListener()` gebruikt.
- ActionScript 2.0 kent geen gebeurtenisstroom, dus de methode `addListener()` kan alleen worden aangeroepen op het object dat de gebeurtenis uitzendt, terwijl in ActionScript 3.0 de methode `addEventListener()` op elk object dat deel uitmaakt van de gebeurtenisstroom kan worden aangeroepen.
- In ActionScript 2.0 kunnen gebeurtenislisteners functies, methoden of objecten zijn, terwijl in ActionScript 3.0 alleen functies of methoden gebeurtenislisteners kunnen zijn.
- De syntaxis `on` (*gebeurtenis*) wordt niet meer ondersteund in ActionScript 3.0. Het is dus niet mogelijk een ActionScript-gebeurteniscode aan een filmclip te koppelen. U kunt `addEventListener()` alleen gebruiken om een gebeurtenislistener toe te voegen.

In het volgende voorbeeld, waarin naar een gebeurtenis `MouseEvent.CLICK` op een component `Button` met de naam `aButton` wordt geluisterd, wordt het basismodel voor gebeurtenisafhandeling van ActionScript 3.0 getoond:

```
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);  
}
```

Raadpleeg *ActionScript 3.0 programmeren* voor meer informatie over gebeurtenisafhandeling van ActionScript 3.0. Zie “[Gebeurtenissen afhandelen](#)” op pagina 26 voor meer informatie over gebeurtenisafhandeling van ActionScript 3.0 voor componenten.

Een eenvoudige toepassing

In deze sectie staan de stappen waarmee u een eenvoudige ActionScript 3.0-toepassing kunt maken met Flash-componenten en het Flash-ontwerpgereedschap. Het voorbeeld wordt gegeven als een FLA-bestand met ActionScript-code op de tijdlijn en ook als een extern ActionScript-klassebestand met een FLA-bestand dat alleen de componenten in de bibliotheek bevat. Over het algemeen kunt u het beste grotere toepassingen ontwikkelen met externe klassebestanden zodat u code tussen klassen en toepassingen kunt delen en uw toepassingen eenvoudiger zijn te onderhouden. Raadpleeg *ActionScript 3.0 programmeren* voor meer informatie over programmeren in ActionScript 3.0.

Ontwerp van de toepassing

Ons eerste voorbeeld van een toepassing met ActionScript-componenten is een variatie op de standaardtoepassing ‘Hallo wereld’ en dus is het ontwerp vrij eenvoudig:

- De toepassing wordt Greetings genoemd.
- Er wordt een TextArea gebruikt om een begroeting weer te geven die in eerste instantie Hallo wereld is.
- Er wordt een ColorPicker gebruikt waarmee de kleur van de tekst kan worden gewijzigd.
- Er worden drie RadioButtons gebruikt waarmee u de grootte van de tekst kunt instellen op klein, groter of grootst.
- Er wordt een ComboBox gebruikt waarmee u een andere begroeting uit een vervolgkeuzelijst kunt selecteren.
- De toepassing gebruikt componenten uit het deelvenster Componenten en maakt ook toepassingselementen via ActionScript-code.

Nu deze definitie is vastgelegd, kunt u beginnen met het bouwen van de toepassing.

De toepassing Greetings maken

Met de volgende stappen wordt de toepassing Greetings gemaakt met het Flash-ontwerpgereedschap. U maakt een FLA-bestand, plaatst componenten in het werkgebied en voegt ActionScript-code aan de tijdlijn toe.

De toepassing Greetings in een FLA-bestand maken:

- 1 Selecteer Bestand > Nieuw.
- 2 Selecteer in het dialoogvenster Nieuw document het type Flash-bestand (ActionScript 3.0) en klik op OK.
Er wordt een nieuw Flash-venster geopend.
- 3 Selecteer Bestand > Opslaan, geef het Flash-bestand de naam **Greetings.fla** en klik op Opslaan.
- 4 Selecteer in het deelvenster Componenten van Flash een component TextArea en sleep deze naar het werkgebied.

- 5 Selecteer de TextArea in het werkgebied en typ in het venster Eigenschappen **aTa** voor de instantienaam en voer de volgende informatie in:
 - Voer **230** in voor de waarde B (breedte).
 - Voer **44** in voor de waarde H (hoogte).
 - Voer **165** in voor de waarde X (horizontale positie).
 - Voer **57** in voor de waarde Y (verticale positie).
 - Voer **Hallo wereld!** in voor de parameter text op het tabblad Parameters.
- 6 Sleep een component ColorPicker naar het werkgebied, plaats het links van de TextArea en geef de component de instantienaam **txtCp**. Voer de volgende informatie in Eigenschapcontrole in:
 - Voer **96** in voor de waarde X.
 - Voer **72** in voor de waarde Y.
- 7 Sleep drie componenten RadioButton naar het werkgebied en geef ze de instantienamen **smallRb**, **largerRb** en **largestRb**. Voer in Eigenschapcontrole de volgende informatie voor de componenten in:
 - Voer **100** voor de waarde B en **22** voor de waarde H in.
 - Voer **155** in voor de waarde X.
 - Voer **120** voor de waarde Y voor smallRb, **148** voor largerRb en **175** voor largestRb in.
 - Voer **fontRbGrp** voor de parameter groupName van elke component in.
 - Voer labels in voor elke waarde op het tabblad Parameters voor **Klein**, **Groter**, **Grootst**.
- 8 Sleep een ComboBox naar het werkgebied en geef deze de instantienaam **msgCb**. Voer in Eigenschapcontrole de volgende informatie in:
 - Voer **130** in voor de waarde B.
 - Voer **265** in voor de waarde X.
 - Voer **120** in voor de waarde Y.
 - Voer op het tabblad Parameters **Begroetingen** in voor de parameter prompt.
 - Dubbelklik op het tekstveld voor de parameter dataProvider om het dialoogvenster Waarden te openen.
 - Klik op het plusteken en vervang de labelwaarde door **Hallo wereld!**
 - Herhaal de vorige stap om de labelwaarden **Prettige dag!** en **Goedemorgen!** toe te voegen.
 - Klik op OK om het dialoogvenster Waarden te sluiten.
- 9 Sla het bestand op.
- 10 Open het deelvenster Handelingen, indien dit nog niet het geval is, met **F9** of door Handelingen in het menu Venster te selecteren. Klik op frame 1 van de hoofdlijn en voer de volgende code in het deelvenster Handelingen in:

```
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);
```


Met de eerste drie regels worden de gebeurtenisklassen die de toepassing gebruikt geïmporteerd. Een gebeurtenis vindt plaats wanneer de gebruiker een van de componenten gebruikt. In de volgende vijf regels worden gebeurtenishandlers geregistreerd voor de gebeurtenissen waarnaar de toepassing wil luisteren. Er vindt een gebeurtenis `click` voor een `RadioButton` plaats wanneer de gebruiker erop klikt. De gebeurtenis `change` vindt plaats wanneer de gebruiker een andere kleur in de `ColorPicker` selecteert. De gebeurtenis `change` vindt plaats op de `ComboBox` wanneer de gebruiker een andere begroeting in de vervolgkeuzelijst kiest.

In de vierde regel wordt de klasse `RadioButtonsGroup` geïmporteerd zodat de toepassing een gebeurtenislistener kan toewijzen aan de groep `RadioButtons` in plaats van de listener aan elke knop afzonderlijk toe te wijzen.

- 11 Voeg de volgende coderegel toe aan het deelvenster Handelingen om het `TextFormat`-object `tf` te maken dat door de toepassing wordt gebruikt om de stijleigenschappen `size` en `color` van de tekst in de `TextArea` te wijzigen.

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

- 12 Voeg de volgende code toe om de gebeurtenisafhandelingsfunctie `rbHandler` te maken. Deze functie handelt de gebeurtenis `click` af wanneer de gebruiker op een van de componenten `RadioButton` klikt.

```
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);
}
```

Deze functie gebruikt een instructie `switch` om de eigenschap `target` van het object `event` te controleren om te bepalen welke `RadioButton` de gebeurtenis heeft geactiveerd. De eigenschap `currentTarget` bevat de naam van het object dat de gebeurtenis heeft geactiveerd. Afhankelijk van de `RadioButton` waarop de gebruiker heeft geklikt, wijzigt de toepassing de tekengrootte van de tekst in de `TextArea` in 14, 18 of 24 punten.

- 13 Voeg de volgende code toe om de functie `cpHandler()` die een wijziging in de waarde in de `ColorPicker` afhandelt, te implementeren:

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

Deze functie stelt de eigenschap `color` van het `TextFormat`-object `tf` in op de kleur die in de `ColorPicker` is geselecteerd en roept vervolgens `setStyle()` aan om dit toe te passen op de tekst in de `TextArea`-instantie `aTa`.

- 14 Voeg de volgende code toe om de functie `cbHandler()` die een wijziging in de selectie in de `ComboBox` afhandelt, te implementeren:

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

Deze functie vervangt de tekst in de `TextArea` door de geselecteerde tekst in de `ComboBox`, `event.target.selectedItem.label`.

15 Selecteer Besturing > Film testen of druk op Ctrl+Enter om de code te compileren en de toepassing Greetings te testen.

In de volgende sectie wordt getoond hoe u dezelfde toepassing kunt bouwen met een externe ActionScript-klasse en een FLA-bestand met alleen de vereiste componenten in de bibliotheek.

De toepassing Greetings2 maken met een extern klassebestand:

1 Selecteer Bestand > Nieuw.

2 Selecteer in het dialoogvenster Nieuw document het type Flash-bestand (ActionScript 3.0) en klik op OK.

Er wordt een nieuw Flash-venster geopend.

3 Selecteer Bestand > Opslaan, geef het Flash-bestand de naam **Greetings2.fla** en klik op Opslaan.

4 Sleep de volgende componenten van het deelvenster Componenten naar de bibliotheek:

- ColorPicker
- ComboBox
- RadioButton
- TextArea

Het gecompileerde SWF-bestand gebruikt al deze elementen, dus u moet ze aan de bibliotheek toevoegen. Sleep de componenten naar de onderzijde van het deelvenster Bibliotheek. Wanneer u deze componenten aan de bibliotheek toevoegt, worden andere elementen (zoals List, TextInput en UIScrollBox) automatisch toegevoegd.

5 Typ in het deelvenster Eigenschappen bij Documentklasse **Greetings2**.

Wanneer Flash een waarschuwing geeft met de tekst 'Kan geen definitie voor de documentklasse vinden', kunt u deze negeren. U definieert de klasse Greetings2 in de volgende stappen. Deze klasse definieert de hoofdfunctionaliteit voor de toepassing.

6 Sla het bestand Greetings2.fla op.

7 Selecteer Bestand > Nieuw.

8 Selecteer ActionScript-bestand in het dialoogvenster Nieuw document en klik op OK.

Er wordt een nieuw Script-venster geopend.

9 Voeg de volgende code aan het Script-venster toe:

```
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() {
```

In het script wordt een ActionScript 3.0-klasse met de naam `Greetings2` gedefinieerd. Het script doet het volgende:

- Er worden klassen geïmporteerd die we in het bestand gaan gebruiken. Meestal voegt u deze `import` instructies toe wanneer u in de code naar verschillende klassen verwijst, maar om het kort te houden worden ze in dit voorbeeld in één stap geïmporteerd.
- Er worden variabelen gedeclareerd. Deze staan voor verschillende soorten componentobjecten die we aan de code toevoegen. Een andere variabele maakt het `TextFormat`-object `tf`.
- Er wordt een constructorfunctie met de naam `Greetings2()` voor de klasse gedefinieerd. We voegen regels aan deze functie toe en voegen vervolgens in de volgende stappen andere methoden aan de klasse toe.

10 Selecteer Bestand > Opslaan, geef het bestand de naam **Greetings2.as** en klik op Opslaan.

11 Voeg de volgende regels code toe aan de functie `Greetings2()`.

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

De functie moet er nu als volgt uitzien:

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

12 Voeg de volgende coderegels toe na de accolade sluiten van de methode `Greeting2()`:

```
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);
}
```

Deze regels doen het volgende:

- Maken een instantie van de componenten die in de toepassing worden gebruikt.
- Stellen de grootte, positie en eigenschappen van elke component in.

- Voeg elke component aan het werkgebied toe met de methode `addChild()`.

13 Voeg na de accolade sluiten van de methode `bldRadioButtons()` de volgende code voor de methode `setUpHandlers()` toe:

```
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;
}
}
```

Deze functies definiëren gebeurtenislisteners voor de componenten.

14 Selecteer Bestand > Opslaan om het bestand op te slaan.

15 Selecteer Besturing > Film testen of druk op Ctrl+Enter om de code te compileren en de toepassing Greetings2 te testen.

Opeenvolgende voorbeelden ontwikkelen en uitvoeren

Nu u de toepassing Greetings hebt ontwikkeld en uitgevoerd, beschikt u over de basiskennis die u nodig hebt om de andere codevoorbeelden in dit boek uit te voeren. De betreffende ActionScript 3.0-code in elk voorbeeld wordt gemarkeerd en behandeld. U kunt de voorbeelden in dit boek kopiëren en plakken in een FLA-bestand en dit compileren en uitvoeren.

Hoofdstuk 3: Werken met componenten

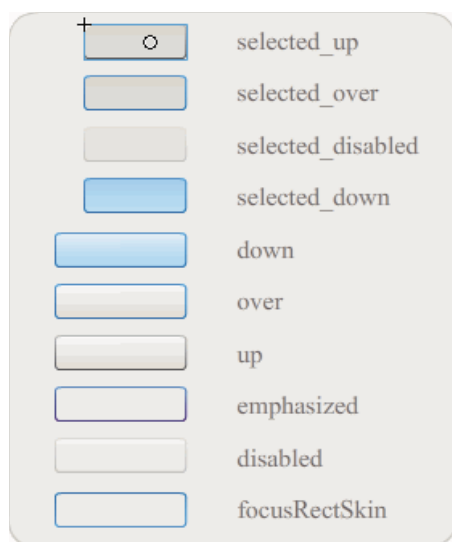
Componentarchitectuur

Adobe® ActionScript® 3.0-componenten worden ondersteund door Adobe® Flash Player versie 9.0.28.0 en hoger. Deze componenten zijn niet compatibel met componenten die met versies vóór Flash CS4 zijn gebouwd. Voor informatie over het gebruik van Adobe® ActionScript® 2.0-componenten, raadpleegt u *Adobe® ActionScript® 2.0-componenten gebruiken* en de *Naslaggids voor Adobe® ActionScript® 2.0 en ActionScript 2.0-componenten*.

De gebruikersinterfacecomponenten (UI-componenten) van Adobe ActionScript 3.0 worden als op FLA gebaseerde componenten geïmplementeerd, maar Flash CS5 ondersteunt zowel op SWC als op FLA gebaseerde componenten. De componenten FLVPlayback en FLVPlaybackCaptioning zijn bijvoorbeeld op SWC gebaseerde componenten. U kunt elk type component in de map Components plaatsen zodat het in het deelvenster Components wordt weergegeven. Deze twee typen componenten worden anders gebouwd, waardoor ze hier afzonderlijk worden beschreven:

Op FLA gebaseerde componenten van ActionScript 3.0

De UI-componenten van ActionScript 3.0 zijn op FLA gebaseerde bestanden (.fla) met ingebouwde skins die u kunt openen voor bewerken door te dubbelklikken op de component in het werkgebied. De skins van de component en andere elementen worden op Frame 2 van de tijdlijn geplaatst. Wanneer u dubbelklikt op de component, springt Flash automatisch naar Frame 2 en wordt een palet geopend van de skins van de component. In de volgende afbeelding wordt het palet van skins getoond die voor de component Button worden weergegeven.



Skins voor de component Button

Zie “[UI-componenten aanpassen](#)” op pagina 103 en “[De component FLVPlayback aanpassen](#)” op pagina 160 voor informatie over componentskins en het aanpassen van componenten.

De op FLA gebaseerde UI-componenten van Flash CS5 bevatten ook een SWC-bestand dat de reeds gecompileerde ActionScript-code van de component bevat, zodat compilatie van uw toepassingen sneller plaatsvindt en conflicten met uw ActionScript 3.0-instellingen worden vermeden. Het SWC-bestand ComponentShim wordt in elk gebruikersinterfacecomponent op Frame 2 van het werkgebied geplaatst om vooraf gecompileerde definities beschikbaar te maken. Deze definities zijn beschikbaar voor ActionScript wanneer de component zich in het werkgebied bevindt of in de bibliotheek met de optie Exporteren in eerste frame ingeschakeld in Koppelingseigenschappen. Wanneer u een component met ActionScript wilt maken, moet u de klasse ook met een aanwijzing `import` importeren om deze te kunnen benaderen. Voor informatie over de instructie `import` gaat u naar de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#).

Op SWC gebaseerde componenten

Op SWC gebaseerde componenten bevatten ook een FLA-bestand en een ActionScript-klassebestand, maar ze zijn als een SWC-bestand gecompileerd en geëxporteerd. Een SWC-bestand is een pakket van vooraf gecompileerde Flash-symbolen en ActionScript-code waarmee u het opnieuw compileren van symbolen en code die niet wijzigt, kunt voorkomen.

De componenten FLVPlayback en FLVPlaybackCaptioning zijn gebaseerd op SWC. In plaats van ingebouwde skins bevatten ze externe skins. De component FLVPlayback bevat een standaardskin die u kunt wijzigen door een skin uit de verzameling vooraf ontworpen skins te selecteren, door besturingselementen van de UI-besturingselementen aan te passen in het deelvenster Componenten (BackButton, BufferingBar, enzovoort) of door een aangepaste skin te maken. Zie “[De component FLVPlayback aanpassen](#)” op pagina 160 voor meer informatie.

In Flash kunt u als volgt een filmclip naar een gecompileerde clip omzetten:

Een filmclip compileren

- Klik met de rechtermuisknop (Windows) of houd de Control-toets ingedrukt en klik (Macintosh) op de filmclip in het deelvenster Bibliotheek en selecteer Omzetten in gecompileerde Clip.

De gecompileerde clip gedraagt zich net als de filmclip waarvan het is gecompileerd, maar gecompileerde clips worden veel sneller weergegeven en gepubliceerd dan gewone filmclips. Gecompileerde clips kunnen niet worden bewerkt, maar de eigenschappen ervan kunnen in Eigenschapcontrole en in Componentcontrole worden weergegeven.

SWC-componenten bevatten een gecompileerde clip, de vooraf gecompileerde ActionScript-definities van de component en andere bestanden die de component beschrijven. Wanneer u uw eigen component maakt, kunt u het als een SWC-bestand exporteren om het te verspreiden.

SWC-bestand exporteren

- Selecteer de filmclip in het deelvenster Bibliotheek, klik met de rechtermuisknop (Windows) of houd de Control-toets ingedrukt en selecteer (Macintosh) SWC-bestand exporteren.

Opmerking: De indeling van een Flash CS4 of later SWC-bestand is compatibel met de Flex SWC-indeling, zodat SWC-bestanden tussen twee producten kunnen worden uitgewisseld. Dit is niet noodzakelijkerwijs zonder wijzigingen.

Ga naar www.adobe.com/go/learn_fl_creating_components voor informatie over componentskins en het aanpassen van componenten.

De API voor ActionScript 3.0-componenten

Elke ActionScript 3.0-component is op een ActionScript 3.0-klasse gebouwd die zich in een pakketmap bevindt en een naam heeft met de indeling `fl.pakketnaam.klassenaam`. De component Button is bijvoorbeeld een instantie van de klasse Button en heeft de pakketnaam `fl.controls.Button`. Wanneer u een componentklasse in uw toepassing importeert, moet u naar de pakketnaam verwijzen. U kunt de klasse Button met de volgende instructie importeren:

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

Zie “[Werken met componentbestanden](#)” op pagina 20 voor meer informatie over het werken met componentbestanden.

De klasse van een component definieert de methoden, eigenschappen, gebeurtenissen en stijlen waarmee u in uw toepassing kunt werken. De ActionScript 3.0 UI-componenten zijn subclasses van de klassen Sprite en UIComponent en overerven eigenschappen, methoden en gebeurtenissen hiervan. De klasse Sprite is de basisbouwsteen en lijkt op een filmclip, maar heeft geen tijdlijn. De klasse UIComponent is de basisklasse voor alle visuele componenten, zowel interactieve als niet-interactieve. In de *Adobe Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS* worden het overervingspad, de eigenschappen, methoden gebeurtenissen en stijlen van elke component beschreven.

Alle ActionScript 3.0-componenten gebruiken het model voor afhandeling van gebeurtenissen van ActionScript 3.0. Raadpleeg “[Gebeurtenissen afhandelen](#)” op pagina 26 en *ActionScript 3.0 programmeren* voor meer informatie over het afhandelen van gebeurtenissen.

Werken met componentbestanden

In deze sectie wordt uitgelegd waar componentbestanden worden opgeslagen, waar u de ActionScript-bronbestanden kunt vinden en hoe u componenten kunt toevoegen aan of verwijderen uit het deelvenster Componenten.

Locatie van componentbestanden

Flash-componenten worden in de configuratiemap op toepassingsniveau opgeslagen.

Opmerking: Raadpleeg ‘*Configuratiemappen geïnstalleerd met Flash*’ in *Flash gebruiken voor informatie over deze mappen*.

Componenten worden op de volgende locatie opgeslagen:

- Windows 2000 of Windows XP: C:\Program Files\Adobe\Adobe Flash CS5\taal\Configuration\Components
- Mac OS X: Macintosh HD/Programma's/Adobe Flash CS5/Configuration/Components

De gebruikersinterfacecomponenten (UI) binnen de map Components bevinden zich in het bestand User Interface.fla en de componenten FLVPlayback (FLVPlaybackAS3.swc) en FLVPlaybackCaptioning staan in de map Video.

U kunt componenten ook op de volgende op gebruiker gebaseerde locaties opslaan:

- Windows 2000 of Windows XP: C:\Documents and Settings\gebruiker\Local Settings\Application Data\Adobe\Adobe Flash CS5\nl\Configuration\Components
- Windows Vista: C:\Gebruikers\gebruiker\Local Settings\Application Data\Adobe\Adobe Flash CS5\nl\Configuration\Components

Opmerking: In Windows is de map Application Data standaard verborgen. U geeft verborgen mappen en bestanden als volgt weer: selecteer Deze computer om Windows Verkenner te open en selecteer vervolgens Extra>Mapopties en het tabblad Weergave. Selecteer op het tabblad Weergave het keuzerondje Verborgen bestanden en mappen weergeven.

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

Locatie bronbestanden van component

De ActionScript-klassebestanden (.as) (of *bronbestanden*) van componenten zijn voor Windows 2000 of Windows XP in de volgende toepassingsmappen geïnstalleerd:

gebruikersinterfacecomponenten C:\Program Files\Adobe\Adobe Flash CS5\nl\Configuration\Component Source\ActionScript 3.0\User Interface\fl

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

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

Voor Mac OS X bevinden de bronbestanden van de component op de volgende locatie:

gebruikersinterfacecomponenten Macintosh HD/Programma's/Adobe Flash CS5/Configuration/Component Source/ActionScript 3.0/User Interface/fl

FLVPlayback Macintosh HD/Programma's/Adobe Flash CS5/Configuration/Component Source/ActionScript 3.0/FLVPlayback/fl/video

FLVPlaybackCaptioning Macintosh HD/Programma's/Adobe Flash CS5/Configuration/Component Source/ActionScript 3.0/FLVPlaybackCaptioning/fl/video

Bronbestanden van componenten en Classpath

Aangezien de code van ActionScript 3.0-componenten in de componenten zelf is gecompileerd, moet u de locatie van de ActionScript-klassebestanden niet in uw variabele Classpath opgeven. Als u de locatie wel in de variabele Classpath opneemt, duurt het langer om uw toepassingen te compileren. Wanneer Flash echter klassenbestanden van de component in uw klasepad vindt, heeft het klassebestand altijd voorrang op gecompileerde code van de component.

Wanneer u foutopsporing op een toepassing met componenten toepast, kunt u de locatie van de bronbestanden van de component aan uw klasepad toevoegen. Zie “[Fouten opsporen in componenttoepassingen](#)” op pagina 22 voor meer informatie.

Componentbestanden wijzigen

Wanneer u SWC gebaseerde componenten bijwerkt, toevoegt of verwijdert of nieuwe FLA gebaseerde componenten aan Flash toevoegt, moet u ze opnieuw laden in het deelvenster Componenten om ze beschikbaar te maken. U kunt de componenten opnieuw laden door Flash opnieuw te starten of door Opnieuw laden te selecteren in het deelvenster Componenten. Hierdoor vangt Flash componenten op die u aan de map Components hebt toegevoegd.


Wanneer Flash wordt uitgevoerd, kunt u als volgt componenten in het deelvenster Componenten opnieuw laden:

- Selecteer Opnieuw laden in het menu van deelvenster Componenten.

Een component uit het deelvenster Componenten verwijderen:

- Verwijder het FLA-, SWC- of MXP-bestand uit de map Components en start Flash opnieuw of selecteer Opnieuw laden in het menu van het deelvenster Componenten. Een MXP-bestand is een componentbestand dat van de Adobe Exchange is gedownload.

Wanneer Flash wordt uitgevoerd, kunt u op SWC gebaseerde componenten verwijderen en vervangen. De wijzigingen worden weergegeven wanneer u de componenten opnieuw laadt. Maar wanneer u op FLA gebaseerde componenten wijzigt of verwijdert, worden de wijzigingen pas aangeduid wanneer u Flash beëindigt en opnieuw start. U kunt echter op FLA gebaseerde componenten toevoegen en ze laden met de opdracht Opnieuw laden.

 *Het wordt door Adobe aangeraden dat u eerst een kopie maakt van het Flash-componentbestand (.fla of .as) dat u wilt wijzigen. Vervolgens kunt u het, indien nodig, opnieuw opslaan.*

Fouten opsporen in componenttoepassingen

De ActionScript 3.0-componenten bevatten alle broncode om de compilatietijd te verminderen wanneer u uw toepassing compileert. De Flash-foutopsporing kan echter geen code binnenin gecompileerde clips controleren. Wanneer u daarom in de broncode van de component foutopsporing op uw toepassing wilt toepassen, moet u de bronbestanden van de component toevoegen aan uw klassepadijnstelling.

De locatie van de pakketmappen van de component is relatief ten opzichte van de locatie van de bronbestanden van het componenttype. Voeg de volgende locatie voor de gebruikersinterfacepakketten toe aan uw klassepadijnstelling om alle ActionScript 3.0-bronbestanden voor alle UI-componenten te verwijzen:

- \$(AppConfig)/Component Source/ActionScript 3.0/User Interface

Opmerking: Hierdoor wordt de gecompileerde code voor alle UI-componenten overschreven en wordt de compilatietijd van uw toepassing vergroot. Wanneer u voor wat voor reden dan ook het bronbestand van een component hebt gewijzigd, kan deze als gevolg ander gedrag vertonen.

Selecteer Voorkeuren in het menu Bewerken, selecteer vervolgens ActionScript in de lijst met categorieën en klik op de knop Instellingen ActionScript 3.0 om het klassepadijnstelling in te stellen. Klik op de plusknop bovenin het venster waar in de huidige instelling wordt weergegeven om een nieuwe invoer toe te voegen.

De variabele \$(AppConfig) verwijst naar de configuratiemap van Flash CS5 op de locatie waar u Flash CS5 hebt geïnstalleerd. Normaalgesproken ziet het pad er als volgt uit:

- Windows 2000 of Windows XP: C:\Program Files\Adobe\Adobe Flash CS5\taal\Configuration\
- Mac OS X: Macintosh HD/Programma's/Adobe Flash CS5/Configuration/

Opmerking: Als u het bronbestand van een component moet wijzigen, wordt het door Adobe sterk aanbevolen dat u het originele bronbestand naar een andere locatie kopieert en deze locatie aan uw klassepadijnstelling toevoegt.

Zie “[Locatie bronbestanden van component](#)” op pagina 21 voor meer informatie over de locatie van bronbestanden van componenten.

Parameters en eigenschappen instellen

Elke component bevat parameters die u kunt instellen om het uiterlijk en het gedrag ervan te wijzigen. Een parameter is een eigenschap van de klasse van de component en wordt in Eigenschapcontrole en in Componentcontrole weergegeven. De eigenschappen die het meest worden gebruikt, worden als ontwerpparameters weergegeven; andere eigenschappen moet u met ActionScript instellen. Alle parameters die tijdens het ontwerpen kunnen worden ingesteld, kunnen ook met ActionScript worden ingesteld. Wanneer u een parameter met ActionScript instelt, worden alle waarden die tijdens het ontwerpen zijn ingesteld, overschreven.

De meeste gebruikersinterfacecomponenten van ActionScript 3.0 overerven de eigenschappen en methoden van de klasse `UIComponent` en van een basisklasse. De klassen `Button` en `CheckBox` overerven bijvoorbeeld eigenschappen van zowel de klasse `UIComponent` als de klasse `BaseButton`. De overerfde eigenschappen van een component en de klasse-eigenschappen ervan zijn voor u beschikbaar. De component `ProgressBar` overerft bijvoorbeeld de eigenschap `ProgressBar.enabled` van `UIComponent`, maar bevat ook een eigen eigenschap `ProgressBar.percentComplete`. U hebt toegang tot beide eigenschappen om met een instantie van de component `ProgressBar` te communiceren. Zie voor meer informatie over de eigenschappen van een component het item over de klasse van de component in de [Naslaggids voor ActionScript 3.0](#).

U kunt parameters voor een componentinstantie instellen met Eigenschapcontrole of Componentcontrole.

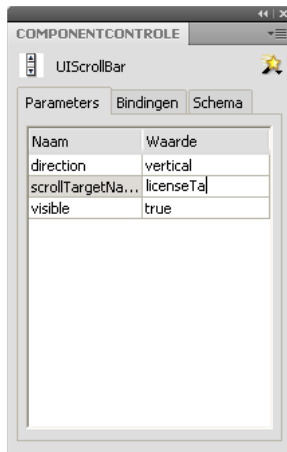
Een instantienaam invoeren voor een component in Eigenschapcontrole:

- 1 Selecteer Venster > Eigenschappen > Eigenschappen.
- 2 Selecteer een instantie van een component in het werkgebied.
- 3 Voer een naam in voor de componentinstantie in het vak waarin <instantienaam > wordt aangeduid, die onder de vervolgkeuzelijst met filmclip staat. Of klik op het tabblad Parameters en voer de naam in het vak in onder het woord *Component*. Voer de waarden van de parameters in die u wilt instellen.

Het is raadzaam een voorvoegsel aan de instantienaam toe te voegen om aan te geven wat voor soort component het is; hierdoor kunt u de ActionScript-code gemakkelijker lezen. In dit voorbeeld de instantienaam **licenseSb**, omdat de component een schuifbalk is die een licentieovereenkomst in het tekstgebied **licenseTa** schuift.

Een componentinstantie invoeren voor een component in Eigenschapcontrole:

- 1 Selecteer Venster > Componentcontrole.
- 2 Selecteer een instantie van een component in het werkgebied.
- 3 Klik op het tabblad Parameters en voer de waarden voor een van de weergegeven parameters in.



Componentparameters in Componentcontrole

Componenteigenschappen instellen in ActionScript

In ActionScript gebruikt u een puntoperator (.) (puntnotatie) om eigenschappen of methoden te benaderen die horen bij een object of instantie in het werkgebied. Een expressie in puntnotatie begint met de naam van de instantie, wordt gevolgd door een punt en eindigt met het element dat u wilt opgeven. Met de volgende ActionScript-code wordt bijvoorbeeld de eigenschap `width` van de `CheckBox`-instantie `aCh` ingesteld op een breedte van 50 pixels:

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

Met de volgende `if`-instructie wordt gecontroleerd of de gebruiker het selectievakje heeft ingeschakeld:

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

Het deelvenster Bibliotheek

Wanneer u een component toevoegt aan een document, wordt de component door Flash als een filmclip geïmporteerd in het deelvenster Bibliotheek. U kunt een component ook vanuit het deelvenster Componenten rechtstreeks naar het deelvenster Bibliotheek slepen en vervolgens een instantie van de component toevoegen aan het werkgebied. In elk geval moet u een component aan de bibliotheek toevoegen voordat u de klasse-elementen van de component kunt openen.

Als u een component aan de bibliotheek wilt toevoegen en hiervan met ActionScript een instantie wilt maken, moet u de klasse van de component eerst importeren met de aanwijzing `import`. In de aanwijzing `import` moet u zowel de pakketnaam als de klassenaam van de component opgeven. Met de volgende instructie wordt bijvoorbeeld de klasse `Button` geïmporteerd:

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

Wanneer u een component in de bibliotheek opneemt, wordt ook een map met de elementen van de component, met de skins voor de verschillende toestanden, geïmporteerd. De *skins* van een component omvatten de verzameling symbolen voor de grafische weergave van de componenten in de toepassing. Eén skin is de grafische weergave, of filmclip, waarmee een bepaalde toestand van de component wordt aangeduid.

Met de inhoud van de map `Component Assets` kunt u de skins van de component desgewenst wijzigen. Zie “[UI-componenten aanpassen](#)” op pagina 103 voor meer informatie.

Wanneer een component zich in de bibliotheek bevindt, kunt u meer instanties van de component aan uw document toevoegen door het pictogram vanuit het deelvenster Componenten of Bibliotheek naar het werkgebied te slepen.

Grootte van componenten wijzigen

Met het gereedschap Vrije transformatie of de methode `setSize()` kunt u de grootte van componentinstanties wijzigen. U kunt de methode `setSize()` vanuit elke componentinstantie aanroepen (zie `UIComponent.setSize()`) om de grootte te wijzigen. Met de volgende code wordt de grootte van een instantie van de component `List` ingesteld op een breedte van 200 pixels en een hoogte van 300 pixels:

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

De grootte van een component wordt niet automatisch aangepast aan het label. Als een componentinstantie die aan een document is toegevoegd niet groot genoeg is om het label weer te geven, wordt de labeltekst afgekapt. U moet de grootte van de component aanpassen aan het label.

Voor meer informatie over het aanpassen van het formaat van een component gaat u naar de desbetreffende component in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#).

Live voorvertoning

Met de functie Live voorvertoning, die standaard is ingeschakeld, kunt u componenten in het werkgebied bekijken zoals ze worden weergegeven in de gepubliceerde Flash-inhoud. De grootte van de componenten komt min of meer overeen.

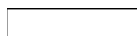
Live voorvertoning in- of uitschakelen

- Selecteer Besturing > Live voorvertoning inschakelen. Naast de optie wordt een vinkje weergegeven om aan te geven dat deze is ingeschakeld.

Voor de live voorvertoning worden verschillende parameters gebruikt voor verschillende componenten. Als u wilt weten welke componentparameters worden weergegeven in de live voorbeeldweergave, gaat u naar de desbetreffende component in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#).



Een component Button met Live voorvertoning ingeschakeld



Een component Button met Live voorvertoning uitgeschakeld

Componenten in Live voorvertoning zijn niet functioneel. Als u de functionaliteit wilt testen, moet u de opdracht Besturing > Film testen selecteren.

Gebeurtenissen afhandelen

Elke component zendt gebeurtenissen uit wanneer de gebruiker ermee werkt. Wanneer de gebruiker op een knop klikt, verzendt deze de gebeurtenis `MouseEvent.CLICK` en wanneer de gebruiker een item in een lijst selecteert, verzendt deze de gebeurtenis `Event.CHANGE`. Er kan ook een gebeurtenis plaatsvinden wanneer er iets belangrijks gebeurt met een component, bijvoorbeeld wanneer inhoud is geladen voor een `UILoader`-instantie. In dat geval wordt de gebeurtenis `Event.COMPLETE` gegenereerd. U handelt een gebeurtenis af door ActionScript-code te schrijven die wordt uitgevoerd wanneer de gebeurtenis plaatsvindt.

Tot de gebeurtenissen van een component behoren de gebeurtenissen van een klasse waarvan de component gebeurtenissen overerft. Dit betekent dat alle ActionScript 3.0-gebruikersinterfacecomponenten gebeurtenissen van de klasse `UIComponent` overerven omdat dit de basisklasse is voor de ActionScript 3.0-gebruikersinterfacecomponenten. Als u de lijst met gebeurtenissen wilt bekijken die door een component worden gebroadcast, gaat u naar de sectie met gebeurtenissen voor de klasse die bij de component hoort in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#).

Zie *ActionScript 3.0 programmeren* voor een volledige uitleg over het afhandelen van gebeurtenissen in ActionScript 3.0.

Gebeurtenislisteners

Bij het afhandelen van gebeurtenissen voor ActionScript 3.0-componenten moet u rekening houden met de volgende belangrijke punten:

- Alle gebeurtenissen worden uitgezonden door een instantie van een componentklasse. De componentinstantie is de *broadcaster*.
- U registreert een *gebeurtenislistener* door de methode `addEventListener()` voor de componentinstantie aan te roepen. Met de volgende coderegel wordt bijvoorbeeld een listener voor de gebeurtenis `MouseEvent.CLICK` toegevoegd aan de `Button`-instantie `aButton`:

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

Met de tweede parameter van de methode `addEventListener()` wordt de naam van de functie, `clickHandler`, geregistreerd om te worden aangeroepen wanneer de gebeurtenis plaatsvindt. Deze functie wordt ook wel een *callback-functie* genoemd.

- U kunt meerdere listeners registreren voor één componentinstantie.
- ```
aButton.addEventListener(MouseEvent.CLICK, clickHandler1);
aButton.addEventListener(MouseEvent.CLICK, clickHandler2);
```
- U kunt één listener registreren voor meerdere componentinstanties.
- ```
aButton.addEventListener(MouseEvent.CLICK, clickHandler1);  
bButton.addEventListener(MouseEvent.CLICK, clickHandler1);
```
- Aan de gebeurtenishandlerfunctie wordt een gebeurtenisobject doorgegeven dat informatie bevat over het gebeurtenistype en de instantie die de gebeurtenis uitzendt. Zie “[Informatie over het gebeurtenisobject](#)” op pagina 27 voor meer informatie.
 - De listener blijft actief tot de toepassing wordt beëindigd of u de listener expliciet verwijdert met de methode `removeEventListener()`. Met de volgende coderegel wordt bijvoorbeeld de listener voor de gebeurtenis `MouseEvent.CLICK` op `aButton` verwijderd:

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

Informatie over het gebeurtenisobject

Het gebeurtenisobject overerft van de Event-objectklasse en heeft eigenschappen met informatie over de gebeurtenis die heeft plaatsgevonden, inclusief de eigenschappen `target` en `type`, die essentiële informatie over de gebeurtenis bevatten:

Eigenschap	Beschrijving
<code>type</code>	Een tekenreeks waarmee het type gebeurtenis wordt aangegeven.
<code>target</code>	Een verwijzing naar de componentinstantie die de gebeurtenis uitzendt.

Als een gebeurtenis extra eigenschappen heeft, worden deze opgenomen in de beschrijving van de klasse die bij de gebeurtenis hoort. Deze beschrijving vindt u in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#).

Wanneer er een gebeurtenis plaatsvindt, wordt het gebeurtenisobject automatisch gegenereerd en doorgegeven aan de gebeurtenishandlerfunctie.

U kunt het gebeurtenisobject in de functie gebruiken om de naam weer te geven van de gebeurtenis die is uitgezonden of de instantienaam weer te geven van de component die de gebeurtenis heeft uitgezonden. Via de instantienaam hebt u weer toegang tot andere componenteigenschappen. In de volgende code wordt bijvoorbeeld de eigenschap `target` van het gebeurtenisobject `evtObj` gebruikt om toegang te krijgen tot de eigenschap `label` van `aButton` en deze weer te geven in het deelvenster Uitvoer:

```
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");
}
```

Werken met het weergaveoverzicht

Alle ActionScript 3.0-componenten overerven van de klasse `DisplayObject` en hebben daarom toegang tot de methoden en eigenschappen van deze klasse om te werken met het weergaveoverzicht. Het *weergaveoverzicht* is de hiërarchie van weergegeven objecten en visuele elementen in een toepassing. Deze hiërarchie bevat de volgende elementen:

- Het werkgebied: de container op hoofdniveau
- Weergaveobjecten die onder andere vormen, filmclips en tekstvelden bevatten
- Containers voor weergaveobjecten: speciale typen weergaveobjecten die onderliggende weergaveobjecten kunnen bevatten.

De volgorde van objecten in het weergaveoverzicht bepaalt hun diepte in de bovenliggende container. De diepte van een object verwijst naar de positie (van boven naar beneden of van voren naar achteren) in het werkgebied of in de weergavecontainer. De dieptevolgorde is duidelijk wanneer objecten elkaar overlappen, maar deze bestaat ook wanneer dit niet het geval is. Voor elk object in het weergaveoverzicht is een diepte ingesteld in het werkgebied. Als u de diepte van een object wilt wijzigen om deze op de voorgrond of achter andere objecten weer te geven, moet u de positie in het weergaveoverzicht wijzigen. De standaardvolgorde van objecten in het weergaveoverzicht is de volgorde waarin ze zijn opgenomen in het werkgebied. Positie 0 in het weergaveoverzicht wordt ingenomen door het laatste object in de dieptevolgorde.

Een component toevoegen aan het weergaveoverzicht

U kunt een object aan een `DisplayObjectContainer`-object toevoegen door de methode `addChild()` of `addChildAt()` van de container aan te roepen. In het geval van het werkgebied kunt u tijdens de ontwerpfase ook een object aan het weergaveoverzicht toevoegen door er een te maken. Componenten kunt u vanuit het deelvenster Componenten naar het werkgebied slepen. Als u een object aan een container wilt toevoegen met `ActionScript`, maakt u eerst een instantie van het object door de constructor aan te roepen met de operator `new`. Vervolgens roept u de methode `addChild()` of `addChildAt()` aan om de instantie op te nemen in het werkgebied en het weergaveoverzicht. Met de methode `addChild()` wordt het object ingevoegd op de volgende positie in het weergaveoverzicht en met `addChildAt()` geeft u op op welke positie het object moet worden ingevoegd. Als u een positie opgeeft die al wordt ingenomen, worden het object op die positie en alle objecten daarboven een positie naar boven verplaatst. De eigenschap `numChildren` van een `DisplayObjectContainer`-object bevat informatie over het aantal weergaveobjecten in het object. U kunt een object uit het weergaveoverzicht ophalen door de methode `getChildAt()` aan te roepen en de positie op te geven. Wanneer u op de hoogte bent van de naam van het object, kunt u de methode `getChildByName()` aanroepen.

Opmerking: Wanneer u een component toevoegt met `ActionScript`, moet u een naam aan de naameigenschap toewijzen als u de component op naam wilt kunnen openen in het weergaveoverzicht.

In het volgende voorbeeld worden de namen en posities van drie componenten in het weergaveoverzicht weergegeven. Sleep eerst een `NumericStepper`-, `Button`- en `ComboBox`-instantie naar het werkgebied zodat ze elkaar overlappen en geef hieraan de instantienamen `aNs`, `aButton` en `aCb`. Voeg vervolgens in het deelvenster Handelingen de volgende code toe aan frame 1 van de tijdlijn:

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

De volgende regels worden weergegeven in het deelvenster Uitvoer:

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

Een component verplaatsen in het weergaveoverzicht

U kunt de positie en weergavediepte van een object in het weergaveoverzicht wijzigen door de methode `addChildAt()` aan te roepen en de naam en de gewenste positie van een object op te geven als de parameters van de methode. U kunt bijvoorbeeld de volgende code toevoegen aan het voorgaande voorbeeld om de `NumericStepper` op de voorgrond weer te geven en de lus te herhalen om de nieuwe posities van de componenten in het weergaveoverzicht weer te geven:


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

In het deelvenster Uitvoer wordt het volgende weergegeven:

```
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
```

De NumericStepper moet ook vóór de andere componenten worden weergegeven op het scherm.

`numChildren` verwijst naar het aantal objecten (van 1 tot n) in het weergaveoverzicht waarbij aan de eerste positie in het weergaveoverzicht het getal 0 wordt toegewezen. Wanneer het overzicht drie objecten bevat, is de indexpositie van het derde object dus 2. Dit betekent dat u naar de laatste positie of het bovenste object in het weergaveoverzicht kunt verwijzen op basis van de weergavediepte, bijvoorbeeld `numChildren - 1`.

Een component verwijderen uit het weergaveoverzicht

U kunt een component uit een weergaveobjectcontainer en het bijbehorende weergaveoverzicht verwijderen met de methoden `removeChild()` en `removeChildAt()`. In het volgende voorbeeld worden drie componenten Button op elkaar geplaatst in het werkgebied en wordt voor elke component een gebeurtenislistener toegevoegd. Wanneer u op een component Button klikt, wordt deze uit het weergaveoverzicht en het werkgebied verwijderd.

- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep een Button van het deelvenster Componenten naar het deelvenster Bibliotheek.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende code toe:

```
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);  
}
```

Zie 'Weergave programmeren' in *ActionScript 3.0 programmeren* voor een volledige uitleg over het weergaveoverzicht.

Werken met FocusManager

Wanneer de gebruiker op Tab drukt om te navigeren in een Flash-toepassing of in een toepassing klikt, bepaalt de klasse FocusManager welke component invoerfocus krijgt. U hoeft geen instantie FocusManager aan een toepassing toe te voegen of een code te schrijven om FocusManager te activeren, tenzij u een component maakt.

Als een RadioButton-object focus krijgt, worden dat object en alle objecten met dezelfde waarde voor `groupName` gecontroleerd door FocusManager en wordt focus ingesteld op het object waarvan de eigenschap `selected` is ingesteld op `true`.

Omdat elke modale venstercomponent een instantie van FocusManager bevat, fungeren de opties in een venster als de eigen tabverzameling. Zo wordt voorkomen dat de gebruiker per ongeluk met de toets Tab naar componenten in andere vensters navigeert.

FocusManager gebruikt het diepteniveau (of de z-volgorde) van elementen in de container als het standaardnavigatieschema (of de *tablus*). De gebruiker navigeert doorgaans door de tablus met Tab, waarbij de focus in stappen wordt verplaatst van de eerste component met focus naar de laatste en vervolgens weer terug naar de eerste. De diepteniveaus worden voornamelijk bepaald door de volgorde waarin componenten naar het werkgebied worden gesleept, maar u kunt ook de opdrachten Wijzigen > Rangschikken > Naar voorgrond/Naar achtergrond gebruiken om de uiteindelijke z-volgorde te bepalen. Zie [“Werken met het weergaveoverzicht”](#) op pagina 27 voor meer informatie over diepteniveaus.

U kunt de methode `setFocus()` aanroepen om een componentinstantie in een toepassing focus te geven. In het volgende voorbeeld wordt een instantie FocusManager gemaakt voor de huidige container (`this`) en wordt focus gegeven aan de Button-instantie `aButton`.

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

U kunt bepalen welke component focus heeft door de methode `getFocus()` aan te roepen en u kunt bepalen welke component in de tablus als volgende focus zal krijgen door de methode `getNextFocusManagerComponent()` aan te roepen. In het volgende voorbeeld bevat het werkgebied een CheckBox, een RadioButton en een Button en heeft elke component listeners voor gebeurtenissen `MouseEvent.CLICK` en `FocusEvent.MOUSE_FOCUS_CHANGE`. Wanneer de gebeurtenis `MouseEvent.CLICK` optreedt omdat de gebruiker op de component heeft geklikt, roept de functie `showFocus()` de methode `getNextFocusManagerComponent()` aan om te bepalen welke component in de tablus als volgende focus krijgt. Vervolgens roept deze de methode `setFocus()` aan om die componentinstantie focus te geven. Wanneer de gebeurtenis `FocusEvent.MOUSE_FOCUS_CHANGE` optreedt, geeft de functie `fc()` de naam van de component weer waarin deze gebeurtenis is opgetreden. Deze gebeurtenis wordt geactiveerd wanneer de gebruiker op een andere component dan de volgende in de tablus klikt.

```
// 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);
}
```

Als u een knop wilt maken die focus krijgt wanneer de gebruiker op Enter (Windows) of Return (Macintosh) drukt, stelt u de eigenschap `FocusManager.defaultButton` in op de instantie `Button` die u als standaardknop wilt gebruiken, zoals in de volgende code:

```
import fl.managers.FocusManager;

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

De klasse `FocusManager` overschrijft de standaardfocusrechthoek van Flash Player en tekent een aangepaste focusrechthoek met ronde hoeken.

Zie voor meer informatie over het maken van een focusschema in een Flash-toepassing de klasse `FocusManager` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#). Als u een aangepaste focusmanager wilt maken, moet u een klasse maken waarmee de interface `IFocusManager` wordt geïmplementeerd. Raadpleeg `IFocusManager` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#) voor meer informatie.

Werken met op List gebaseerde componenten

De componenten `List`, `DataGrid` en `TileList` overerven alle van de basisklasse `SelectableList`. Daarom worden deze componenten beschouwd als op `List` gebaseerde componenten. Omdat een `ComboBox` uit een tekstvak en een lijst bestaat, is ook dit een op `List` gebaseerde component.

Een component List bestaat uit rijen. Componenten DataGrid en TileList bestaan uit rijen die kunnen worden onderverdeeld in meerdere kolommen. Een cel is het snijpunt tussen een rij en een kolom. In een component List, die bestaat uit één kolom rijen, is elke rij een cel. De volgende twee belangrijke aspecten zijn van toepassing op een cel:

- De gegevenswaarden in cellen worden items genoemd. Een *item* is een ActionScript-object dat wordt gebruikt voor het opslaan van de informatie-eenheden in een lijst. Een List kan worden vergeleken met een array; elk geïndexeed onderdeel van de array is een item. In een List is een item een object dat doorgaans een eigenschap `label` bevat die wordt weergegeven en een eigenschap `data` die wordt gebruikt voor het opslaan van gegevens. Een *gegevensaanbieder* is een gegevensmodel van de items in een List. Met een gegevensaanbieder kunt u een op lijst gebaseerde component eenvoudig vullen door deze toe te wijzen aan de eigenschap `dataProvider` van de component.
- Een cel kan verschillende typen gegevens bevatten, variërend van tekst tot afbeeldingen, filmclips of klassen die u kunt maken. Daarom moet een cel worden getekend op een wijze die geschikt is voor de inhoud. Als gevolg daarvan hebben op lijst gebaseerde componenten een *celrenderer*. In het geval van de component DataGrid is elke kolom een DataGridColumn-object, dat ook de eigenschap `cellRenderer` heeft, zodat elke kolom op een voor de inhoud geschikte wijze kan worden gerenderd.

Alle op List gebaseerde componenten hebben de eigenschappen `cellRenderer` en `dataProvider`. Deze kunt u instellen om de cellen van deze componenten te laden en te renderen. Zie “[Werken met een DataProvider](#)” op pagina 32 and “[Werken met een CellRenderer](#)” op pagina 40 voor informatie over het gebruik van deze eigenschappen en het werken met op List gebaseerde componenten.

Werken met een DataProvider

Een DataProvider is een gegevensbron die u kunt gebruiken om gegevens op te geven in de componenten ComboBox, DataGrid, List en TileList. Elk van deze componentklassen heeft de eigenschap `dataProvider` waaraan u een DataProvider-object kunt toewijzen om de cellen van de component te vullen met gegevens. Doorgaans bestaat een gegevensaanbieder uit een verzameling gegevens, zoals een Array- of XML-object.

Een DataProvider maken

Voor de componenten ComboBox, List en TileList kunt u in de ontwerpomgeving een DataProvider maken met de parameter `dataProvider`. In Eigenschapcontrole is voor de component DataGrid de eigenschap `dataProvider` niet opgenomen omdat deze component meerdere kolommen kan bevatten en de gegevensaanbieder daarom complexer is. U kunt ook ActionScript gebruiken om een DataProvider voor deze componenten en voor de component DataGrid te maken.

De parameter dataProvider gebruiken

U kunt een eenvoudige gegevensaanbieder voor de componenten ComboBox, List en TileList maken door op de parameter `dataProvider` te klikken op het tabblad Parameters van Eigenschapcontrole of Componentcontrole.

Als u dubbelklikt op de waardecel, waarin eerst een lege array wordt weergegeven, opent u het dialoogvenster Waarden waarin u meerdere label- en gegevenswaarden kunt invoeren om de gegevensaanbieder te maken.



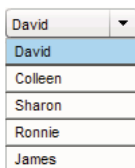
Dialogvenster Waarden voor dataProvider

Klik op het plusteken om een item toe te voegen aan de parameter dataProvider. Klik op het minteken om een item te verwijderen. Klik op de pijl-omhoog of pijl-omlaag om een geselecteerd item naar boven of beneden te verplaatsen in de lijst. In de volgende afbeelding wordt het dialogvenster Waarden weergegeven waarin een lijst met namen en verjaardagen van kinderen wordt gemaakt.



Dialogvenster Waarden met gegevens

De array die u maakt, bestaat uit paren label- en waardevelden. De labelvelden zijn `label` en `data`, en in de waardevelden worden de namen en verjaardagen van de kinderen weergegeven. In het labelveld wordt aangegeven welke inhoud wordt weergegeven in de lijst, in dit geval de namen van de kinderen. De resulterende ComboBox ziet er zo uit:



De ComboBox die is gevuld door de DataProvider

Wanneer u de gegevens hebt toegevoegd, klikt u op OK om het dialogvenster te sluiten. De array in de parameter dataProvider wordt nu gevuld met de items die u hebt gemaakt.

Werken met componenten

allowMultiple...	false
dataProvider	[{label:David,data:11/19/1995},{label:Colleen,data:4/20/1993},{label:Ronnie,data:9/6/1997}]
enabled	true
horizontalIn...	1
horizontalPa...	0
horizontalSc...	auto
verticalLineS...	1

Parameter dataProvider met gegevens

U kunt de gemaakte label- en gegevenswaarden openen door met ActionScript de eigenschap `dataProvider` van de component te openen.

Een DataProvider maken met ActionScript

U kunt een `DataProvider` maken door de gegevens in een Array- of XML-object te maken en het object op te geven als de parameter `value` voor de `DataProvider`-constructor.

Opmerking: In ActionScript 3.0 kunt u een Array- of XML-object niet rechtstreeks aan de eigenschap `dataProvider` toewijzen omdat de eigenschap wordt gedefinieerd als een `DataProvider`-object en alleen een object van het type `DataProvider` kan ontvangen.

In het volgende voorbeeld wordt een component `List`, één kolom met rijen, gevuld met de namen van verschillende kinderen en hun verjaardagen. In het voorbeeld wordt de lijst in de array `items` gedefinieerd en wordt deze bij het maken van de `DataProvider`-instantie (`new DataProvider(items)`) opgegeven als de parameter en wordt hieraan de eigenschap `dataProvider` van de component `List` toegewezen.

```
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);
```

De array bestaat uit paren label- en waardeelden. De labelvelden zijn `label` en `data`, en in de waardeelden worden de namen en verjaardagen van de kinderen weergegeven. In het labelveld wordt aangegeven welke inhoud wordt weergegeven in de lijst, in dit geval de namen van de kinderen. De lijst ziet er zo uit:

David
Colleen
Sharon
Ronnie
James

Een lijst die is gevuld door een DataProvider

De waarde van het gegevensveld is beschikbaar wanneer de gebruiker een item in de lijst selecteert door erop te klikken en de gebeurtenis `change` veroorzaakt. In het volgende voorbeeld worden een tekstgebied (`aTa`) en een gebeurtenishandler (`changeHandler`) aan het voorgaande voorbeeld toegevoegd om de verjaardag van het kind weer te geven wanneer de gebruiker een naam selecteert in de lijst.

```
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);

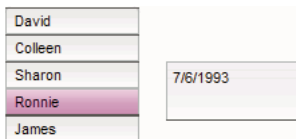
addChild(aList);
addChild(aTa);

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

aList.addEventListener(Event.CHANGE, changeHandler);

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

Wanneer de gebruiker nu de naam van een kind in de lijst selecteert, wordt in het tekstgebied de verjaardag van het kind weergegeven, zoals wordt weergegeven in de volgende afbeelding. Dit wordt gedaan door de functie `changeHandler()` wanneer deze de eigenschap `text` van de `TextArea`-component (`aTa.text`) instelt op de waarde van het gegevensveld in het geselecteerde item (`event.target.selectedItem.data`). De eigenschap `event.target` is het object dat de gebeurtenis heeft geactiveerd, in dit geval de component `List`.



Het gegevensveld van de `DataProvider` van een component `List` weergeven

U kunt andere gegevens dan tekst opnemen in een `DataProvider`. In het volgende voorbeeld worden filmclips opgenomen in een `DataProvider` die gegevens aanbiedt aan een component `TileList`. De `DataProvider` wordt gemaakt door `addItem()` aan te roepen om elk item toe te voegen nadat de filmclip, een gekleurd vak, is gemaakt.

```
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();
}
```

U kunt in plaats van een array ook XML-gegevens gebruiken om een DataProvider-object te vullen. Met de volgende code worden gegevens in het XML-object `employeesXML` opgeslagen en vervolgens wordt dat object doorgegeven als de waardeparameter van de constructorfunctie `DataProvider()`:

```
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;
```

U kunt gegevens als attributen van de XML-gegevens opgeven, zoals in de vorige code, of als eigenschappen van de XML-gegevens, zoals in de volgende code:


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

De DataProvider bevat tevens een set methoden en eigenschappen waarmee u deze kunt openen en bewerken. U kunt de API van de DataProvider gebruiken om items toe te voegen, te verwijderen, te vervangen, te sorteren en samen te voegen in een DataProvider.

Een DataProvider bewerken

U kunt items aan een DataProvider toevoegen met de methoden `addItem()` en `addItemAt()`. In het volgende voorbeeld worden items toegevoegd die de gebruiker invoert in het tekstveld van een bewerkbare ComboBox. Er is een ComboBox naar het werkgebied geslept en hieraan is de naam `aCb` gegeven.

```
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);
}
```

Via de DataProvider kunt u tevens componentitems vervangen en verwijderen. In het volgende voorbeeld worden de twee afzonderlijke List-componenten `listA` en `listB` geïmplementeerd en wordt een knop met het label `Sync` weergegeven. Wanneer de gebruiker op de knop klikt, wordt de methode `replaceItemAt()` gebruikt om de items in `listB` te vervangen door de items in `listA`. Als `listA` langer is dan `listB`, wordt de methode `addItem()` aangeroepen om de extra items toe te voegen aan `listB`. Als `listB` langer is dan `listA`, wordt de methode `removeItemAt()` aangeroepen om de extra items te verwijderen uit `listB`.

```
// 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++);
    }
}
}
}
}

```

U kunt een `DataProvider` ook gebruiken voor samenvoegings- en sorteerbewerkingen met de methoden `merge()`, `sort()` en `sortOn()`. In het volgende voorbeeld worden twee `DataGrid`-instanties (`aDg` en `bDg`) gevuld met gedeeltelijke spelerslijsten van twee softbalteams. Er wordt een knop met het label Merge toegevoegd en wanneer de gebruiker hierop klikt, voegt de gebeurtenishandler (`mrgHandler`) de lijst voor `bDg` samen met de lijst voor `aDg` en wordt het resulterende `DataGrid` gesorteerd op de kolom Naam.

```

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");
}
```

Zie de klasse `DataProvider` in de [Naslaggids voor ActionScript 3.0](#) voor meer informatie.

Werken met een CellRenderer

`CellRenderer` is een klasse die door op lijst gebaseerde componenten, zoals `List`, `DataGrid`, `TileList` en `ComboBox`, worden gebruikt om aangepaste celinhoud voor elke rij te bewerken en weer te geven. Een aangepaste cel kan tekst bevatten, een vooraf gebouwde component zoals `CheckBox`, of een weergaveobjectklasse die u kunt maken. Als u gegevens rendert met een aangepaste `CellRenderer`, kunt u de klasse `CellRenderer` uitbreiden of de interface `ICellRenderer` implementeren om uw eigen aangepaste `CellRenderer`-klasse te maken.

De klassen `List`, `DataGrid`, `TileList` en `ComboBox` zijn subclasses van de klasse `SelectableList`. De klasse `SelectableList` bevat de stijl `cellRenderer`. Met deze stijl wordt het weergaveobject gedefinieerd dat de component gebruikt om cellen te renderen.

U kunt de opmaak van de stijlen die worden gebruikt door de `CellRenderer` aanpassen door de methode `setRendererStyle()` van het lijstobject aan te roepen (zie “[Cellen opmaken](#)” op pagina 40). U kunt ook een aangepaste klasse definiëren om als de `CellRenderer` te gebruiken (zie “[Een aangepaste CellRenderer-klasse definiëren](#)” op pagina 41).

Cellen opmaken

De klasse `CellRenderer` bevat een aantal stijlen waarmee u de opmaak van de cel kunt bepalen.

Met de volgende stijlen kunt u de skins definiëren die worden gebruikt voor de verschillende toestanden van de cel (uitgeschakeld, omlaag, over en omhoog):

- `disabledSkin` en `selectedDisabledSkin`
- `downSkin` en `selectedDownSkin`
- `overSkin` en `selectedOverSkin`
- `upSkin` en `selectedUpSkin`

De volgende stijlen worden toegepast op tekstopmaak:

- disabledTextFormat
- textFormat
- textPadding

U kunt deze stijlen instellen door de methode `setRendererStyle()` van het lijstobject of de methode `setStyle()` van het `CellRenderer`-object aan te roepen. U kunt deze stijlen verkrijgen door de methode `getRendererStyle()` van het lijstobject of de methode `getStyle()` van het `CellRenderer`-object aan te roepen. U kunt ook via de eigenschap `rendererStyles` van het lijstobject of de methode `getStyleDefinition()` van het `CellRenderer`-object een object openen waarmee alle renderstijlen worden gedefinieerd (als benoemde eigenschappen van het object).

U kunt de methode `clearRendererStyle()` aanroepen om de standaardwaarde van een stijl te herstellen.

Gebruik de eigenschap `rowHeight` van het lijstobject om de hoogte van de rijen in de lijst op te halen of in te stellen.

Een aangepaste `CellRenderer`-klasse definiëren

Een klasse maken waarmee de klasse `CellRenderer` wordt uitgebreid om een aangepaste `CellRenderer` te definiëren

De volgende code bevat bijvoorbeeld twee klassen. De klasse `ListSample` instantieert een component `List` en gebruikt de andere klasse, `CustomRenderer`, om de celrenderer te definiëren die moet worden gebruikt voor de component `List`. De klasse `CustomRenderer` breidt de klasse `CellRenderer` uit.

- 1 Selecteer Bestand > Nieuw.
- 2 Selecteer Flash-bestand (ActionScript 3.0) in het dialoogvenster Nieuw document dat wordt weergegeven en klik vervolgens op OK.
- 3 Selecteer Venster > Componenten om het deelvenster Componenten weer te geven.
- 4 Sleep een component `List` van het deelvenster Componenten naar het werkgebied.
- 5 Selecteer Venster > Eigenschappen > Eigenschappen wanneer Flash Eigenschapcontrole niet weergeeft.
- 6 Selecteer de component `List` en stel de eigenschappen in het deelvenster Eigenschappen in:
 - Instantienaam: `myList`
 - B (breedte): 200
 - H (hoogte): 300
 - X: 20
 - Y: 20
- 7 Selecteer frame 1 van laag 1 in de tijdlijn en selecteer Venster > Handelingen.
- 8 Typ het volgende script in het deelvenster Handelingen:

```
myList.setStyle("cellRenderer", CustomCellRenderer);
myList.addItem({label:"Burger -- $5.95"});
myList.addItem({label:"Fries -- $1.95"});
```
- 9 Selecteer Bestand > Opslaan. Geef het bestand een naam en klik op OK.
- 10 Selecteer Bestand > Nieuw.
- 11 Selecteer ActionScript-bestand (ActionScript 3.0) in het dialoogvenster Nieuw document dat wordt weergegeven en klik vervolgens op OK.
- 12 Voer in het scriptvenster de volgende code in om de aangepaste `CustomCellRenderer`-klasse te definiëren:

```
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 Selecteer Bestand > Opslaan. Geef het bestand de naam CustomCellRenderer.as, sla het bestand op in dezelfde map als het FLA-bestand en klik op OK.

14 Selecteer Besturing > Film testen.

Een klasse gebruiken waarmee de interface ICellRenderer wordt geïmplementeerd om een aangepaste CellRenderer te definiëren

U kunt ook een CellRenderer definiëren met een klasse die de klasse DisplayObject overerft en de interface ICellRenderer implementeert. Met de volgende code worden bijvoorbeeld twee klassen gedefinieerd. De klasse ListSample2 voegt een List-object toe aan het weergaveoverzicht en de CellRenderer wordt zo gedefinieerd dat de klasse CustomRenderer wordt gebruikt. De klasse CustomRenderer breidt de klasse CheckBox (waarmee de klasse DisplayObject wordt uitgebreid) uit en implementeert de interface ICellRenderer. Met de klasse CustomRenderer worden de methoden getter en setter gedefinieerd voor de eigenschappen data en listData die zijn gedefinieerd in de interface ICellRenderer. Andere eigenschappen en methoden die zijn gedefinieerd in de interface ICellRenderer (de eigenschap selected en de methode setSize()), zijn al gedefinieerd in de klasse CheckBox:

- 1 Selecteer Bestand > Nieuw.
- 2 Selecteer Flash-bestand (ActionScript 3.0) in het dialoogvenster Nieuw document dat wordt weergegeven en klik vervolgens op OK.
- 3 Selecteer Venster > Componenten om het deelvenster Componenten weer te geven.
- 4 Sleep een component List van het deelvenster Componenten naar het werkgebied.
- 5 Selecteer Venster > Eigenschappen > Eigenschappen wanneer Flash Eigenschapcontrole niet weergeeft.
- 6 Selecteer de component List en stel de eigenschappen in het deelvenster Eigenschappen in:

- Instantienaam: myList
- B (breedte): 100
- H (hoogte): 300
- X: 20
- Y: 20

7 Selecteer frame 1 van laag 1 in de tijdlijn en selecteer Venster > Handelingen.

8 Typ het volgende script in het deelvenster Handelingen:

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

9 Selecteer Bestand > Opslaan. Geef het bestand een naam en klik op OK.

10 Selecteer Bestand > Nieuw.

11 Selecteer ActionScript-bestand (ActionScript 3.0) in het dialoogvenster Nieuw document dat wordt weergegeven en klik vervolgens op OK.

12 Voer in het scriptvenster de volgende code in om de aangepaste CustomCellRenderer-klasse te definiëren:

```
package
{
    import fl.controls.CheckBox;
    import fl.controls.listClasses.ICellRenderer;
    import fl.controls.listClasses.ListData;
    public class CustomCellRenderer extends CheckBox implements ICellRenderer {
        private var _listData:ListData;
        private var _data:Object;
        public function CustomCellRenderer() {
        }
        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 Selecteer Bestand > Opslaan. Geef het bestand de naam CustomCellRenderer.as, sla het bestand op in dezelfde map als het FLA-bestand en klik op OK.

14 Selecteer Besturing > Film testen.

Een symbool geruiken om een CellRenderer te definiëren

U kunt ook een symbool in de bibliotheek gebruiken om een CellRenderer te definiëren. Het symbool moet worden geëxporteerd voor ActionScript en aan de klassenaam voor het bibliotheeksymbool moet een klassebestand zijn gekoppeld waarmee de interface ICellRenderer wordt geïmplementeerd of de klasse CellRenderer (of een van de subklassen) wordt uitgebreid.

In het volgende voorbeeld wordt een aangepaste CellRenderer gedefinieerd met een bibliotheeksymbool.

1 Selecteer Bestand > Nieuw.

2 Selecteer Flash-bestand (ActionScript 3.0) in het dialoogvenster Nieuw document dat wordt weergegeven en klik vervolgens op OK.

3 Selecteer Venster > Componenten om het deelvenster Componenten weer te geven.

4 Sleep een component List van het deelvenster Componenten naar het werkgebied.

5 Selecteer Venster > Eigenschappen > Eigenschappen wanneer Flash Eigenschapcontrole niet weergeeft.

6 Selecteer de component List en stel de eigenschappen in het deelvenster Eigenschappen in:

- Instantienaam: myList

- B (breedte): 100
 - H (hoogte): 400
 - X: 20
 - Y: 20
- 7 Klik in het deelvenster Parameters en dubbelklik op de tweede kolom in de dataProvider-rij.
 - 8 Klik in het dialoogvenster Waarden twee keer op het plusteken om twee gegevenselementen toe te voegen (met labels ingesteld op label0 en label1) en klik vervolgens op de knop OK.
 - 9 Gebruik het gereedschap Tekst om een tekstveld in het werkgebied te tekenen.
 - 10 Wanneer het tekstveld is geselecteerd, stelt u de eigenschappen in Eigenschapcontrole in:
 - Texttype: Dynamische tekst
 - Instantienaam: textField
 - B (breedte): 100
 - Tekengrootte: 24
 - X: 0
 - Y: 0
 - 11 Selecteer Wijzigen > Omzetten in symbool terwijl het tekstveld is geselecteerd.
 - 12 Geef in het dialoogvenster Omzetten in symbool de volgende instellingen op en klik vervolgens op OK.
 - Naam: MyCellRenderer
 - Type: MovieClip
 - Exporteren voor ActionScript: Geselecteerd
 - Exporteren in eerste frame: Geselecteerd
 - Klasse: MyCellRenderer
 - Basisklasse: flash.display.MovieClip

Als er een waarschuwingsbericht over de ActionScript-klasse wordt weergegeven, klikt u in het bericht op de knop OK.
 - 13 Verwijder de instantie van het nieuwe filmclipsymbool uit het werkgebied.
 - 14 Selecteer frame 1 van laag 1 in de tijdlijn en selecteer Venster > Handelingen.
 - 15 Typ het volgende script in het deelvenster Handelingen:

```
myList.setStyle("cellRenderer", MyCellRenderer);
```
 - 16 Selecteer Bestand > Opslaan. Geef het bestand een naam en klik op OK.
 - 17 Selecteer Bestand > Nieuw.
 - 18 Selecteer ActionScript-bestand (ActionScript 3.0) in het dialoogvenster Nieuw document dat wordt weergegeven en klik vervolgens op OK.
 - 19 Voer in het scriptvenster de volgende code in om de klasse MyCellRenderer te definiëren:


```
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 Selecteer Bestand > Opslaan. Geef het bestand de naam MyCellRenderer.as, sla het bestand op in dezelfde map als het FLA-bestand en klik op OK.

21 Selecteer Besturing > Film testen.

Eigenschappen van CellRenderer

De eigenschap `data` is een object dat alle eigenschappen bevat die zijn ingesteld voor de `CellRenderer`. In de volgende klasse, waarmee een aangepaste `CellRenderer` wordt gedefinieerd waarmee de klasse `CheckBox` wordt uitgebreid, wordt bijvoorbeeld met de functie `setter` voor de eigenschap `data` de waarde van `data.label` doorgegeven aan de eigenschap `label` die wordt overerfd van de klasse `CheckBox`:

```
public class CustomRenderer extends CheckBox implements ICellRenderer {
    private var _listData:ListData;
    private var _data:Object;
    public function CustomRenderer() {
    }
    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;
    }
}
}
```

Met de eigenschap `selected` wordt gedefinieerd of een cel in de lijst wordt geselecteerd of niet.

Een CellRenderer toepassen voor een kolom van een DataGrid-object

Een object `DataGrid` kan meerdere kolommen bevatten en u kunt verschillende celrenderers opgeven voor elke kolom. Elke kolom van een `DataGrid` wordt vertegenwoordigd door een object `DataGridColumn` en de klasse `DataGridColumn` bevat de eigenschap `cellRenderer` waarvoor u de `CellRenderer` voor de kolom kunt definiëren.

Een CellRenderer definiëren voor een bewerkbare cel

Met de klasse `DataGridCellEditor` wordt een renderer voor bewerkbare cellen in een `DataGrid`-object gedefinieerd. Dit wordt de renderer voor een cel wanneer de eigenschap `editable` van het `DataGrid`-object wordt ingesteld op `true` en de gebruiker op de cel klikt die moet worden bewerkt. Als u een `CellRenderer` voor de bewerkbare cel wilt definiëren, stelt u de eigenschap `itemEditor` in voor elk element van de array `columns` van het `DataGrid`-object.

Een afbeelding, SWF-bestand of filmclip gebruiken als een CellRenderer

Met de klasse `ImageCell`, een subklasse van `CellRenderer`, wordt een object gedefinieerd dat wordt gebruikt om cellen te renderen waarvan de belangrijkste inhoud uit een afbeelding, SWF-bestand of filmclip bestaat. De klasse `ImageCell` bevat de volgende stijlen voor het definiëren van de weergave van de cel:

- `imagePadding`—De opvulling die fungeert als scheiding tussen de rand van de cel en de rand van de afbeelding, in pixels
- `selectedSkin`—De skin die wordt gebruikt om de geselecteerde toestand aan te duiden
- `textOverlayAlpha`—De dekking van de bedekking achter het cellabel

- `textPadding`—De opvulling die fungeert als scheiding tussen de rand van de cel en de rand van de tekst, in pixels
De klasse `ImageCell` is de standaard `CellRenderer` voor de klasse `TileList`.

Componenten toegankelijk maken

U kunt visuele inhoud in uw Flash-toepassingen via een schermlezer toegankelijk maken voor mensen met een beperkt gezichtsvermogen. Hiermee wordt de inhoud van het scherm hardop gelezen. Zie Hoofdstuk18, 'Toegankelijke inhoud maken', in *Flash gebruiken* voor informatie over de wijze waarop u een Flash-toepassing toegankelijk maakt voor een schermlezer.

Als u een ActionScript 3.0-component toegankelijk wilt maken voor een schermlezer, moet u ook de toegankelijkheidsklasse van die component importeren en de methode `enableAccessibility()` van die klasse aanroepen. U kunt de volgende ActionScript 3.0-componenten toegankelijk maken voor een schermlezer:

Component	Klasse Accessibility
Button	ButtonAccImpl
CheckBox	CheckBoxAccImpl
ComboBox	ComboBoxAccImpl
List	ListAccImpl
RadioButton	RadioButtonAccImpl
TileList	TileListAccImpl

De componenttoegankelijkheidsklassen bevinden zich in het pakket `fl.accessibility`. Als u een `CheckBox` toegankelijk wilt maken, moet u bijvoorbeeld de volgende instructies toevoegen aan de toepassing:

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

U stelt toegankelijkheid voor een component slechts eenmaal in, ongeacht het aantal instanties dat u maakt.

Opmerking: Wanneer u toegankelijkheid inschakelt, wordt het bestand iets groter omdat tijdens de compilatie de vereiste klassen worden opgenomen.

In de meeste componenten kan ook met het toetsenbord worden genavigeerd. Zie de secties over gebruikersinteractie van “[UI-componenten gebruiken](#)” op pagina 48 en de toegankelijkheidsklassen in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie over het inschakelen van toegankelijke componenten en het navigeren met het toetsenbord.

Hoofdstuk 4: UI-componenten gebruiken

In dit hoofdstuk wordt uitgelegd hoe u de gebruikersinterfacecomponenten (UI-componenten) van ActionScript 3.0 die met Flash worden geleverd, kunt gebruiken.

De component Button gebruiken

De component Button is een rechthoekige knop waarvan de grootte kan worden gewijzigd en waarop de gebruiker met de muisaanwijzer of de spatiebalk kan klikken om een actie in de toepassing te starten. U kunt een aangepast pictogram aan een Button toevoegen. U kunt ook het gedrag van een Button wijzigen van een drukknop in een schakelknop. Een schakelknop blijft ingedrukt wanneer hierop wordt geklikt en keert terug naar de toestand Omhoog wanneer er nogmaals op wordt geklikt.

Een Button is een fundamenteel onderdeel van vele formulieren en webtoepassingen. U kunt deze knoppen gebruiken wanneer u wilt dat de gebruiker een gebeurtenis start. De meeste formulieren hebben bijvoorbeeld een knop Verzenden. U kunt ook knoppen Vorige en Volgende aan een presentatie toevoegen.

Gebruikersinteractie met het component Button

U kunt een Button in een toepassing in- of uitschakelen. In de uitgeschakelde toestand ontvangt de component geen muis- of toetsenbordinput. Een ingeschakelde Button ontvangt focus wanneer u erop klikt of er met Tab naartoe gaat. Wanneer een instantie Button focus heeft, kunt u de volgende toetsen gebruiken om de instantie te beheren:

Toets	Beschrijving
Shift+Tab	Hiermee wordt de focus naar het vorige object verplaatst.
Spatiebalk	Hiermee drukt u op de knop of laat u deze los en activeert u de gebeurtenis <code>click</code> .
Tab	Hiermee wordt de focus naar het volgende object verplaatst.
Enter/Return	Hiermee wordt de focus naar het volgende object verplaatst als de knop als de standaard Button van FocusManager is ingesteld.

Zie de `IFocusManager`-interface en de klasse `FocusManager` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) en “[Werken met FocusManager](#)” op pagina 30 voor meer informatie over focusbeheer.

Een live voorvertoning van elke instantie Button geeft direct wijzigingen weer in parameters in Eigenschapcontrole of Componentcontrole tijdens het ontwerpen.

Opmerking: Als een pictogram groter dan de knop is, strekt het pictogram zich buiten de grenzen van de knop uit.

Als u een knop als de standaarddrukknop in een toepassing wilt aanwijzen (de knop die de gebeurtenis `click` ontvangt wanneer de gebruiker op Enter drukt), stelt u `FocusManager.defaultButton` in. De volgende code stelt bijvoorbeeld de standaardknop in op een knopinstantie met de naam `submitButton`.

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

Wanneer u de component Button aan een toepassing toevoegt, kunt u deze toegankelijk maken voor schermlezers door de volgende ActionScript-coderegels toe te voegen:

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

U stelt toegankelijkheid voor een component slechts eenmaal in, ongeacht het aantal instanties dat u maakt.

Parameters van de component Button

U kunt in Eigenschapcontrole instellen (Venster > Eigenschappen > Eigenschappen) of in Componentcontrole (Venster > Componentcontrole) de volgende ontwerpparameters voor elke instantie van de component Button: `emphasized`, `label`, `labelPlacement`, `selected` en `toggle`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Wanneer u een waarde toewijst aan deze parameters, stelt u de initiële status in van de eigenschap in de toepassing. Bij het instellen van de eigenschap in ActionScript wordt de ingestelde waarde in de parameter genegeerd. Zie de klasse Button in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

Een toepassing maken met de component Button

De volgende procedure laat zien hoe u een component Button tijdens het ontwerpen aan een toepassing kunt toevoegen. In dit voorbeeld wordt met de klasse Button de status van de component ColorPicker gewijzigd wanneer u erop klikt.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component Button van het deelvenster Componenten naar het werkgebied en voer de volgende waarden voor deze component in Eigenschapcontrole in:
 - Voer de instantienaam **aButton** in.
 - Voer **Show** in voor de parameter label.
- 3 Voeg een ColorPicker aan het werkgebied toe en geef deze de instantienaam **aCp**.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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;
    }
}
```

De tweede coderegel registreert de functie `clickHandler()` als de gebeurtenishandlerfunctie voor de gebeurtenis `MouseEvent.CLICK`. De gebeurtenis treedt op wanneer de gebruiker op de Button klikt, waardoor de functie `clickHandler()` afhankelijk van de waarde van de Button een van de volgende handelingen uitvoert:

- Met Show wordt de ColorPicker zichtbaar gemaakt en wordt het label van de Button in Disable gewijzigd.
- Met Disable wordt de ColorPicker uitgeschakeld en wordt het label van de Button in Enable gewijzigd.
- Met Enable wordt de ColorPicker ingeschakeld en wordt het label van de Button in Hide gewijzigd.
- Met Hide wordt de ColorPicker onzichtbaar gemaakt en wordt het label van de Button in Show gewijzigd.

5 Selecteer Besturing > Film testen om de toepassing uit te voeren.

Een toepassing maken met de component Button

In de volgende procedure wordt een schakelknop gemaakt via ActionScript en wordt het gebeurtenistype in het deelvenster Uitvoer weergegeven wanneer u op de Button klikt. In het voorbeeld wordt een instantie van Button gemaakt door de constructor van de klasse aan te roepen en wordt deze aan het werkgebied toegevoegd door de methode `addChild()` aan te roepen.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component Button van het deelvenster Componenten naar het deelvenster Bibliotheek van het huidige document.

Hiermee voegt u de component aan de bibliotheek toe, maar de component wordt hierdoor niet zichtbaar in de toepassing.

- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voer de volgende code in om een instantie Button te maken:

```
import fl.controls.Button;

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

De methode `move()` plaatst de knop op locatie 50 (x-coördinaat), 50 (y-coördinaat) in het werkgebied.

- 4 Voeg nu de volgende ActionScript-code toe om een gebeurtenislistener en een gebeurtenishandlerfunctie te maken:

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

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

- 5 Selecteer Besturing > Film testen.

Wanneer u op de knop klikt, wordt het bericht 'Event type: click' in het deelvenster Uitvoer weergegeven.

De component CheckBox gebruiken

Een CheckBox is een vierkant vakje dat kan worden in- of uitgeschakeld. Wanneer het is geselecteerd, wordt een vinkje in het vakje weergegeven. U kunt een tekstlabel aan een CheckBox toevoegen en deze links, rechts, boven of onder de CheckBox plaatsen.

U kunt meerdere componenten CheckBox gebruiken om een set met waarden `true` of `false` te verzamelen die elkaar niet uitsluiten. Een toepassing die bijvoorbeeld informatie verzamelt over de auto die u wilt kopen, maakt waarschijnlijk gebruik van componenten CheckBox zodat u bepaalde kenmerken kunt selecteren.

Gebruikersinteractie met CheckBox

U kunt een CheckBox in een toepassing in- of uitschakelen. Als een CheckBox is ingeschakeld en de gebruiker hierop of op het label klikt, ontvangt de CheckBox invoerfocus en geeft deze het geklikte uiterlijk weer. Als de gebruiker de aanwijzer verplaatst tot buiten het selectiegebied van een CheckBox of het label terwijl de muisknop wordt ingedrukt, keert het uiterlijk van de component terug naar de oorspronkelijke toestand en behoudt deze invoerfocus. De toestand van een CheckBox wordt pas gewijzigd wanneer de muis boven de component wordt losgelaten. Bovendien heeft de CheckBox twee uitgeschakelde toestanden, geselecteerd en niet geselecteerd, die respectievelijk gebruikmaken van `selectedDisabledSkin` en `disabledSkin`, die geen muis- of toetsenbordinteractie toestaan.

Wanneer een CheckBox is uitgeschakeld, geeft deze het uitgeschakelde uiterlijk weer, ongeacht gebruikersinteractie. In de uitgeschakelde toestand ontvangt een CheckBox geen muis- of toetsenbordinvoer.

Een instantie CheckBox ontvangt focus wanneer de gebruiker erop klikt of er met Tab naartoe gaat. Wanneer een instantie CheckBox focus heeft, kunt u de volgende toetsen gebruiken om de instantie te beheren:

Toets	Beschrijving
Shift+Tab	Hiermee wordt de focus naar het vorige element verplaatst.
Spatiebalk	Hiermee selecteert u de component of heft u de selectie ervan op en activeert u de gebeurtenis <code>change</code> .
Tab	Hiermee wordt de focus naar het volgende element verplaatst.

Zie “[Werken met FocusManager](#)” op pagina 30 en de klasse FocusManager in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie over focusbeheer.

Een live voorvertoning van elke instantie CheckBox geeft direct wijzigingen weer in parameters in Eigenschapcontrole of Componentcontrole tijdens het ontwerpen.

Wanneer u de component CheckBox aan een toepassing toevoegt, kunt u deze toegankelijk maken voor schermlezers door de volgende ActionScript-coderegels toe te voegen:

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

U stelt toegankelijkheid voor een component slechts eenmaal in, ongeacht het aantal componentinstanties dat u gebruikt.

Parameters van de component CheckBox

U kunt in Eigenschapcontrole of Componentcontrole de volgende ontwerpparameters instellen voor elke instantie van de component CheckBox: `label`, `labelPlacement` en `selected`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse `CheckBox` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

Een toepassing maken met de CheckBox

De volgende procedure laat aan de hand van een fragment uit een aanvraagformulier voor een lening zien hoe u een component CheckBox tijdens het ontwerpen aan een toepassing kunt toevoegen. Het formulier bevat de vraag of de aanvrager huiseigenaar is en biedt een CheckBox waarmee de gebruiker ‘ja’ kan antwoorden. Zo ja, geeft het formulier twee keuzerondjes weer waarmee voor de gebruiker de relatieve waarde van het huis kan aangeven.

Een toepassing maken met de component CheckBox

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component CheckBox van het deelvenster Componenten naar het werkgebied.
- 3 Ga als volgt te werk in Eigenschapcontrole:
 - Voer bij de instantienaam **homeCh** in.
 - Voer **140** in voor de waarde `width` (W).
 - Voer bij de parameter label **Huiseigenaar?** in.
- 4 Sleep twee instanties van de component RadioButton van het deelvenster Componenten naar de werkruimte en zet ze onder en rechts van de CheckBox neer. Voer in Eigenschapcontrole de volgende waarden voor ze in:
 - Voer bij de instantienamen **underRb** en **overRb** in.
 - Voer **120** in voor de parameter `W` (breedte) van beide RadioButtons.
 - Voer **Minder dan € 500.000?** in voor de parameter label van `underRb`.
 - Voer **Meer dan € 500.000?** in voor de parameter label van `overRb`.
 - Voer **valueGrp** in voor de parameter `groupName` van beide RadioButtons.
- 5 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:


```
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;
}
```

Met deze code maakt u een gebeurtenishandler voor een gebeurtenis `CLICK` die de RadioButtons `underRb` en `overRb` inschakelt wanneer de CheckBox `homeCh` is geselecteerd en deze uitschakelt wanneer `homeCh` niet is geselecteerd. Zie de klasse `MouseEvent` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

6 Selecteer Besturing > Film testen.

Het volgende voorbeeld dupliceert de voorafgaande toepassing maar maakt de CheckBox en RadioButtons met ActionScript.

Een CheckBox maken met ActionScript

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component CheckBox en de component RadioButton van het deelvenster Componenten naar het deelvenster Bibliotheek van het huidige document. Als het deelvenster Bibliotheek niet is geopend, drukt u op `Ctrl+L` of selecteert u `Venster > Bibliotheek` om het deelvenster Bibliotheek te openen.

Op deze manier zijn de componenten voor de toepassing beschikbaar maar worden ze niet in het werkgebied geplaatst.

- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende code toe om componentinstanties te maken en plaatsen:

```
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?";
```

Deze code gebruikt de constructors `CheckBox()` en `RadioButton()` om de componenten te maken en de methode `addChild()` om deze in het werkgebied te plaatsen. De methode `move()` wordt gebruikt om de componenten op de gewenste locatie in het werkgebied te plaatsen.

- 4 Voeg nu de volgende ActionScript-code toe om een gebeurtenislistener en een gebeurtenishandlerfunctie te maken:

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

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

Met deze code maakt u een gebeurtenishandler voor een gebeurtenis `CLICK` die de keuzerondjes `underRb` en `overRb` inschakelt wanneer de CheckBox `homeCh` is geselecteerd en deze uitschakelt wanneer `homeCh` niet is geselecteerd. Zie de klasse `MouseEvent` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

- 5 Selecteer Besturing > Film testen.

De component ColorPicker gebruiken

Met de component `ColorPicker` kan de gebruiker een kleur uit een staallijst te kiezen. In de standaardmodus van de `ColorPicker` wordt een enkele kleur in een vierkante knop weergegeven. Wanneer de gebruiker op de knop klikt, wordt een lijst met beschikbare kleuren in een staallijst weergegeven, samen met een tekstveld dat de hexadecimale waarde van de huidige kleurselectie weergeeft.

U kunt de kleuren die in `ColorPicker` worden weergegeven instellen door zijn eigenschap `colors` op de kleurwaarden in te stellen die u wilt laten weergegeven.

Gebruikersinteractie met de component ColorPicker

In een `ColorPicker` kan de gebruiker een kleur selecteren en deze op een ander object in de toepassing toepassen. Als u de gebruiker bijvoorbeeld wilt toestaan elementen in de toepassing aan te passen, zoals de achtergrondkleur of tekstkleur, kunt u een `ColorPicker` opnemen en de kleur toepassen die de gebruiker selecteert.

Een gebruiker kiest een kleur door op de staalkleur in het deelvenster te klikken of door de hexadecimale waarde in het tekstveld in te voeren. Wanneer de gebruiker een kleur kiest, kunt u de eigenschap `selectedColor` van `ColorPicker` gebruiken om de kleur op tekst of een ander object in de toepassing toe te passen.

Een instantie `ColorPicker` ontvangt focus als de gebruiker er met de muis overheen beweegt of er met Tab naartoe gaat. Wanneer het deelvenster `Stalen` van `ColorPicker` is geopend, kunt u de volgende toetsen gebruiken om het te beheren:

Toets	Beschrijving
Startpagina	Hiermee wordt de selectie naar de eerste kleur in het deelvenster <code>Stalen</code> verplaatst.
Pijl-omhoog	Hiermee wordt de selectie een rij hoger in het deelvenster <code>Stalen</code> geplaatst.
Pijl-omlaag	Hiermee wordt de selectie een rij lager in het deelvenster <code>Stalen</code> geplaatst.
Pijl-rechts	Hiermee wordt de selectie een kleur naar rechts in het deelvenster <code>Stalen</code> geplaatst.
Pijl-links	Hiermee wordt de selectie een kleur naar links in het deelvenster <code>Stalen</code> geplaatst.
End	Hiermee wordt de selectie naar de laatste kleur in het deelvenster <code>Stalen</code> geplaatst.

Parameters van de component ColorPicker

U kunt de volgende ontwerpparameters in Eigenschapcontrole of Componentcontrole voor elke instantie CheckBox instellen: `selectedColor` en `showTextField`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse `ColorPicker` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

Een toepassing maken met de component ColorPicker

Het volgende voorbeeld laat zien hoe u een component `ColorPicker` tijdens het ontwerpen aan een toepassing toevoegt. Elke keer dat u in dit voorbeeld de kleur in de `ColorPicker` wijzigt, roept de functie `changeHandler()` de functie `drawBox()` aan om een nieuw vak te tekenen met de kleur die u hebt geselecteerd in de `ColorPicker`.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een `ColorPicker` van het deelvenster Componenten naar het midden van het werkgebied en geef deze de instantienaam `aCp`.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen.
- 5 Klik op de `ColorPicker` en selecteer een kleur waarmee het vak moet worden gekleurd.

Een ColorPicker maken met ActionScript

Het volgende voorbeeld gebruikt de constructor `ColorPicker()` en `addChild()` om een `ColorPicker` in het werkgebied te maken. Deze stelt de eigenschap `colors` in op de kleurwaarden voor rood (`0xFF0000`), groen (`0x00FF00`) en blauw (`0x0000FF`) om de kleuren op te geven die de `ColorPicker` moet weergegeven. Er wordt ook een `TextArea` gemaakt en elke keer dat u een andere kleur in de `ColorPicker` selecteert, wijzigt het voorbeeld de kleur van de tekst in de `TextArea` om met deze kleur overeen te komen.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component `ColorPicker` van het deelvenster Componenten naar het deelvenster Bibliotheek.
- 3 Sleep de component `TextArea` van het deelvenster Componenten naar het deelvenster Bibliotheek.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen.

De component ComboBox gebruiken

Met de component ComboBox kan de gebruiker één item in een vervolgkeuzelijst selecteren. Een ComboBox kan statisch of bewerkbaar zijn. In een bewerkbare ComboBox kan de gebruiker rechtstreeks tekst aan de bovenkant van de lijst in het tekstveld invoeren. Als de vervolgkeuzelijst het einde van het document raakt, opent deze naar boven in plaats van naar beneden. De ComboBox is opgebouwd uit drie subcomponenten: de componenten BaseButton, TextInput en List.

In een bewerkbare ComboBox, is alleen de knop het raakgebied en niet het tekstvak. Voor een statische ComboBox vormen de knop en het tekstvak het raakgebied. Het raakgebied reageert door de vervolgkeuzelijst te openen en sluiten.

Wanneer de gebruiker een selectie in de lijst maakt, zowel met de muis als het toetsenbord, wordt het label van de selectie naar het tekstveld aan de bovenkant van de ComboBox gekopieerd.

Gebruikersinteractie met de component ComboBox

U kunt een component ComboBox in elk formulier of toepassing gebruiken waarin een enkele keuze uit een lijst moet worden gemaakt. U kunt bijvoorbeeld een vervolgkeuzelijst van provincies aanbieden in het adresformulier van een klant. Voor meer complexe scenario's kunt u een bewerkbare ComboBox gebruiken. In een toepassing met een routebeschrijving kunt u bijvoorbeeld een bewerkbare ComboBox gebruiken, zodat de gebruiker het beginpunt en de bestemming kan invoeren. Deze vervolgkeuzelijst zou het laatst ingevoerde adres bevatten.

Als de ComboBox bewerkbaar is (de eigenschap `editable` is `true`) verwijderen de volgende toetsen de focus van het tekst invoervak en wordt de vorige waarde vastgehouden. De uitzondering hierop vormt de Enter-toets, die eerst de nieuwe waarde toepast wanneer de gebruiker tekst invoert.

Toets	Beschrijving
Shift+Tab	Hiermee wordt de focus naar het vorige item verplaatst. Als er een nieuw item wordt geselecteerd, wordt een gebeurtenis <code>change</code> verzonden.
Tab	Hiermee wordt de focus naar het volgende item verplaatst. Als er een nieuw item wordt geselecteerd, wordt een gebeurtenis <code>change</code> verzonden.
Pijl-omlaag	Hiermee wordt de selectie een item naar beneden geplaatst.
End	Hiermee wordt de selectie naar de onderkant van de lijst verplaatst.
Escape	Hiermee wordt de vervolgkeuzelijst gesloten en wordt de focus naar ComboBox geretourneerd.
Enter	Hiermee wordt de vervolgkeuzelijst gesloten en wordt de focus naar ComboBox geretourneerd. Wanneer de ComboBox bewerkbaar is en de gebruiker tekst invoert, stelt de Enter-toets de waarde op de ingevoerde tekst in.
Startpagina	Hiermee wordt de selectie naar de bovenkant van de lijst verplaatst.
PageUp	Hiermee wordt de selectie een pagina naar boven verplaatst.
PageDown	Hiermee wordt de selectie een pagina naar beneden verplaatst.

Wanneer u de component ComboBox aan een toepassing toevoegt, kunt u deze toegankelijk maken voor schermlezers door de volgende ActionScript-coderegels toe te voegen:

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

U stelt toegankelijkheid voor een component slechts eenmaal in, ongeacht het aantal componentinstanties dat u gebruikt.

Parameters van de component ComboBox

U kunt de volgende parameters in Eigenschapcontrole of Componentcontrole voor elke instantie CheckBox instellen: `dataProvider`, `editable`, `prompt` en `rowCount`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse `ComboBox` in de *Naslagids voor ActionScript 3.0 voor Adobe Flash Professional CS5* voor informatie over de mogelijk waarden voor deze parameters. Zie “[De parameter dataProvider gebruiken](#)” op pagina 32 voor informatie over het gebruik van de parameter `dataProvider`.

Een toepassing maken met de component ComboBox

De volgende procedure laat zien hoe u een component ComboBox tijdens het ontwerpen aan een toepassing kunt toevoegen. De ComboBox is bewerkbaar en als u **Add** in het tekstveld invoert, voegt het voorbeeld een item aan de vervolgkeuzelijst toe.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een ComboBox naar het werkgebied en geef deze de instantienaam **aCb**. Stel op het tabblad Parameters de parameter `editable` in op `true`.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voer de volgende code in:

```
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 Selecteer Besturing > Film testen.

Een ComboBox maken met ActionScript

In het volgende voorbeeld wordt een ComboBox gemaakt met ActionScript en wordt deze gevuld met een lijst van universiteiten in en rond San Francisco, Californië. Het stelt de eigenschap `width` van de ComboBox in zodat de volledige vraagtekst kan worden weergegeven en stelt de eigenschap `dropdownWidth` enigszins breder in om de langste universiteitnaam weer te geven.

Het voorbeeld maakt een lijst van universiteiten in een instantie Array, waarbij de eigenschap `label` wordt gebruikt om de schoolnamen op te slaan en de eigenschap `data` om de URL's van elke school op te slaan. De Array wordt aan de ComboBox toegewezen door zijn eigenschap `dataProvider` in te stellen.

Wanneer de gebruiker een universiteit uit de lijst selecteert, wordt een gebeurtenis `Event.CHANGE` geactiveerd en een aanroep naar de functie `changeHandler()` gemaakt, waardoor de eigenschap `data` in een URL-aanvraag voor toegang tot de website van een school wordt geladen.

De laatste regel stelt de eigenschap `selectedIndex` van de instantie ComboBox op -1 in om de vraag opnieuw weer te geven wanneer de lijst wordt gesloten. Zoniet, zou de vraag worden vervangen door de naam van de school die het laatst is geselecteerd.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component ComboBox van het deelvenster Componenten naar het deelvenster Bibliotheek.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen.

U kunt dit voorbeeld in de Flash-ontwerpomgeving implementeren en uitvoeren maar u ontvangt waarschuwingsberichten als u probeert toegang te krijgen tot de websites van de universiteit door op de items in de ComboBox te klikken. Benader de volgende locatie om toegang te krijgen tot de volledige functionele ComboBox op het internet:

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

De component DataGrid gebruiken

Met de component DataGrid kunt u gegevens in een raster van rijen en kolommen weergeven. Deze gegevens worden getekend op basis van een array of extern XML-bestand dat u in een array voor de DataProvider kunt parsen. De component DataGrid bevat verticaal en horizontaal schuiven, gebeurtenisondersteuning (inclusief ondersteuning voor bewerkbare cellen) en sorteermogelijkheden.

U kunt de grootte van kenmerken wijzigen en kenmerken zoals het lettertype, de kleur en de randen van kolommen in een raster aanpassen. U kunt een aangepaste filmclip als een celrenderder voor een kolom in een raster gebruiken. (Een celrenderder geeft de inhoud van een cel weer.) U kunt de schuifbalken uitschakelen en de methoden `DataGrid` gebruiken om een weergavestijl van één pagina te maken. Zie de klasse `DataGridColumn` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie over aanpassingen.

Meer Help-onderwerpen

[De component `DataGrid` maken, vullen en het formaat ervan wijzigen](#)

[De component `DataGrid` aanpassen en sorteren](#)

[Gegevens in de component `DataGrid` filteren en opmaken](#)

Gebruikersinteractie met de component `DataGrid`

U kunt de muis en het toetsenbord gebruiken om met een component `DataGrid` te communiceren.

Als de eigenschap `sortableColumns` en de eigenschap `sortable` van de kolom beide `true` zijn, zorgt het klikken op een kolomkop ervoor dat de gegevens op de waarden van de kolom worden gesorteerd. U kunt de mogelijkheid tot het sorteren van een individuele kolom uitschakelen door de eigenschap `sortable` in te stellen op `false`.

Als de eigenschap `resizableColumns` de waarde `true` heeft, kunt u de grootte van kolommen aanpassen door de kolomscheiders in de koprij te slepen.

Als er op een bewerkbare cel wordt geklikt, krijgt die cel focus en als er op een niet bewerkbare cel wordt geklikt heeft dit geen effect op de focus. Een individuele cel is bewerkbaar wanneer de eigenschap `DataGrid.editable` en `DataGridColumn.editable` van de cel `true` zijn.

Zie de klassen `DataGrid` en `DataGridColumn` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

Wanneer een instantie `DataGrid` focus heeft doordat erop is geklikt of doordat er met Tab naartoe is gegaan, kunt u de volgende toetsen gebruiken om de instantie te beheren:

Toets	Beschrijving
Pijl-omlaag	Wanneer een cel wordt bewerkt, schuift het invoegpunt naar het einde van de celtekst. Als een cel niet bewerkbaar is, handelt Pijl-omlaag het selecteren op dezelfde manier af als de component <code>List</code> .
Pijl-omhoog	Wanneer een cel wordt bewerkt, schuift het invoegpunt naar het begin van de celtekst. Als een cel niet bewerkbaar is, handelt Pijl-omhoog het selecteren op dezelfde manier af als de component <code>List</code> .
Shift+Pijl-omhoog/Pijl-omlaag	Als de <code>DataGrid</code> niet bewerkbaar is en als <code>allowMultipleSelection</code> de waarde <code>true</code> heeft, worden rijen opeenvolgend geselecteerd. Als u met de tegenovergestelde pijl de richting omkeert heft u de selectie van de geselecteerde rijen op totdat u de beginrij passeert, waarop de rijen in die richting worden geselecteerd.
Shift+klikken	Als <code>allowMultipleSelection</code> <code>true</code> is, worden alle rijen tussen de geselecteerde rij en het huidige invoegteken (gemarkeerde cel) geselecteerd.
Ctrl+klikken	Als <code>allowMultipleSelection</code> <code>true</code> is, worden alle additionele rijen geselecteerd. Ze hoeven niet opeenvolgend te zijn.
Pijl-rechts	Wanneer een cel wordt bewerkt, schuift het invoegpunt een teken naar rechts. Als een cel niet bewerkbaar is, heeft Pijl-rechts geen effect.
Pijl-links	Wanneer een cel wordt bewerkt, schuift het invoegpunt een teken naar links. Als een cel niet bewerkbaar is, heeft Pijl-links geen effect.
Startpagina	Hiermee wordt de eerste rij in de <code>DataGrid</code> geselecteerd.

Toets	Beschrijving
End	Hiermee wordt de laatste rij in de DataGrid geselecteerd.
PageUp	Hiermee wordt de eerste rij op een pagina in de DataGrid geselecteerd. Een pagina bestaat uit het aantal rijen dat de DataGrid zonder schuiven kan weergeven.
PageDown	Hiermee wordt de laatste rij op een pagina in de DataGrid geselecteerd. Een pagina bestaat uit het aantal rijen dat de DataGrid zonder schuiven kan weergeven.
Return/Enter/Shift+Enter	Wanneer een cel bewerkbaar is, wordt de wijziging doorgevoerd en wordt het invoegpunt naar een cel in dezelfde kolom, op de volgende rij geplaatst (boven of beneden, afhankelijk van de instelling van de Shift-toets).
Shift+Tab/Tab	Als de DataGrid bewerkbaar is, wordt de focus naar het volgende/vorige item verplaatst totdat het einde van de kolom wordt bereikt en wordt vervolgens naar de vorige/volgende rij verplaatst totdat de eerste of laatste cel wordt bereikt. Als de eerste cel is geselecteerd, verplaatst Shift+Tab de focus naar het vorige besturingselement. Als de laatste cel is geselecteerd, verplaatst Tab de focus naar het volgende besturingselement. Als de DataGrid niet bewerkbaar is, wordt de focus naar het vorige/volgende besturingselement verplaatst.

U kunt de component DataGrid gebruiken als de basis voor verschillende soorten toepassingen die met gegevens werken. U kunt eenvoudig een tabellarisch opgemaakt gegevensaanzicht weergeven, maar u kunt ook de celrenderermogelijkheden gebruiken om geavanceerde en bewerkbare gebruikersinterfaceonderdelen te maken. De component DataGrid wordt doorgaans voor de volgende doeleinden gebruikt:

- Een webmailclient
- Pagina's met zoekresultaten
- Spreadsheets-toepassingen zoals rekenmachines voor leningen en belastingformuliertoepassingen

Wanneer u een toepassing ontwerpt met de component DataGrid, is het nuttig om het ontwerp van de component List te begrijpen omdat de klasse DataGrid de klasse SelectableList uitbreidt. Zie de klassen SelectableList en List in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5 voor meer informatie over de klasse SelectableList en de component List](#).

Wanneer u de component DataGrid aan een toepassing toevoegt, kunt u deze toegankelijk maken voor een schermlezer door de volgende ActionScript-coderegels toe te voegen:

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

U stelt toegankelijkheid voor een component slechts eenmaal in, ongeacht het aantal instanties van de component. Raadpleeg Hoofdstuk 18, 'Toegankelijke inhoud maken' in *Flash gebruiken* voor meer informatie.

Parameters van de component DataGrid

U kunt de volgende ontwerpparameters in Eigenschapcontrole of Componentcontrole voor elke componentinstantie DataGrid instellen: allowMultipleSelection, editable, headerHeight, horizontalLineScrollSize, horizontalPageScrollSize, horizontalScrollPolicy, resizableColumns, rowHeight, showHeaders, verticalLineScrollSize, verticalPageScrollSize en verticalScrollPolicy. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse DataGrid in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

Een toepassing maken met de component DataGrid

Als u een toepassing met de component DataGrid wilt maken, moet u eerst bepalen waar de gegevens vandaan komen. Doorgaans komen de gegevens uit een Array, die u in het raster kunt opnemen door de eigenschap `dataProvider` in te stellen. U kunt ook de methoden van de klassen DataGrid en DataGridColumn gebruiken om gegevens aan het raster toe te voegen.

Een lokale gegevensaanbieder met een component DataGrid gebruiken:

In dit voorbeeld wordt een DataGrid gemaakt om het rooster van een softbalteam weer te geven. Het voorbeeld definieert het rooster in een Array (`aRoster`) en wijst deze toe aan de eigenschap `dataProvider` van de DataGrid.

- 1 Selecteer in Flash Bestand > Nieuw en selecteer vervolgens Flash-bestand (ActionScript 3.0).
- 2 Sleep de component DataGrid van het deelvenster Componenten naar het werkgebied.
- 3 Voer in Eigenschapcontrole de instantienaam **aDg** in.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdlijn en voeg de volgende ActionScript-code toe:

```
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);
};
```

De functie `bldRosterGrid()` stelt de grootte van DataGrid in en stelt de kolomvolgorde en de grootte van de kolommen in.

- 5 Selecteer Besturing > Film testen.

In een toepassing kolommen opgeven en sorteren toevoegen voor een component DataGrid

U kunt op elke kolomkop klikken om de inhoud van DataGrid op basis van de waarden van de kolom in aflopende volgorde te sorteren.

In het volgende voorbeeld wordt de methode `addColumn()` gebruikt om instanties van `DataGridColumn` aan een `DataGrid` toe te voegen. De kolommen vertegenwoordigen de namen van spelers en hun scores. In dit voorbeeld wordt bovendien de eigenschap `sortOptions` ingesteld om de sorteeropties voor elke kolom op te geven: `Array.CASEINSENSITIVE` voor de kolom Naam en `Array.NUMERIC` voor de kolom Score. Het wijzigt vervolgens de grootte van `DataGrid` door de lengte op het aantal rijen in te stellen en de breedte op 200.

- 1 Selecteer in Flash Bestand > Nieuw en selecteer vervolgens Flash-bestand (ActionScript 3.0).
- 2 Sleep de component `DataGrid` van het deelvenster Componenten naar het werkgebied.
- 3 Voer in Eigenschapcontrole de instantienaam `aDg` in.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
import fl.controls.dataGridClasses.DataGridColumn;
import fl.events.DataGridEvent;
import fl.data.DataProvider;
// Create columns to enable sorting of data.
var nameDGC:DataGridColumn = new DataGridColumn("name");
nameDGC.sortOptions = Array.CASEINSENSITIVE;
var scoreDGC:DataGridColumn = new DataGridColumn("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 Selecteer Besturing > Film testen.

Een componentinstantie DataGrid maken met ActionScript

In het volgende voorbeeld wordt met ActionScript een `DataGrid` gemaakt en aangevuld met een `Array` van spelernamen en scores.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component `DataGrid` van het deelvenster Componenten naar het deelvenster Bibliotheek van het huidige document.

Hiermee voegt u de component aan de bibliotheek toe, maar de component wordt hierdoor niet zichtbaar in de toepassing.

- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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);
```

Deze code maakt een instantie `DataGrid` en past vervolgens de grootte van het raster aan en positioneert deze.

- 4 Maak een array, voeg gegevens aan de array toe en identificeer de array als de gegevensaanbieder voor DataGrid:

```
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 Selecteer Besturing > Film testen.

Een DataGrid met een XML-bestand laden

Het volgende voorbeeld gebruikt de klasse DataGridColumn om kolommen voor DataGrid te maken. Het vult de DataGrid door een object XML door te geven als de parameter `value` van de constructor `DataProvider()`.

- 1 Gebruik een teksteditor om een XML-bestand te maken met de volgende gegevens en sla het bestand op onder de naam `team.xml` in dezelfde map waarin u het FLA-bestand opslaat.

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

- 2 Maak een nieuw Flash-document (ActionScript 3.0).
- 3 Dubbelklik in het deelvenster Componenten op de component DataGrid om deze aan het werkgebied toe te voegen.
- 4 Voer in Eigenschapcontrole de instantienaam **aDg** in.
- 5 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen.

De component Label gebruiken

In de component Label wordt één regel tekst weergegeven, doorgaans om een ander element of activiteit op een webpagina te identificeren. U kunt opgeven dat een label met HTML wordt opgemaakt om voordeel te doen van de tekstopmaaktags. U kunt ook de uitlijning en grootte van een label bepalen. Componenten Label hebben geen randen, kunnen geen focus krijgen en zenden geen gebeurtenissen uit.

Een live voorvertoning van elke instantie Label geeft direct wijzigingen weer in parameters in Eigenschapcontrole of Componentcontrole tijdens het ontwerpen. Aangezien de label geen rand heeft, kan de live voorvertoning alleen worden weergegeven wanneer de tekstparameter wordt ingesteld.

Gebruikersinteractie met de component Label

Gebruik een component Label om een tekstlabel voor een andere component in een formulier te maken, zoals het label 'Naam:' links van een veldTextInput waarin een gebruiksnaam kan worden ingevoerd. Het is raadzaam een component Label in plaats van een standaardtekstveld te gebruiken omdat u zo stijlen kunt gebruiken om een consequent uiterlijk aan te houden.

Als u een component Label wilt roteren, moet u de lettertypen insluiten, zoniet wordt deze niet weergegeven wanneer u de film test.

Parameters van de component label

U kunt in Eigenschapcontrole of Componentcontrole de volgende ontwerpparameters instellen voor elke instantie van de component CheckBox: `autoSize`, `condenseWhite`, `selectable`, `text` en `wordWrap`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse Label in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

Een toepassing met de component Label maken

De volgende procedure laat zien hoe u een component Label tijdens het ontwerpen aan een toepassing kunt toevoegen. In dit voorbeeld geeft het label alleen de tekst 'Expiration Date' weer.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component Label van het deelvenster Componenten naar het werkgebied en voer de volgende waarden voor deze component in Eigenschapcontrole in:
 - Voer bij de instantienaam **aLabel** in.
 - Voer **80** in voor de waarde W.
 - Voer **100** in voor de waarde X.
 - Voer **100** in voor de waarde Y.
 - Voer **Expiration Date** in voor de parameter `text`.
- 3 Sleep een component TextArea naar het werkgebied en voer de volgende waarden voor deze component in Eigenschapcontrole in:
 - Voer bij de instantienaam **aTa** in.
 - Voer **22** in voor de waarde H.
 - Voer **200** in voor de waarde X.
 - Voer **100** in voor de waarde Y.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen.

Een componentinstantie Label maken met ActionScript

Het volgende voorbeeld maakt met ActionScript een parameter Label. Het gebruikt Label om de functie van een component ColorPicker te identificeren en de eigenschap `htmlText` om opmaak op de tekst van Label toe te passen.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component Label van het deelvenster Componenten naar het deelvenster Bibliotheek van het huidige document.
- 3 Sleep een component ColorPicker van het deelvenster Componenten naar het deelvenster Bibliotheek van het huidige document.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen.

De component List gebruiken

De component List is een schuifbaar vak met een enkelvoudige of meervoudige keuzelijst. Een lijst kan ook afbeeldingen weergeven, inclusief andere componenten. U voegt de items die in de lijst worden weergegeven toe via het dialoogvenster Waarden dat wordt weergegeven wanneer u op de labels of op de parameterelden data klikt. U kunt ook de methoden `List.addItem()` en `List.addItemAt()` gebruiken om items aan de lijst toe te voegen.

De component List gebruikt een op nul gebaseerde index, waarbij het item met index nul het bovenste item is dat wordt weergegeven. Wanneer u met de klassemethoden van List items aan de lijst toevoegt, items uit de lijst verwijdert of items vervangt, moet u mogelijk de index van het item in de lijst opgeven.

Gebruikersinteractie met de component List

U kunt een lijst instellen zodat de gebruiker een enkele selectie of meerdere selecties kan maken. Een gebruiker die bijvoorbeeld een e-commercewebsite bezoekt moet kunnen selecteren welk item hij wil kopen. De lijst bevat 30 items. De gebruiker kan door de lijst schuiven en een item kiezen door erop te klikken.

U kunt ook een List ontwerpen die aangepaste filmclips als rijen gebruikt zodat u meer informatie aan de gebruiker kunt tonen. In een e-mailtoepassing kan iedere mailbox bijvoorbeeld een component List zijn en kan elke rij pictogrammen bevatten om de prioriteit en status aan te geven.

De List ontvangt focus wanneer u erop klikt of er met de Tab naartoe gaat en u kunt de volgende toetsen gebruiken om dit te beheren:

Toets	Beschrijving
Alfanumerieke toetsen	Hiermee wordt naar het volgende item gegaan waarvan het eerste teken in zijn label <code>Key.getAscii()</code> is.
Ctrl	Schakelknop die meerdere niet-openvolgende selecties en het opheffen van selecties toestaat.
Pijl-omlaag	Hiermee wordt de selectie een item naar beneden verplaatst.
Startpagina	Hiermee wordt de selectie naar de bovenkant van de lijst verplaatst.
PageDown	Hiermee wordt de selectie een pagina naar beneden verplaatst.
PageUp	Hiermee wordt de selectie een pagina naar boven verplaatst.
Shift	Hiermee wordt opeenvolgende selectie toegestaan.
Pijl-omhoog	Hiermee wordt de selectie een item naar boven verplaatst.

Opmerking: De schuifgrootten zijn in pixels en niet in rijen.

Opmerking: De paginagrootte die wordt gebruikt door de toetsen `PageUp` en `PageDown` is één minder dan het aantal items dat in het scherm past. Als u bijvoorbeeld een vervolgkeuzelijst met tien regels doorloopt, worden de items 0-9, 9-18, 18-27 enzovoort weergegeven, waarbij per pagina een item wordt overlapt.

Zie de `IFocusManager`-interface en de klasse `FocusManager` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) en “[Werken met FocusManager](#)” op pagina 30 voor meer informatie over focusbeheer.

Een live voorvertoning van elke instantie List in het werkgebied geeft direct wijzigingen weer in parameters in Eigenschapcontrole of Componentcontrole tijdens het ontwerpen.

Wanneer u de component List aan een toepassing toevoegt, kunt u deze toegankelijk maken voor schermlezers door de volgende ActionScript-coderegels toe te voegen:

```
import fl.accessibility.ListAccImpl;

ListAccImpl.enableAccessibility();
```

U stelt toegankelijkheid voor een component slechts eenmaal in, ongeacht het aantal instanties van de component. Raadpleeg Hoofdstuk 18, ‘Toegankelijke inhoud maken’ in *Flash gebruiken* voor meer informatie.

Parameters van de component List

U kunt in Eigenschapcontrole of Componentcontrole de volgende parameters instellen voor elke instantie van de component List: `allowMultipleSelection`, `,`, `dataProvider`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `multipleSelection`, `verticalLineScrollSize`, `verticalPageScrollSize` en `verticalScrollPolicy`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse List in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters. Zie “[De parameter dataProvider gebruiken](#)” op pagina 32 voor informatie over het gebruik van de parameter `dataProvider`.

Een toepassing met de component List maken

Het volgende voorbeeld laat zien hoe u een component List tijdens het ontwerpen aan een toepassing kunt toevoegen.

Een component List aan een toepassing toevoegen

In dit voorbeeld bestaat List uit labels die automodellen en gegevensvelden met prijzen identificeren.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component List van het deelvenster Componenten naar het werkgebied.
- 3 Ga als volgt te werk in Eigenschapcontrole:
 - Voer de instantienaam **aList** in.
 - Wijs de waarde **200** aan W (breedte) toe.
- 4 Gebruik het gereedschap Tekst om een tekstveld onder **aList** te maken en geef het de instantienaam **aTf**.
- 5 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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;
}
```

Deze code gebruikt de methode `addItem()` om **aList** met drie items te vullen, en wijst elk de waarde `label` (die in de lijst wordt weergegeven) en de waarde `data` toe. Wanneer u een item in de lijst selecteert, roept de gebeurtenislistener de functie `showData()` aan, die de waarde `data` voor het geselecteerde item weergeeft.

- 6 Selecteer Besturing > Film testen om de toepassing te compileren en uit te voeren.

Een instantie List met een gegevensaanbieder vullen

In dit voorbeeld wordt een lijst van automodellen en de prijzen hiervan gemaakt. Er wordt een gegevensaanbieder gebruikt om de lijst te vullen in plaats van de methode `addItem()`.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component List van het deelvenster Componenten naar het werkgebied.
- 3 Ga als volgt te werk in Eigenschapcontrole:
 - Voer de instantienaam **aList** in.
 - Wijs de waarde **200** aan W (breedte) toe.
- 4 Gebruik het gereedschap Tekst om een tekstveld onder **aList** te maken en geef het de instantienaam **aTf**.

- 5 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om List met zijn items te zien.

Een component List gebruiken om een instantie MovieClip te beheren

In het volgende voorbeeld wordt een lijst van kleurnamen gemaakt en wanneer u een kleur selecteert, wordt deze op een MovieClip toegepast.

- 1 Maak een Flash-document (ActionScript 3.0).
- 2 Sleep een component List van het deelvenster Componenten naar het werkgebied en voer de volgende waarden voor deze component in Eigenschapcontrole in:
 - Voer bij de instantienaam **aList in**.
 - Voer **60** in voor de waarde H.
 - Voer **100** in voor de waarde X.
 - Voer **150** in voor de waarde Y.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.
- 5 Klik op de kleuren in de lijst om ze in een MovieClip weer te geven.

Een componentinstantie List met ActionScript maken

In dit voorbeeld wordt met ActionScript een eenvoudige lijst gemaakt, waarna de lijst wordt gevuld met de methode addItem().

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component List van het deelvenster Componenten naar het deelvenster Bibliotheek.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.

De component NumericStepper gebruiken

Met de component NumericStepper kan een gebruiker een geordende reeks getallen doorlopen. De component bestaat uit een getal in een tekstvak dat wordt weergegeven naast kleine pijlknoppen omhoog of omlaag. Wanneer de gebruiker op de knoppen drukt, wordt dit getal stapsgewijs verhoogd of verlaagd op basis van de unit die is opgegeven in de parameter `stepSize` totdat de gebruiker de knoppen loslaat of totdat de maximale of minimale waarde is bereikt. De tekst in het tekstvak van de component NumericStepper is ook bewerkbaar.

Een live voorvertoning van elke instantie NumericStepper geeft de instelling van de parameter `value` weer in Eigenschapcontrole of Componentcontrole. Er is echter geen muis- of toetsenbordinteractie met de pijltoetsen van NumericStepper in de live voorvertoning.

Gebruikersinteractie met de component NumericStepper

U kunt de component NumericStepper overal gebruiken waar u wilt dat de gebruiker een numerieke waarde selecteert. U kunt een component NumericStepper bijvoorbeeld in een formulier gebruiken om de maand, dag en het jaar van de vervaldatum van een creditcard in te stellen. U kunt een component NumericStepper ook gebruiken om de gebruiker toe te staan de grootte van een lettertype te vergroten of verkleinen.

De component NumericStepper verwerkt alleen numerieke gegevens. U moet tijdens het ontwerpen ook de grootte van de stapfunctie wijzigen om meer dan twee numerieke plaatsen weer te geven (bijvoorbeeld de getallen 5246 of 1,34).

U kunt een NumericStepper in een toepassing in- of uitschakelen. In de uitgeschakelde toestand ontvangt de NumericStepper geen muis- of toetsenbordinput. Wanneer deze is uitgeschakeld ontvangt de NumericStepper focus als u erop klikt of er met Tab naartoe gaat en wordt de interne focus op het tekstvak ingesteld. Wanneer een instantie NumericStepper focus heeft, kunt u de volgende toetsen gebruiken om de instantie te beheren:

Toets	Beschrijving
Pijl-omlaag	Hiermee wordt de waarde met één unit gewijzigd.
Pijl-links	Hiermee wordt het invoegpunt binnen het tekstvak naar de linkerkant verplaatst.
Pijl-rechts	Hiermee wordt het invoegpunt binnen het tekstvak naar de rechterkant verplaatst.
Shift+Tab	Hiermee wordt de focus naar het vorige object verplaatst.
Tab	Hiermee wordt de focus naar het volgende object verplaatst.
Pijl-omhoog	Hiermee wordt de waarde met één unit gewijzigd.

Zie de klasse FocusManager in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) en “[Werken met FocusManager](#)” op pagina 30 voor meer informatie over focusbeheer.

Parameters van de component NumericStepper

U kunt in de Eigenschapcontrole of de Componentcontrole de volgende parameters instellen voor elke instantie van NumericStepper: `maximum`, `minimum`, `stepSize` en `value`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse NumericStepper in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijke waarden voor deze parameters.

Een toepassing met NumericStepper maken

De volgende procedure laat zien hoe u een component NumericStepper tijdens het ontwerpen aan een toepassing kunt toevoegen. In het voorbeeld wordt een component NumericStepper en een component Label in het werkgebied geplaatst en wordt een listener voor een gebeurtenis `Event.CHANGE` voor de instantie NumericStepper gemaakt. Wanneer de waarde in de numerieke stapfunctie wijzigt, geeft het voorbeeld de nieuwe waarde in de eigenschap `text` van de instantie Label weer.

- 1 Sleep een component NumericStepper van het deelvenster Componenten naar het werkgebied.
- 2 Voer in Eigenschapcontrole de instantienaam **aNs** in.
- 3 Sleep een component Label van het deelvenster Componenten naar het werkgebied.
- 4 Voer in Eigenschapcontrole de instantienaam **aLabel** in.
- 5 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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;
};
```

In dit voorbeeld wordt de eigenschap `text` van het label ingesteld op de waarde van de NumericStepper. De functie `changeHandler()` werkt de eigenschap `text` van het label bij op het moment dat de waarde in de instantie NumericStepper wijzigt.

- 6 Selecteer Besturing > Film testen.

Een NumericStepper maken met ActionScript

In dit voorbeeld worden drie NumericSteppers gemaakt met ActionScript-code om respectievelijk de maand, de dag en het jaar van de geboortedatum van de gebruiker in te voeren. Het voegt ook labels voor een vraag en de id's van NumericSteppers toe.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een Label naar het deelvenster Bibliotheek.
- 3 Sleep een NumericStepper naar het deelvenster Bibliotheek.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.

De component ProgressBar gebruiken

De component ProgressBar geeft de voortgang van het laden van de inhoud weer. Dit is geruuststellend voor de gebruiker wanneer de inhoud groot is en de uitvoering van de toepassing wordt vertraagd. De voortgangsbalk is nuttig voor het weergeven van de voortgang van het laden van afbeeldingen en gedeelten van een toepassing. Het laadproces kan bepaald of onbepaald zijn. Via een component Een *bepaalde* voortgangsbalk is een lineaire weergave van de voortgang van een taak in de loop van de tijd. Deze wordt gebruikt wanneer de hoeveelheid inhoud die wordt geladen bekend is. Een *onbepaalde* voortgangsbalk wordt gebruikt wanneer de hoeveelheid inhoud die moet worden geladen onbekend is. U kunt ook een component Label toevoegen om de voortgang van het laden als een percentage weer te geven.

De component ProgressBar gebruikt 9-delige schaling en heeft een skin voor een balk, een skin voor het voortgangsvak en een onbepaalde skin.

Gebruikersinteractie met de component ProgressBar

Er bestaan drie modi waarin u de component ProgressBar kunt gebruiken. De meest gebruikte modi zijn de gebeurtenismodus en de pollingmodus. Deze modi geven een laadproces op die ofwel de gebeurtenissen `progress` en `complete` verzendt (gebeurtenismodus en pollingmodus) of de eigenschappen `bytesLoaded` en `bytesTotal` beschikbaar maakt (pollingmodus). U kunt de component ProgressBar ook in de handmatige modus gebruiken door de eigenschappen `maximum`, `minimum` en `value`, en de methode `ProgressBar.setProgress()` aan te roepen. U kunt de onbepaalde eigenschap instellen om aan te geven dat de ProgressBar een gestreepte vulling en een bron van onbekende grootte (`true`) of een effen vulling en een bron van bekende grootte (`false`) heeft.

U stelt de `modus` van de ProgressBar in door de bijbehorende eigenschap `mode` in te stellen. U doet dit via de parameter `mode` in Eigenschapcontrole of Componentcontrole of via ActionScript.

Als u de ProgressBar gebruikt om een verwerkingsstatus weer te geven (zoals de verwerkingsstatus van het parseren van 100.000 items) en deze zich in een enkele framelus bevindt, zal de ProgressBar niet zichtbaar worden bijgewerkt omdat het scherm niet opnieuw wordt getekend.

Parameters van de component ProgressBar

U kunt in Eigenschapcontrole of Componentcontrole de volgende parameters instellen voor elke instantie van ProgressBar: `direction`, `mode` en `source`. Elk van deze heeft een bijbehorende ActionScript-eigenschap met dezelfde naam.

U kunt ActionScript schrijven om deze en additionele opties voor een component ProgressBar via zijn eigenschappen, methoden en gebeurtenissen te beheren. Zie de klasse ProgressBar in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

Een toepassing met ProgressBar maken

De volgende procedure laat zien hoe u een component ProgressBar tijdens het ontwerpen aan een toepassing kunt toevoegen. In dit voorbeeld gebruikt de ProgressBar de gebeurtenismodus. In de gebeurtenismodus verzendt de ladende inhoud de gebeurtenissen `progress` en `complete`, die vervolgens door de ProgressBar worden verzonden om de voortgang aan te duiden. Wanneer de gebeurtenis `progress` plaatsvindt, werkt het voorbeeld een label bij om het geladen percentage van de inhoud aan te duiden. Wanneer de gebeurtenis `complete` optreedt, geeft het voorbeeld 'Loading complete' weer en de waarde van de eigenschap `bytesTotal`, die staat voor de grootte van het bestand.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component ProgressBar van het deelvenster Componenten naar het werkgebied.
 - Voer in Eigenschapcontrole de instantienaam **aPb** in.
 - Voer in de sectie Parameters **200** in voor de waarde X.
 - Voer **260** in voor de waarde Y.
 - Selecteer `event` voor de parameter `mode`.
- 3 Sleep de component Button van het deelvenster Componenten naar het werkgebied.
 - Voer in Eigenschapcontrole de instantienaam **loadButton** in.
 - Voer **220** in voor de parameter X.
 - Voer **290** in voor de parameter Y.
 - Voer **Load Sound** in voor de parameter `label`.
- 4 Sleep een component Label naar het werkgebied en geef deze de instantienaam **progLabel**.
 - Voer **150** in voor de waarde W.
 - Voer **200** in voor de parameter X.
 - Voer **230** in voor de parameter Y.
 - Wis in de sectie Parameters de waarde voor de parameter `text`.
- 5 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe. De code laadt een MP3-audiobestand:


```
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 Selecteer Besturing > Film testen.

Een toepassing met de component **ProgressBar** maken in de pollingmodus

In het volgende voorbeeld wordt de **ProgressBar** op de pollingmodus ingesteld. In de pollingmodus wordt de voortgang bepaald door het luisteren naar de gebeurtenissen `progress` voor de inhoud die wordt geladen en met zijn eigenschappen `bytesLoaded` en `bytesTotal` wordt de voortgang berekend. In dit voorbeeld wordt een object **Sound** geladen, geluisterd naar gebeurtenissen `progress` en het geladen percentage berekend via de eigenschappen `bytesLoaded` en `bytesTotal`. Het geladen percentage wordt in een label en in het deelvenster **Uitvoer** weergegeven.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component **ProgressBar** van het deelvenster **Componenten** naar het werkgebied en voer de volgende waarden voor deze component in **Eigenschapcontrole** in:
 - Voer bij de instantienaam **aPb** in.
 - Voer **185** in voor de waarde *X*.
 - Voer **225** in voor de waarde *Y*.
- 3 Sleep een component **Label** naar het werkgebied en voer de volgende waarden voor deze component in **Eigenschapcontrole** in:
 - Voer bij de instantienaam **progLabel** in.
 - Voer **180** in voor de waarde *X*.

- Voer **180** in voor de waarde Y.
 - Wis in de sectie Parameters de waarde voor de parameter text.
- 4 Open het deelvenster Handelingen, selecteer frame 1 in de hoofdtijdlijn en voer de volgende ActionScript-code in, waarmee een object Sound wordt gemaakt (aSound) en waardoor loadSound() wordt aangeroepen om een geluid in het object Sound te laden:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.

Een toepassing met de component ProgressBar maken in de handmatige modus

In het volgende voorbeeld wordt de ProgressBar op de handmatige modus ingesteld. In de handmatige modus, moet u de voortgang handmatig instellen door de methode setProgress() aan te roepen en deze de huidige en maximale waarden geven om de mate van de voortgang te bepalen. U stelt de eigenschap source niet in de handmatige modus in. In het voorbeeld wordt een component NumericStepper gebruikt, met een maximale waarde van 250, om de ProgressBar te verhogen. Wanneer de waarde in de NumericStepper verandert en deze een gebeurtenis CHANGE activeert, roept de gebeurtenishandler (nsChangeHandler) de methode setProgress() aan om de ProgressBar naar voren te plaatsen. Het geeft op basis van de maximale waarde ook het percentage weer van de voortgang die is voltooid.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component ProgressBar van het deelvenster Componenten naar het werkgebied en voer de volgende waarden voor deze component in Eigenschapcontrole in:
 - Voer bij de instantienaam **aPb** in.
 - Voer **180** in voor de waarde X.
 - Voer **175** in voor de waarde Y.
- 3 Sleep een component NumericStepper naar het werkgebied en voer de volgende waarden voor deze component in Eigenschapcontrole in:
 - Voer bij de instantienaam **aNs** in.
 - Voer **220** in voor de waarde X.
 - Voer **215** in voor de waarde Y.

- Voer in de sectie Parameters **250** in voor de parameter maximum, **0** voor de waarde van minimum, **1** voor de parameter stepSize en **0** voor de parameter value.
- 4 Sleep een component Label naar het werkgebied en voer de volgende waarden voor deze component in Eigenschapcontrole in:
- Voer bij de instantienaam **progLabel** in.
 - Voer **150** in voor de waarde W.
 - Voer **180** in voor de waarde X.
 - Voer **120** in voor de waarde Y.
 - Wis op het tabblad Parameters de waarde Label voor de parameter text.
- 5 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voer de volgende code in:
- ```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.
- 7 Klik op de pijlknop omhoog in de NumericStepper om de ProgressBar verder te plaatsen.

### Een ProgressBar maken met ActionScript

In dit voorbeeld wordt met ActionScript een ProgressBar gemaakt. Bovendien dupliceert deze de functionaliteit van het vorige voorbeeld, die een ProgressBar in handmatige modus maakt.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component ProgressBar naar het deelvenster Bibliotheek.
- 3 Sleep de component NumericStepper naar het deelvenster Bibliotheek.
- 4 Sleep de component Label naar het deelvenster Bibliotheek.
- 5 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voer de volgende code in:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.
- 7 Klik op de pijlknop omhoog in de NumericStepper om de ProgressBar verder te plaatsen.

## De component RadioButton gebruiken

Met de component RadioButton kunt u een gebruiker dwingen één keuze te maken uit een reeks mogelijkheden. Deze component moet worden gebruikt in een groep van ten minste twee instanties RadioButton. Er kan slechts een lid van de groep tegelijk worden geselecteerd. Wanneer een keuzerondje in een groep wordt geselecteerd, wordt de huidige selectie van een keuzerondje in de groep opgeheven. U stelt de parameter `groupName` in om aan te geven tot welke groep een keuzerondje behoort.

Een keuzerondje is een fundamenteel onderdeel van vele formulieren op het web. U kunt keuzerondjes gebruiken wanneer u wilt dat de gebruiker een keuze maakt uit een groep opties. U gebruikt in een formulier bijvoorbeeld keuzerondjes voor de vraag welke creditcard de gebruiker wil gebruiken.

## Gebruikersinteractie met de component RadioButton

Een keuzerondje kan worden ingeschakeld of uitgeschakeld. Een uitgeschakeld keuzerondje kan geen muis- of toetsenbordinput ontvangen. Wanneer de gebruiker klikt op een groep met componenten RadioButton of deze met Tab selecteert, krijgt alleen het geselecteerde keuzerondje focus. De gebruiker kan de volgende toetsen gebruiken om dit te beheren:

| Toets                   | Beschrijving                                                                            |
|-------------------------|-----------------------------------------------------------------------------------------|
| Pijl-omhoog/Pijl-links  | Hiermee gaat de selectie naar het vorige keuzerondje in een groep keuzerondjes.         |
| Pijl-omlaag/Pijl-rechts | Hiermee gaat de selectie naar het volgende keuzerondje in een groep keuzerondjes.       |
| Tab                     | Hiermee wordt de focus van de groep keuzerondjes naar de volgende component verplaatst. |

Zie de IFocusManager-interface en de klasse FocusManager in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) en “[Werken met FocusManager](#)” op pagina 30 voor meer informatie over focusbeheer.

Een live voorvertoning van elke instantie RadioButton in het werkgebied geeft direct wijzigingen weer in parameters in Eigenschapcontrole of Componentcontrole tijdens het ontwerpen. De wederzijdse selectie-uitsluiting wordt echter niet in de live voorvertoning weergegeven. Als u de geselecteerde parameter voor twee keuzerondjes in dezelfde groep instelt op true, worden ze allebei als geselecteerd weergegeven ondanks het feit dat bij uitvoering alleen de laatste instantie als geselecteerd wordt weergegeven. Zie “[Parameters van de component RadioButton](#)” op pagina 81 voor meer informatie.

Wanneer u de component RadioButton aan een toepassing toevoegt, kunt u deze voor een schermlezer toegankelijk maken door de volgende coderegels toe te voegen:

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

U stelt toegankelijkheid voor een component slechts eenmaal in, ongeacht het aantal componentinstanties dat u gebruikt. Raadpleeg Hoofdstuk 18, ‘Toegankelijke inhoud maken’ in Flash gebruiken voor meer informatie.

## Parameters van de component RadioButton

U kunt in Eigenschapcontrole of Componentcontrole de volgende ontwerpparameters instellen voor elke instantie van RadioButton: `groupName`, `label`, `LabelPlacement`, `selected` en `value`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse RadioButton in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

U kunt ActionScript schrijven om additionele opties voor instanties RadioButton via de methoden, eigenschappen en gebeurtenissen van de klasse RadioButton in te stellen.

## Een toepassing met de component RadioButton maken

De volgende procedure laat zien hoe u een component RadioButton tijdens het ontwerpen aan een toepassing kunt toevoegen. In dit voorbeeld worden de RadioButtons gebruikt om een ja/nee-vraag weer te geven. De gegevens van RadioButton worden in een TextArea weergegeven.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep twee componenten RadioButton van het deelvenster Componenten naar het werkgebied.
- 3 Selecteer het eerste keuzerondje. Geef deze in Eigenschapcontrole de instantienaam **yesRb** en de groepsnaam **rbGroup**.

- 4 Selecteer het tweede keuzerondje. Geef deze in Eigenschapcontrole de instantienaam **noRb** en de groepsnaam **rbGroup**.
- 5 Sleep een component TextArea van het deelvenster Componenten naar het werkgebied en geef deze de instantienaam **aTa**.
- 6 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.

### Een RadioButton maken met ActionScript

In dit voorbeeld wordt ActionScript gebruikt om drie RadioButtons te maken voor de kleuren rood, blauw en groen en er wordt een grijs vak getekend. De eigenschap `value` voor elke RadioButton geeft de hexadecimale waarde op voor de kleur die aan de knop is gekoppeld. Wanneer de gebruiker op een van de RadioButtons klikt, roept de functie `clickHandler()` de `drawBox()` aan en wordt de kleur van de eigenschap `value` van de RadioButton doorgegeven om het vak te kleuren.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component RadioButton naar het deelvenster Bibliotheek.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.

Zie de klasse `RadioButton` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

## De component `ScrollPane` gebruiken

U kunt de component `ScrollPane` gebruiken om inhoud weer te geven die te groot is voor het gebied waarin deze inhoud wordt geladen. Als u bijvoorbeeld een grote afbeelding in een toepassing met weinig ruimte wilt plaatsen, kunt u de afbeelding in een `ScrollPane` laden. De `ScrollPane` accepteert filmclips, JPEG-, PNG-, GIF- en SWF-bestanden.

Componenten zoals de ScrollPane en de UILoader hebben gebeurtenissen `complete` die u toestaan te bepalen wanneer het laden van inhoud is voltooid. Als u eigenschappen voor de inhoud van een ScrollPane of UILoader wilt instellen, luistert u naar de gebeurtenis `complete` en stelt u de eigenschap in de gebeurtenishandler in. De volgende code maakt bijvoorbeeld een listener voor de gebeurtenis `Event.COMPLETE` en een gebeurtenishandler die de eigenschap `alpha` van de inhoud van ScrollPane op 0,5 instelt:

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

Als u een locatie opgeeft tijdens het laden van inhoud in ScrollPane, moet u de locatie (x- en y-coördinaten) 0, 0 opgeven. De volgende code laadt bijvoorbeeld de eigenschap ScrollPane omdat het vak op locatie 0, 0 is getekend:

```
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
```

Zie de klasse ScrollPane in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

## Gebruikersinteractie met de component ScrollPane

Een ScrollPane kan worden ingeschakeld of uitgeschakeld. Een uitgeschakelde ScrollPane kan geen muis- of toetsenbordinput ontvangen. De gebruiker kan de volgende toetsen gebruiken om een ScrollPane te beheren wanneer deze focus heeft:

| Toets       | Beschrijving                                                                   |
|-------------|--------------------------------------------------------------------------------|
| Pijl-omlaag | Hiermee wordt de inhoud een verticale regelschuiving omhoog verplaatst.        |
| Pijl-omhoog | Hiermee wordt de inhoud een verticale regelschuiving omlaag verplaatst.        |
| End         | Hiermee wordt de inhoud naar de onderkant van de ScrollPane verplaatst.        |
| Pijl-links  | Hiermee wordt de inhoud een horizontale regelschuiving naar rechts verplaatst. |
| Pijl-rechts | Hiermee wordt de inhoud een horizontale regelschuiving naar links verplaatst.  |
| Startpagina | Hiermee wordt de inhoud naar de bovenkant van de ScrollPane verplaatst.        |
| End         | Hiermee wordt de inhoud naar de onderkant van de ScrollPane verplaatst.        |
| PageDown    | Hiermee wordt de inhoud een verticale paginaschuiving omhoog verplaatst.       |
| PageUp      | Hiermee wordt de inhoud een verticale paginaschuiving omlaag verplaatst.       |

Een gebruiker kan de muis gebruiken om met de inhoud en de verticale en horizontale schuifbalken van ScrollPane te werken. De gebruiker kan inhoud met de muis slepen wanneer de eigenschap `scrollDrag` ingesteld staat op `true`. De weergave van een handje-cursor boven de inhoud geeft aan dat de gebruiker de inhoud kan slepen. In tegenstelling tot de meeste besturingselementen, treden handelingen op wanneer de muisknop wordt ingedrukt en deze worden uitgevoerd tot het moment waarop de knop wordt losgelaten. Als de inhoud geldige tabstops bevat, moet u `scrollDrag` instellen op `false`. Zoniet hebben alle muisklikken op de inhoud schuifslepen tot gevolg.



## Parameters van de component ScrollPane

U kunt in Eigenschapcontrole of Componentcontrole de volgende parameters instellen voor elke instantie van ScrollPane: `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `scrollDrag`, `source`, `verticalLineScrollSize`, `verticalPageScrollSize`, en `verticalScrollPolicy`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse ScrollPane in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

U kunt ActionScript schrijven om deze en additionele opties voor een component ScrollPane via zijn eigenschappen, methoden en gebeurtenissen te beheren.

## Een toepassing met de component ScrollPane maken

De volgende procedure laat zien hoe u een component ScrollPane tijdens het ontwerpen aan een toepassing kunt toevoegen. In dit voorbeeld laadt ScrollPane een afbeelding uit een pad dat is opgegeven door de eigenschap `source`.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component ScrollPane van het deelvenster Componenten naar het werkgebied en geef deze de instantienaam **aSp**.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.

## Een instantie ScrollPane met ActionScript maken

In het voorbeeld wordt een ScrollPane gemaakt, de grootte wordt ervan ingesteld en er wordt een afbeelding in geladen via de eigenschap `source`. Er worden ook twee listeners gemaakt. De eerste listener luistert naar een gebeurtenis `scroll` en geeft de positie van de afbeelding weer terwijl de gebruiker verticaal of horizontaal schuift. De tweede listener luistert naar een gebeurtenis `complete` en geeft een bericht in het deelvenster Uitvoer weer dat aangeeft dat het laden van de afbeelding voltooid is.

In dit voorbeeld wordt met ActionScript een ScrollPane gemaakt en hierin wordt een MovieClip (een rood vak) geplaatst met een breedte van 150 pixels en een hoogte van 300 pixels.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component ScrollPane van het deelvenster Componenten naar het deelvenster Bibliotheek.

- 3 Sleep de component DataGrid van het deelvenster Componenten naar het deelvenster Bibliotheek.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.

## De component Slider gebruiken

Met de component Slider kan de gebruiker een waarde selecteren door een *blokje* te verschuiven tussen de eindpunten van een vak dat overeenkomt met een waardebereik. U kunt een schuifregelaar gebruiken om de gebruiker de mogelijkheid te bieden een waarde te selecteren, zoals een aantal of een percentage. U kunt ook ActionScript gebruiken om te zorgen dat de waarde van de schuifregelaar het gedrag van een tweede object beïnvloedt. U zou de schuifregelaar bijvoorbeeld kunnen koppelen aan een afbeelding en deze verkleinen of vergroten op basis van de relatieve positie, of waarde, van het schuifregelaarblokje.

De huidige waarde van de Slider wordt bepaald door de relatieve locatie van het blokje tussen de eindpunten van het vak van de schuifregelaar, overeenkomend met de minimum- en maximumwaarden van de Slider.

De Slider maakt een continue reeks van waarden tussen de minimum- en maximumwaarden mogelijk, maar u kunt ook de parameter `snapInterval` gebruiken om intervallen tussen de minimum- en maximumwaarden op te geven. Op een Slider kunnen op bepaalde intervallen langs het schuifregelaarvak verdeelstreepjes worden weergegeven, onafhankelijk van de toegewezen waarden van de schuifregelaar.

De schuifregelaar werkt standaard in horizontale richting, maar u kunt dit wijzigen in de verticale richting door de waarde van de parameter `direction` op `vertical` in te stellen. Het schuifregelaarvak loopt van het ene uiteinde tot het andere en de verdeelstreepjes worden van links naar rechts net boven het vak geplaatst.

### Gebruikersinteractie met Slider

Wanneer een instantie Slider focus heeft, kunt u de volgende toetsen gebruiken om de instantie te beheren:

| Toets       | Beschrijving                                                                     |
|-------------|----------------------------------------------------------------------------------|
| Pijl-rechts | Hiermee wordt de gekoppelde waarde voor een horizontale schuifregelaar verhoogd. |
| Pijl-omhoog | Hiermee wordt de gekoppelde waarde voor een verticale schuifregelaar verhoogd.   |
| Pijl-links  | Hiermee wordt de gekoppelde waarde voor een horizontale schuifregelaar verlaagd. |
| Pijl-omlaag | Hiermee wordt de gekoppelde waarde voor een verticale schuifregelaar verlaagd.   |
| Shift+Tab   | Hiermee wordt de focus naar het vorige object verplaatst.                        |
| Tab         | Hiermee wordt de focus naar het volgende object verplaatst.                      |

Zie de `IFocusManager`-interface en de klasse `FocusManager` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) en “[Werken met FocusManager](#)” op pagina 30 voor meer informatie over focusbeheer.

Een live voorvertoning van elke instantie `Slider` geeft direct wijzigingen weer in parameters in Eigenschapcontrole of Componentcontrole tijdens het ontwerpen.

## Parameters van de component `Slider`

U kunt in Eigenschapcontrole of Componentcontrole de volgende ontwerpparameters instellen voor elke instantie van de component `Slider`: `direction`, `liveDragging`, `maximum`, `minimum`, `snapInterval`, `tickInterval` en `value`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse `Slider` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

## Een toepassing met `Slider` maken

In het volgende voorbeeld wordt een instantie `Slider` gemaakt waarmee de gebruiker zijn of haar mate van tevredenheid over een denkbeeldige gebeurtenis kan uitdrukken. De gebruiker verplaatst de `Slider` naar rechts of links om een hogere of lagere mate van tevredenheid aan te geven.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component `Label` van het deelvenster Componenten naar het midden van het werkgebied.
  - Geef deze de instantienaam `valueLabel`.
  - Wijs de waarde **0procent** aan de parameter `text` toe.
- 3 Sleep een component `Slider` vanuit het deelvenster Componenten en centreer deze onder `value_lbl`.
  - Geef deze de instantienaam `aSlider`.
  - Voer **200** in voor de waarde `W` (breedte).
  - Voer **10** in voor de waarde `H` (hoogte).
  - Wijs de waarde **100** aan de parameter `maximum` toe.
  - Wijs de waarde **10** aan zowel de parameter `snapInterval` als de parameter `tickInterval` toe.
- 4 Sleep nog een instantie `Label` vanuit het deelvenster Bibliotheek en centreer deze onder `aSlider`.
  - Geef deze de instantienaam `promptLabel`.
  - Voer 250 in voor de waarde `W` (breedte).
  - Voer 22 in voor de waarde `H` (hoogte).
  - Voer de tekst **Geef uw mate van tevredenheid aan** in voor de parameter `text`.

- 5 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen.

Wanneer in dit voorbeeld het blokje van de schuifregelaar van het ene interval naar het andere wordt verplaatst, werkt een listener voor de gebeurtenis `SliderEvent.CHANGE` de eigenschap `text` van `valueLabel` bij om het percentage weer te geven dat overeenkomt met de positie van het blokje.

## Met ActionScript een toepassing met de component Slider maken

In het volgende voorbeeld wordt met ActionScript een Slider gemaakt. Er wordt een afbeelding van een bloem gedownload en de Slider wordt gebruikt om de afbeelding fletser of helderder te maken door de eigenschap `alpha` aan te passen zodat deze overeenkomt met de waarde van de Slider.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component Label en de component Slider van het deelvenster Componenten naar het deelvenster Bibliotheek van het huidige document.  
  
Hiermee voegt u de componenten aan de bibliotheek toe, maar de componenten worden hierdoor niet zichtbaar in de toepassing.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende code toe om componentinstanties te maken en plaatsen:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.
- 5 Beweeg het schuifregelaarblokje naar links om de afbeelding fletser te maken en naar rechts om het helderder te maken.

## De component TextArea gebruiken

De component TextArea is een omslag voor het object TextField van ActionScript. Met de component TextArea kunt u tekst weergeven en tevens tekstinput bewerken en ontvangen wanneer de eigenschap `editable` op `true` is ingesteld. De component kan meerdere tekstregels weergeven of ontvangen en zorgt bij lange tekstregels voor tekstomloop wanneer de eigenschap `wordWrap` op `true` is ingesteld. Met de eigenschap `restrict` kunt u de tekens die de gebruiker kan gebruiken, beperken en met `maxChars` kunt u het maximum aantal tekens opgeven dat kan worden ingevoerd. Wanneer de tekst de horizontale of verticale grenzen van het tekstgebied overschrijdt, worden automatisch horizontale en verticale schuifbalken weergegeven, tenzij de bijbehorende eigenschappen `horizontalScrollPolicy` en `verticalScrollPolicy` op `off` zijn ingesteld.

U kunt een component `TextArea` gebruiken wanneer u een tekstveld met meerdere regels nodig hebt. U kunt een component `TextArea` bijvoorbeeld gebruiken als opmerkingenveld in een formulier. U zou een listener kunnen instellen die controleert of het veld leeg is wanneer de gebruiker het veld met Tab verlaat. Die listener zou dan een foutbericht kunnen weergegeven waarin wordt aangegeven dat in het veld een opmerking moet worden ingevoerd.

Gebruik de component `TextInput` wanneer u een tekstveld voor één regel tekst nodig hebt.

U kunt de stijl `textFormat` instellen met de methode `setStyle()` om de stijl te wijzigen van tekst die in een instantie `TextArea` wordt weergegeven. U kunt de tekst in een component `TextArea` ook met HTML opmaken door de eigenschap `htmlText` in ActionScript te gebruiken. Bovendien kunt u de eigenschap `displayAsPassword` op `true` instellen om tekst te maskeren met sterretjes. Wanneer u de eigenschap `condenseWhite` op `true` instelt, verwijdert Flash extra witruimte in nieuwe tekst als gevolg van spaties, regeleinden, enzovoort. Dit heeft geen effect op tekst die al in het besturingselement staat.

## Gebruikersinteractie met de component `TextArea`

Een component `TextArea` kan worden in- of uitgeschakeld in een toepassing. In de uitgeschakelde toestand kan de component geen muis- of toetsenbordinput ontvangen. Wanneer de component is ingeschakeld, volgt deze dezelfde focus-, selectie- en navigatieregels als een ActionScript-object `TextField`. Wanneer een instantie `TextArea` focus heeft, kunt u de volgende toetsen gebruiken om de instantie te beheren:

| Toets       | Beschrijving                                                                                                                        |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Pijltoetsen | Hiermee wordt het invoegpunt binnen de tekst omhoog, omlaag, naar links of naar rechts verplaatst (wanneer de tekst bewerkbaar is). |
| PageDown    | Hiermee wordt het invoegpunt naar het einde van de tekst verplaatst (wanneer de tekst bewerkbaar is).                               |
| PageUp      | Hiermee wordt het invoegpunt naar het begin van de tekst verplaatst (wanneer de tekst bewerkbaar is).                               |
| Shift+Tab   | Hiermee wordt de focus naar het vorige object in de tablus verplaatst.                                                              |
| Tab         | Hiermee wordt de focus naar het volgende object in de tablus verplaatst.                                                            |

Zie de klasse `FocusManager` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) en “[Werken met FocusManager](#)” op pagina 30 voor meer informatie over focusbeheer.

## Parameters van de component `TextArea`

U kunt in Eigenschapcontrole of Componentcontrole de volgende ontwerpparameters instellen voor elke instantie van de component `TextArea`: `condenseWhite`, `editable`, `horizontalScrollPolicy`, `maxChars`, `restrict`, `text`, `verticalScrollPolicy` en `wordwrap`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse `TextArea` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

Een live voorvertoning van elke instantie `TextArea` geeft direct wijzigingen weer in parameters in Eigenschapcontrole of Componentcontrole tijdens het ontwerpen. Wanneer een schuifbalk nodig is, wordt deze wel in de live voorvertoning weergegeven maar werkt deze niet. In de live voorvertoning is tekst niet selecteerbaar en u kunt in de componentinstantie in het werkgebied geen tekst invoeren.

U kunt ActionScript-code schrijven om deze en aanvullende opties voor de component `TextArea` te beheren met de eigenschappen, methoden en gebeurtenissen ervan. Zie de klasse `TextArea` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

## Een toepassing met de component TextArea maken

De volgende procedure laat zien hoe u een component TextArea tijdens het ontwerpen aan een toepassing kunt toevoegen. In het voorbeeld wordt een gebeurtenishandler `focusOut` op de instantie TextArea ingesteld. Die controleert of de gebruiker iets in het tekstgebied typt voordat de focus aan een ander deel van de interface wordt gegeven.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component TextArea van het deelvenster Componenten naar het werkgebied en geef deze de instantienaam **aTa**. Laat de parameters op de standaardinstellingen staan.
- 3 Sleep een tweede component TextArea van het deelvenster Componenten naar het werkgebied, plaats deze onder de eerste en geef de component de instantienaam **bTa**. Laat de parameters op de standaardinstellingen staan.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
import flash.events.FocusEvent;

aTa.restrict = "a-z, '\n' \n";
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();
}
```

In dit voorbeeld wordt het aantal tekens dat in het tekstgebied `aTa` kan worden ingevoerd, beperkt tot kleine letters, komma's, apostrofs en spaties. Bovendien worden gebeurtenishandlers voor de gebeurtenissen `change`, `KEY_FOCUS_CHANGE` en `MOUSE_FOCUS_CHANGE` op het tekstgebied `aTa` ingesteld. De functie `changeHandler()` zorgt dat de tekst die in het tekstgebied `aTa` wordt ingevoerd, automatisch in het tekstgebied `bTa` wordt weergegeven door `aTa.text` aan `bTa.text` toe te wijzen bij elke gebeurtenis `change`. De functie `k_m_fHandler()` voor de gebeurtenissen `KEY_FOCUS_CHANGE` en `MOUSE_FOCUS_CHANGE` zorgt dat de gebruiker niet met Tab naar het volgende veld kan gaan zonder eerst tekst in te voeren. Dit wordt bewerkstelligd door het standaardgedrag te voorkomen.

- 5 Selecteer Besturing > Film testen.

Wanneer u op Tab drukt om de focus naar het tweede tekstgebied te verplaatsen zonder tekst in te voeren, wordt een foutbericht weergegeven en keert de focus terug naar het eerste tekstgebied. Terwijl u in het eerste tekstgebied tekst invoert, wordt deze in het tweede tekstgebied weergegeven.

## Een instantie TextArea met ActionScript maken

In het volgende voorbeeld wordt met ActionScript een component TextArea gemaakt. De eigenschap `condenseWhite` wordt op `true` ingesteld om witruimte te verkleinen en aan de eigenschap `htmlText` wordt tekst toegewezen om de attributen voor HTML-tekstopmaak te kunnen gebruiken.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component TextArea naar het deelvenster Bibliotheek.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
import fl.controls.TextArea;

var aTa:TextArea = new TextArea();

aTa.move(100,100);
aTa.setSize(200, 200);
aTa.condenseWhite = true;
aTa.htmlText = 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. <u>Vivamus quis nisl vel tortor nonummy vulputate.</u> 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.';
addChild(aTa);
```

In dit voorbeeld wordt de eigenschap `htmlText` gebruikt om de HTML-attributen `vet` en `onderstreept` op een tekstblok toe te passen en dit in het tekstgebied `a_ta` weer te geven. Bovendien wordt de eigenschap `condenseWhite` op `true` ingesteld om de witruimte binnen het tekstblok te verkleinen. De methode `setSize()` stelt de hoogte en de breedte van het tekstgebied in en de methode `move()` stelt de positie ervan in. De methode `addChild()` voegt de instantie `TextArea` aan het werkgebied toe.

4 Selecteer Besturing > Film testen.

## De component `TextInput` gebruiken

De component `TextInput` wordt gebruikt voor tekst op één regel en omvat het native ActionScript-object `TextField`. Gebruik de component `TextArea` wanneer u een tekstveld met meerdere regels nodig hebt. U kunt een component `TextInput` bijvoorbeeld gebruiken als wachtwoordveld in een formulier. U zou ook een listener kunnen instellen die controleert of het veld voldoende tekens bevat wanneer de gebruiker het veld met `Tab` verlaat. Deze zou dan een foutbericht kunnen weergegeven waarin wordt aangegeven dat een juist aantal tekens moet worden ingevoerd.

U kunt de eigenschap `textFormat` instellen met de methode `setStyle()` om de stijl te wijzigen van tekst die in een instantie `TextInput` wordt weergegeven. Een component `TextInput` kan ook worden opgemaakt in HTML of als een wachtwoordveld dat de tekst verbergt.

## Gebruikersinteractie met de component `TextInput`

Een component `TextInput` kan worden in- of uitgeschakeld in een toepassing. In de uitgeschakelde toestand kan de component geen muis- of toetsenbordinput ontvangen. Wanneer de component is ingeschakeld, volgt deze dezelfde focus-, selectie- en navigatieregels als een ActionScript-object `TextField`. Wanneer een instantie `TextInput` focus heeft, kunt u ook de volgende toetsen gebruiken om de instantie te beheren:

| Toets       | Beschrijving                                                            |
|-------------|-------------------------------------------------------------------------|
| Pijltoetsen | Hiermee wordt het invoegpunt één teken naar links of rechts verplaatst. |
| Shift+Tab   | Hiermee wordt de focus naar het vorige object verplaatst.               |
| Tab         | Hiermee wordt de focus naar het volgende object verplaatst.             |

Zie de klasse `FocusManager` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) en “[Werken met FocusManager](#)” op pagina 30 voor meer informatie over focusbeheer.



Een live voorvertoning van elke instantie TextInput geeft direct wijzigingen weer in parameters in Eigenschapcontrole of Componentcontrole tijdens het ontwerpen. In de live voorvertoning is tekst niet selecteerbaar en u kunt in de componentinstantie in het werkgebied geen tekst invoeren.

Wanneer u de component TextInput aan een toepassing toevoegt, kunt u deze toegankelijk maken voor schermlezers via het deelvenster Toegankelijkheid.

## Parameters van de component TextInput

U kunt in Eigenschapcontrole of Componentcontrole de volgende ontwerpparameters instellen voor elke instantie van de component TextInput: `editable`, `displayAsPassword`, `maxChars`, `restrict` en `text`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie de klasse TextInput in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

U kunt ActionScript-code schrijven om deze optie en aanvullende opties voor de component TextInput te beheren met de eigenschappen, methoden en gebeurtenissen van de component. Zie de klasse TextInput in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

## Een toepassing maken met de component TextInput

De volgende procedure laat zien hoe u een component TextInput aan een toepassing kunt toevoegen. In het voorbeeld worden twee TextInput-velden gebruikt om een wachtwoord te ontvangen en te bevestigen. Er wordt gebruikgemaakt van een gebeurtenislistener om te controleren of ten minste acht tekens zijn ingevoerd en of de tekst in de twee velden overeenkomt.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component Label van het deelvenster Componenten naar het werkgebied en voer de volgende waarden voor deze component in Eigenschapcontrole in:
  - Voer de instantienaam **pwdLabel** in.
  - Voer de waarde **100** in voor W.
  - Voer de waarde **50** in voor X.
  - Voer de waarde **150** in voor Y.
  - Voer in het gedeelte Parameters de waarde **Wachtwoord:** in voor de parameter text.
- 3 Sleep een tweede component Label van het deelvenster Componenten naar het werkgebied en voer de volgende waarden voor deze component in:
  - Voer de instantienaam **confirmLabel** in.
  - Voer de waarde **100** in voor W.
  - Voer de waarde **50** in voor X.
  - Voer de waarde **200** in voor Y.
  - Voer in het gedeelte Parameters de waarde **Wachtwoord bevestigen:** in voor de parameter text.
- 4 Sleep een component TextInput van het deelvenster Componenten naar het werkgebied en voer de volgende waarden voor deze component in:
  - Voer de instantienaam **pwdTi** in.
  - Voer de waarde **150** in voor W.
  - Voer de waarde **190** in voor X.

- Voer de waarde **150** in voor Y.
  - Dubbelklik in het gedeelte Parameters op de waarde voor de parameter `displayAsPassword` en selecteer **true**. Hierdoor wordt de waarde die in het tekstveld wordt ingevoerd, gemaskeerd met sterretjes.
- 5 Sleep een tweede component `TextInput` van het deelvenster Componenten naar het werkgebied en voer de volgende waarden voor deze component in:
- Voer de instantienaam **confirmTi** in.
  - Voer de waarde **150** in voor W.
  - Voer de waarde **190** in voor X.
  - Voer de waarde **200** in voor Y.
  - Dubbelklik in het gedeelte Parameters op de waarde voor de parameter `displayAsPassword` en selecteer **true**. Hierdoor wordt de waarde die in het tekstveld wordt ingevoerd, gemaskeerd met sterretjes.
- 6 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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);
```

Met deze code wordt een gebeurtenishandler `enter` op de instantie `TextInput` met de naam `confirmTi` ingesteld. Wanneer de twee wachtwoorden niet overeenkomen of de gebruiker minder dan acht tekens invoert, wordt het volgende bericht weergegeven. “Het wachtwoord is onjuist. Voer dit nogmaals in.” Wanneer de wachtwoorden minimaal acht tekens lang zijn en overeenkomen, wordt de ingevoerde waarde in het deelvenster Uitvoer weergegeven.

- 7 Selecteer Besturing > Film testen.

## Een instantie `TextInput` met ActionScript maken

In het volgende voorbeeld wordt met ActionScript een component `TextInput` gemaakt. Er wordt bovendien een `Label` gemaakt dat wordt gebruikt om de gebruiker te vragen zijn of haar naam in te voeren. In het voorbeeld wordt de eigenschap `restrict` van de component ingesteld, zodat alleen hoofd- en kleine letters, punten en spaties worden toegestaan. Er wordt tevens een object `TextFormat` gemaakt dat wordt gebruikt om de tekst in zowel de component `Label` als de component `TextInput` op te maken.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component `TextInput` van het deelvenster Componenten naar het deelvenster Bibliotheek.
- 3 Sleep een component `Label` van het deelvenster Componenten naar het deelvenster Bibliotheek.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om de toepassing uit te voeren.

## De component TileList gebruiken

De component `TileList` bestaat uit een lijst met rijen en kolommen die worden gevuld met gegevens die afkomstig zijn van een gegevensaanbieder. Een *item* verwijst naar een gegevensseenheid die in een cel in de `TileList` wordt opgeslagen. Een item, afkomstig van de gegevensaanbieder, heeft meestal een eigenschap `label` en een eigenschap `source`. De eigenschap `label` identificeert de inhoud die in een cel moet worden weergegeven en de eigenschap `source` geeft daarvoor een waarde op.

U kunt een instantie `Array` maken of ophalen van een server. De component `TileList` heeft methoden die worden doorgegeven aan de gegevensaanbieder (bijvoorbeeld `addItem()` en `removeItem()`). Wanneer geen externe gegevensaanbieder in de lijst is opgegeven, wordt met deze methoden automatisch een gegevensaanbiederinstantie gemaakt, die via `List.dataProvider` beschikbaar wordt gemaakt.

### Gebruikersinteractie met de component TileList

Met `TileList` wordt elke cel gerenderd met een `Sprite` die de interface `ICellRenderer` implementeert. U kunt deze renderer opgeven met de eigenschap `cellRenderer` van `TileList`. De standaard-`cellRenderer` van de component `TileList` is `ImageCell`, die een afbeelding (klasse, bitmap, instantie of URL) en een optioneel label weergeeft. Het label is een enkele regel die altijd met de onderkant van de cel wordt uitgelijnd. U kunt een `TileList` slechts in een enkele richting schuiven.

Wanneer een instantie `TileList` focus heeft, kunt u ook de volgende toetsen gebruiken om items binnen de instantie te benaderen:

| Toets                      | Beschrijving                                                                                                                                                                                                                                                                                         |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pijl-omhoog en Pijl-omlaag | Hiermee kunt u in een kolom omhoog en omlaag schuiven. Wanneer de eigenschap <code>allowMultipleSelection</code> op <code>true</code> is ingesteld, kunt u deze toetsen in combinatie met Shift gebruiken om meerdere cellen te selecteren.                                                          |
| Pijl-links en Pijl-rechts  | Hiermee kunt u in een rij naar links of rechts schuiven. Wanneer de eigenschap <code>allowMultipleSelection</code> op <code>true</code> is ingesteld, kunt u deze toetsen in combinatie met Shift gebruiken om meerdere cellen te selecteren.                                                        |
| Startpagina                | Hiermee wordt de eerste cel in een <code>TileList</code> geselecteerd. Als de eigenschap <code>allowMultipleSelection</code> op <code>true</code> is ingesteld, worden door Shift vast te houden en op Home te drukken alle cellen vanaf de huidige selectie tot en met de eerste cel geselecteerd.  |
| End                        | Hiermee wordt de laatste cel in een <code>TileList</code> geselecteerd. Als de eigenschap <code>allowMultipleSelection</code> op <code>true</code> is ingesteld, worden door Shift vast te houden en op End te drukken alle cellen vanaf de huidige selectie tot en met de laatste cel geselecteerd. |
| Ctrl                       | Wanneer de eigenschap <code>allowMultipleSelection</code> op <code>true</code> is ingesteld, kunt u meerdere cellen selecteren, in willekeurige volgorde.                                                                                                                                            |

Wanneer u de component `TileList` aan een toepassing toevoegt, kunt u deze toegankelijk maken voor schermlezers door de volgende ActionScript-coderegels toe te voegen:

```
import fl.accessibility.TileListAccImpl;

TileListAccImpl.enableAccessibility();
```

U stelt toegankelijkheid voor een component slechts eenmaal in, ongeacht het aantal instanties van de component. Raadpleeg Hoofdstuk 18, 'Toegankelijke inhoud maken' in *Flash gebruiken* voor meer informatie.

## Parameters van de component `TileList`

U kunt in Eigenschapcontrole of Componentcontrole de volgende ontwerpparameters instellen voor elke instantie van de component `TileList`: `allowMultipleSelection`, `columnCount`, `columnWidth`, `dataProvider`, `direction`, `horizontalScrollLineSize`, `horizontalScrollPageSize`, `labels`, `rowCount`, `rowHeight`, `ScrollPolicy`, `verticalScrollLineSize` en `verticalScrollPageSize`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Zie "De parameter `dataProvider` gebruiken" op pagina 32 voor informatie over het gebruik van de parameter `dataProvider`.

U kunt ActionScript-code schrijven om aanvullende opties voor instanties `TileList` in te stellen met de methoden, eigenschappen en gebeurtenissen ervan. Zie de klasse `TileList` in de *Naslagids voor ActionScript 3.0 voor Adobe Flash Professional CS5* voor meer informatie.

## Een toepassing maken met de component `TileList`

In het volgende voorbeeld worden filmclips gebruikt om een `TileList` te vullen met een array aan kleuren.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component `TileList` naar het werkgebied en geef deze de instantienaam `aTL`.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om de toepassing te testen.

## Een TileList met ActionScript maken

In dit voorbeeld wordt op een dynamische manier een instantie van TileList gemaakt en worden instanties van de componenten ColorPicker, ComboBox, NumericStepper en CheckBox toegevoegd. Er wordt een Array gemaakt die labels en de namen van de componenten bevat die moeten worden weergegeven en de Array (dp) wordt aan de eigenschap dataProvider van de TileList toegewezen. De eigenschappen columnWidth en rowHeight en de methode setSize() worden gebruikt om de TileList op te maken, de methode move() wordt gebruikt om de TileList in het werkgebied te plaatsen, de stijl contentPadding wordt gebruikt om ruimte op te nemen tussen de grenzen van de instantie TileList en de inhoud ervan en de methode sortItemsOn() wordt gebruikt om de inhoud te sorteren op de labels.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de volgende componenten van het deelvenster Componenten naar het deelvenster Bibliotheek: ColorPicker, ComboBox, NumericStepper, CheckBox en TileList.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen om de toepassing te testen.

## De component UILoader gebruiken

De component UILoader is een container die SWF-, JPEG-, progressieve JPEG-, PNG- en GIF-bestanden kan weergeven. U kunt een UILoader gebruiken wanneer u inhoud vanaf een externe locatie moet ophalen en in een Flash-toepassing moet opnemen. U zou bijvoorbeeld een UILoader kunnen gebruiken om een bedrijfslogo (JPEG-bestand) aan een formulier toe te voegen. U kunt de component UILoader ook gebruiken in een toepassing die foto's weergeeft. Gebruik de methode `load()` om inhoud te laden, de eigenschap `percentLoaded` om te bepalen hoeveel inhoud is geladen en de gebeurtenis `complete` om te bepalen wanneer het laden is voltooid.

U kunt de inhoud van de UILoader schalen of de grootte van de UILoader zelf wijzigen, zodat deze bij de grootte van de inhoud past. De inhoud wordt standaard geschaald op de grootte van de UILoader. U kunt tevens inhoud laden bij uitvoering en de laadvoortgang controleren (wanneer inhoud wordt geladen nadat deze in cache is geplaatst, springt de voortgang echter snel naar 100%). Wanneer u een locatie opgeeft bij het laden van inhoud in UILoader, moet u de locatie (x- en y-coördinaten) 0, 0 opgeven.

### Gebruikersinteractie met de component UILoader

Een component UILoader kan geen focus krijgen. Inhoud die in de component UILoader is geladen kan echter wel focus accepteren en eigen focusinteracties hebben. Zie de klasse `FocusManager` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) en “[Werken met FocusManager](#)” op pagina 30 voor meer informatie over focusbeheer.

## Parameters van de component UILoader

U kunt in Eigenschapcontrole of Componentcontrole de volgende ontwerpparameters instellen voor elke instantie van de component UILoader: `autoLoad`, `maintainAspectRatio`, `source` en `scaleContent`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam.

Een live voorvertoning van elke instantie UILoader geeft direct wijzigingen weer in parameters in Eigenschapcontrole of Componentcontrole tijdens het ontwerpen.

U kunt ActionScript-code schrijven om aanvullende opties voor instanties UILoader in te stellen met de methoden, eigenschappen en gebeurtenissen ervan. Zie de klasse UILoader in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

## Een toepassing maken met de component UILoader

De volgende procedure laat zien hoe u een component UILoader tijdens het ontwerpen aan een toepassing kunt toevoegen. In dit voorbeeld laadt de UILoader een GIF-afbeelding van een logo.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component UILoader van het deelvenster Componenten naar het werkgebied.
- 3 Voer in Eigenschapcontrole de instantienaam `aUI` in.
- 4 Selecteer de UILoader in het werkgebied en in Componentcontrole en voer `http://www.helpexamples.com/images/logo.gif` in bij parameter `source`.

## Een componentinstantie UILoader met ActionScript maken

In dit voorbeeld wordt met ActionScript een component UILoader gemaakt en wordt een JPEG-afbeelding van een bloem geladen. Wanneer de gebeurtenis `complete` plaatsvindt, wordt het aantal geladen bytes in het deelvenster Uitvoer weergegeven.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component UILoader van het deelvenster Componenten naar het deelvenster Bibliotheek.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen.

## De component UIScrollBar gebruiken

Met de component UIScrollBar kunt u een schuifbalk toevoegen aan een tekstveld. U kunt met ActionScript een schuifbalk aan een tekstveld toevoegen tijdens het ontwerpen of bij uitvoering. Wanneer u de component UIScrollBar wilt gebruiken, maakt u een tekstveld in het werkgebied en sleept u de component UIScrollBar van het deelvenster Componenten naar de gewenste zijde van het selectiekader van het tekstveld.

Wanneer de lengte van de schuifbalk korter is dan de gezamenlijke grootte van de schuifpijlen, wordt deze niet goed weergegeven. Een van de pijlknoppen wordt in dat geval achter de andere verborgen. In Flash wordt hier geen foutcontrole op uitgevoerd. In een dergelijk geval is het raadzaam de schuifbalk met ActionScript te verbergen. Wanneer de grootte van de schuifbalk zodanig is dat er onvoldoende ruimte is voor het schuifblokje, wordt het schuifblokje onzichtbaar gemaakt.

De component UIScrollBar werkt als elke andere schuifbalk. Deze bevat een pijlknop aan elk uiteinde met daartussen een schuifvak met een schuifblokje. De component kan aan elke rand van een tekstveld worden gekoppeld en zowel verticaal als horizontaal worden gebruikt.

Zie de klasse TextField in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over TextField.

### Gebruikersinteractie met de component UIScrollBar

In tegenstelling tot vele andere componenten kan de component UIScrollBar continue muisinvoer ontvangen (bijvoorbeeld wanneer de gebruiker de muisknop ingedrukt houdt) in plaats van herhaalde klikhandelingen.

Er is geen toetsenbordinteractie met de component UIScrollBar.

### Parameters van de component UIScrollBar

U kunt in Eigenschapcontrole of Componentcontrole de volgende ontwerpparameters instellen voor elke instantie van de component CheckBox: `direction` en `scrollTargetName`. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam.

U kunt ActionScript-code schrijven om aanvullende opties voor instanties UIScrollBar in te stellen met de methoden, eigenschappen en gebeurtenissen ervan. Zie de klasse UIScrollBar in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

### Een toepassing met de component UIScrollBar maken

De volgende procedure laat zien hoe u een component UIScrollBar tijdens het ontwerpen aan een toepassing kunt toevoegen.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Maak een dynamisch tekstveld dat hoog genoeg is voor een of twee tekstregels en geef deze in Eigenschapcontrole de instantienaam **myText**.
- 3 Stel in Eigenschapcontrole het lijntype van het tekstinvoerveld in op Meerdere regels of Meerdere regels, geen tekstomloop wanneer u de schuifbalk horizontaal wilt gebruiken.
- 4 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe om de eigenschap `text` te vullen, zodat de gebruiker moet schuiven om alles te kunnen zien:

```
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."
```



***Opmerking:** Zorg dat het tekstveld in het werkgebied klein genoeg is om te moeten schuiven om alle tekst te kunnen zien. Als dit niet het geval is, wordt de schuifbalk niet weergegeven of wordt deze mogelijk als twee lijnen zonder blokje weergegeven (het deel dat u sleept om door de inhoud te schuiven).*

- 5 Controleer of magnetische objectuitlijning is ingeschakeld (Weergave > Magnetisch uitlijnen > Objecten magnetisch).
- 6 Sleep een instantie UIScrollBar van het deelvenster componenten naar het tekstinvuerveld in de buurt van de zijde waar u deze wilt plaatsen. De component moet met het tekstveld overlappen wanneer de muisknop wordt losgelaten, anders wordt deze niet juist aan het veld gebonden. Geef deze de instantienaam **mySb**.

De eigenschap `scrollTargetName` van de component wordt automatisch gevuld met de instantienaam van het tekstveld in Eigenschapcontrole en Componentcontrole. Wanneer deze niet op het tabblad Parameters wordt weergegeven, hebt u de instantie UIScrollBar mogelijk niet genoeg laten overlappen.

- 7 Selecteer Besturing > Film testen.

### **Een componentinstantie UIScrollBar met ActionScript maken**

U kunt met ActionScript een instantie UIScrollBar maken en deze aan een tekstveld koppelen bij uitvoering. In het volgende voorbeeld wordt een horizontale instantie UIScrollBar gemaakt en gekoppeld aan de onderzijde van een tekstveldinstantie met de naam **myTxt**, waarin tekst uit een URL wordt geladen. Bovendien wordt de grootte van de schuifbalk ingesteld, zodat deze overeenkomt met de grootte van het tekstveld.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component UIScrollBar naar het deelvenster Bibliotheek.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voeg de volgende ActionScript-code toe:

```
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 Selecteer Besturing > Film testen.

# Hoofdstuk 5: UI-componenten aanpassen

## UI-component aanpassen

U kunt de weergave van componenten in uw toepassingen aanpassen door een van de (of beide) volgende elementen te wijzigen:

**Stijlen** Elke component heeft een reeks stijlen die u kunt instellen om op te geven welke waarden Flash gebruikt om de weergave van de component te renderen. Stijlen geven doorgaans skins en pictogrammen voor een component in zijn verschillende toestanden op en ook welke waarden voor tekstmaak en opvulling moeten worden gebruikt.

**Skins** Een *skin* bestaat uit een verzameling symbolen die de grafische weergave van de component in een gegeven toestand bepalen. Een stijl geeft op welke skin wordt gebruikt en een skin is een grafisch element dat Flash gebruikt om de component te tekenen. *Skin toewijzen is het proces waarbij de weergave van een component wordt gewijzigd door het grafische element ervan te vervangen.*

**Opmerking:** *De standaardweergave van ActionScript 3.0-componenten kan worden beschouwd als een thema (Aeon Halo), maar deze skins zijn ingebouwd in de componenten. De ActionScript 3.0-componenten bieden geen ondersteuning voor de externe themabestanden die wel door ActionScript 2.0-componenten worden ondersteund.*

## Stijlen instellen

De stijlen van een component geven doorgaans waarden voor skins, pictogrammen, tekstmaak en opvulling wanneer Flash de component in zijn verschillende toestanden tekent. Flash tekent bijvoorbeeld een Button met een andere skin om de toestand Omlaag weer te geven, die optreedt wanneer u er met de muisknop op klikt, dan de skin voor de normale toestand (Omhoog). Er wordt ook een andere skin gebruikt voor de uitgeschakelde toestand die wordt veroorzaakt door de eigenschap `enabled` op `false` in te stellen.

U kunt stijlen voor componenten instellen op document-, klasse- of instantieniveau. Bovendien kunnen sommige stijleigenschappen worden overerfd van een bovenliggende component. De component List overerft bijvoorbeeld ScrollBar-stijlen van een bovenliggende BaseScrollPane.

U kunt als volgt stijlen instellen om een component aan te passen:

- Stel stijlen in op een componentinstantie. U kunt kleur- en teksteigenschappen voor een enkele componentinstantie wijzigen. Dit is in sommige gevallen effectief, maar het kan veel tijd in beslag nemen om afzonderlijke eigenschappen voor alle componenten in een document in te stellen.
- Stel stijlen in voor alle componenten van gegeven type in een document. Wanneer u een consistent uiterlijk voor alle componenten van een zelfde type wilt toepassen (bijvoorbeeld alle instanties CheckBox of Button in een document), kunt u stijlen op componentniveau instellen.

De waarden van de stijleigenschappen die op containers zijn ingesteld, worden overerfd door de componenten in de container.

Flash geeft de wijzigingen die in stijleigenschappen worden gemaakt niet weer wanneer u de componenten in het werkgebied met live voorvertoning weergeeft.

## Stijlinstellingen

Hier volgen een aantal belangrijke aanwijzingen over het gebruik van stijlen:

**Overerving** Een onderliggende component wordt ingesteld om een stijl van de bovenliggende component te overerven. U kunt binnen ActionScript geen overerving voor stijlen instellen.

**Prioriteit** Wanneer een componentstijl op meer dan één manier wordt ingesteld, gebruikt Flash de eerste stijl die het tegenkomt volgens de prioriteitsvolgorde. Flash zoekt in de volgende volgorde naar stijlen, tot het een waarde vindt:

- 1 Flash zoekt naar een stijleigenschap in de componentinstantie.
- 2 Als de stijl een van de overervende stijlen is, zoekt Flash in de bovenliggende hiërarchie naar een overerfde waarde.
- 3 Flash zoekt naar de stijl in de component.
- 4 Flash zoekt naar een algemene instelling in StyleManager.
- 5 Als de eigenschap nog steeds niet is gedefinieerd, krijgt de eigenschap de waarde `undefined`.

## De standaardstijlen van een component gebruiken

U kunt de standaardstijlen voor een component benaderen met de statische methode `getStyleDefinition()` voor de componentklasse. De volgende code haalt bijvoorbeeld de standaardstijlen voor de component `ComboBox` op en geeft de standaardwaarden voor de eigenschappen `buttonWidth` en `downArrowDownSkin` weer:

```
import fl.controls.ComboBox;
var styleObj:Object = ComboBox.getStyleDefinition();
trace(styleObj.buttonWidth); // 24
trace(styleObj.downArrowDownSkin); // ScrollArrowDown_downSkin
```

## Stijlen voor een componentinstantie instellen en ophalen

Elke instantie van een UI-component kan de methoden `setStyle()` en `getStyle()` direct aanroepen om een stijl in te stellen of op te halen. Met de volgende syntaxis stelt u een stijl en waarde voor een componentinstantie in:

```
instanceName.setStyle("styleName", value);
```

Met deze syntaxis haalt u een stijl voor een componentinstantie op:

```
var a_style:Object = new Object();
a_style = instanceName.getStyle("styleName");
```

De methode `getStyle()` retourneert het type `Object` omdat het meerdere stijlen met verschillende gegevenstypen kan retourneren. De volgende code stelt bijvoorbeeld de tekenstijl voor een instantie `TextArea` (`aTa`) in en haalt deze vervolgens op met de methode `getStyle()`. In het voorbeeld wordt de geretourneerde waarde naar een object `TextFormat` gecast om het toe te wijzen aan een variabele `TextFormat`. Zonder de cast zou de compiler een fout geven voor een poging om een variabele `Object` naar een variabele `TextFormat` te brengen.

```
import flash.text.TextFormat;

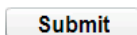
var tf:TextFormat = new TextFormat();
tf.font = "Georgia";
aTa.setStyle("textFormat", tf);
aTa.text = "Hello World!";
var aStyle:TextFormat = aTa.getStyle("textFormat") as TextFormat;
trace(aStyle.font);
```

## TextFormat gebruiken om teksteigenschappen in te stellen

Gebruik het object `TextFormat` om tekst voor een componentinstantie op te maken. Het object `TextFormat` heeft eigenschappen waarmee u teksteigenschappen kunt opgeven, zoals `bold`, `bullet`, `color`, `font`, `italic`, `size` en een aantal andere. U kunt deze eigenschappen in het object `TextFormat` instellen en vervolgens de methode `setStyle()` aanroepen om deze op een componentinstantie toe te passen. Met de volgende code worden bijvoorbeeld de eigenschappen `font`, `size` en `bold` van een object `TextFormat` ingesteld en op een instantie `Button` toegepast:

```
/* 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);
```

In het volgende afbeelding wordt het effect van deze instellingen op een verzendknop getoond:



Stijleigenschappen die met `setStyle()` op een componentinstantie zijn ingesteld, hebben de hoogste prioriteit en overschrijven alle andere stijlinstellingen. Hoe meer eigenschappen u op een enkele componentinstantie instelt met `setStyle()`, hoe langzamer de component bij uitvoering wordt gerenderd.

## Een stijl voor alle instanties van een component instellen

U kunt een stijl voor alle instanties van een componentklasse instellen met de statische methode `setComponentStyle()` van de klasse `StyleManager`. U kunt bijvoorbeeld de kleur van tekst voor alle `Buttons` op rood instellen door eerst een `Button` naar het werkgebied te slepen en vervolgens in het deelvenster `Handelingen` de volgende `ActionScript`-code aan `frame 1` van de `tijdlijn` toe te voegen:

```
import fl.managers.StyleManager;
import fl.controls.Button;

var tf:TextFormat = new TextFormat();
tf.color = 0xFF0000;
StyleManager.setComponentStyle(Button, "textFormat", tf);
```

Alle `Buttons` die u vervolgens aan het werkgebied toevoegt krijgen rode labels.

## Een stijl voor alle componenten instellen

U kunt een stijl instellen voor alle componenten met behulp van de statische methode `setStyle()` van de klasse `StyleManager`.

- 1 Sleep een component `List` naar het werkgebied en geef deze de instantienaam `aList`.
- 2 Sleep een component `Button` naar het werkgebied en geef deze de instantienaam `aButton`.
- 3 Druk op **F9** of selecteer `Handelingen` in het menu `Venster` om het deelvenster `Handelingen` te openen als het niet al open is, en voer de volgende code in `frame 1` van de `tijdlijn` in om de tekstkleur in te stellen op rood voor alle componenten:

```
import fl.managers.StyleManager;

var tf:TextFormat = new TextFormat();
tf.color = 0xFF0000;
StyleManager.setStyle("textFormat", tf);
```

- 4 Voeg de volgende code toe aan het deelvenster Handelingen om de lijst met tekst te vullen.

```
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 Selecteer Besturing > Film testen of druk op Ctrl+Enter om de code te compileren en de inhoud te testen. De tekst in zowel het knoplabel als de lijst zou nu rood moeten zijn.

## Skins

De weergave van een component wordt samengesteld van grafische elementen zoals omtrek, vulling en zelfs andere componenten. Een ComboBox, bijvoorbeeld bevat een component List en een component List bevat een UIScrollBar. De grafische elementen vormen samen de weergave van de ComboBox. De weergave van een component wisselt echter op basis van de huidige toestand. Een CheckBox ziet er bijvoorbeeld zonder label ongeveer zo uit wanneer het in uw toepassing wordt weergegeven:



*Een CheckBox in de normale toestand (Omhoog)*

Als u op de muisknop klikt en deze ingedrukt houdt op de CheckBox, verandert de weergave als volgt:



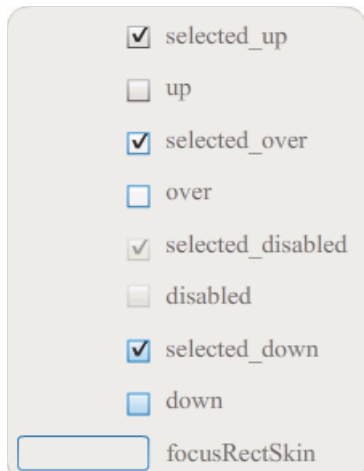
*Een CheckBox in de ingedrukte toestand (Omlaag)*

En wanneer u de muisknop loslaat, keert de CheckBox terug naar de oorspronkelijke weergave, maar nu met een vinkje om te tonen dat deze is geselecteerd.



*Een CheckBox in de geselecteerde toestand*

De pictogrammen die de component in de verschillende toestanden vertegenwoordigen, worden gezamenlijk zijn *skins genoemd*. U kunt de weergave van een component wijzigen in een of meer toestanden door de skins in Flash te bewerken zoals u dat met elk Flash-symbool kunt doen. U hebt op twee manieren toegang tot de skins van een component. De eenvoudigste manier is om het component naar het werkgebied te slepen erop te dubbelklikken. Dit opent een palet met de skins van de component, die er voor een CheckBox als volgt uitziet.



De skins van een CheckBox

U hebt toegang tot de afzonderlijke skins van een component via het deelvenster Bibliotheek. Wanneer u een component naar het werkgebied sleept, kopieert u deze ook naar de Bibliotheek, samen met een map met zijn elementen en eventuele andere componenten die het bevat. Als u een ComboBox dus naar het werkgebied sleept, bevat het deelvenster Bibliotheek ook de componenten List, ScrollBar en TextInput, die in de ComboBox zijn ingebouwd, een map met skins voor elk van deze componenten en een map Shared Assets die elementen bevat die deze componenten delen. U kunt de skins voor elk van deze componenten bewerken door zijn map skins te openen (ComboBoxSkins, ListSkins, ScrollBarSkins of TextInputSkins) en te dubbelklikken op het pictogram van de skin die u wilt bewerken. Dubbelklikken op ComboBox\_downSkin opent bijvoorbeeld de skin in symboolbewerkmodus, zoals in de volgende afbeelding wordt getoond:



De ComboBox\_downSkin

## Een nieuwe skin maken

Als u een nieuw uiterlijk voor een component in uw document wilt maken, kunt u de skins van de component bewerken om de weergave te wijzigen. U kunt dubbelklikken op het component in het werkgebied om een palet van zijn skins te openen. Vervolgens dubbelklikt u op de skin die u wilt bewerken om het te openen in symboolbewerkmodus. Dubbelklik bijvoorbeeld op de component TextArea in het werkgebied om zijn elementen in symboolbewerkmodus te openen. Stel het zoombesturingselement in op 400%, of hoger als u dat wilt, en bewerk het symbool om het uiterlijk te wijzigen. Als u klaar bent, is de wijziging van toepassing op alle instanties van de component in het document. U kunt ook dubbelklikken op een bepaalde skin in het deelvenster Bibliotheek om deze in symboolbewerkmodus te openen in het werkgebied.

U kunt de skin van een component als volgt wijzigen:

- Maak een nieuwe skin voor alle instanties
- Maak nieuwe skins voor enkele instanties

## Een skin maken voor alle instanties

Wanneer u de skin van een component bewerkt, wijzigt u standaard de weergave van een component voor alle instanties ervan in het document. Als u verschillende weergaven voor dezelfde component wilt maken, moet u de skins die u wilt wijzigen dupliceren en ze verschillende namen geven, ze bewerken en vervolgens de juiste stijlen instellen om toe te passen. Zie “[Skins maken voor enkele instanties](#)” op pagina 108 voor meer informatie.

In dit hoofdstuk wordt beschreven hoe u één of meer skins voor elk van de UI-componenten kunt wijzigen. Als u een van deze procedures volgt om één of meer skins van een UI-component te wijzigen, wijzigt u deze voor alle instanties in het document.

## Skins maken voor enkele instanties

U kunt een skin maken voor enkele instanties van een component met behulp van de volgende algemene procedure:

- Selecteer de skin in de map Elementen van de component in het deelvenster Bibliotheek.
- Dupliceer de skin en wijs er een unieke klassenaam aan toe.
- Bewerk de skin om deze de weergave te geven die u wilt.
- Roep de methode `setStyle()` aan voor de componentinstantie om de nieuwe skin aan de skinstijl toe te wijzen.

De volgende procedure maakt een nieuwe `selectedDownSkin` voor één of twee instanties `Button`.

- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep twee `Buttons` van het deelvenster Componenten naar het werkgebied en geef ze de instantienamen `aButton` en `bButton`.
- 3 Open het deelvenster Bibliotheek en open vervolgens de mappen `Component Assets` en `ButtonSkins` in het deelvenster.
- 4 Klik op de skin `selectedDownSkin` om deze te selecteren.
- 5 Klik met de rechtermuisknop om het contextmenu te openen en selecteer `Dupliceren`.
- 6 Geef de nieuwe skin een unieke naam in het dialoogvenster `Symbool dupliceren`, bijvoorbeeld `Button_mySelectedDownSkin`. Klik vervolgens op `OK`.
- 7 Selecteer `Button_mySelectedDownSkin` in `Bibliotheek > Component Assets > ButtonSkins` en klik met de rechtermuisknop om het contextmenu te openen. Selecteer `Koppeling` om het dialoogvenster `Koppelingseigenschappen` te openen.
- 8 Schakel het selectievakje `Exporteren voor ActionScript` in. Houdt het selectievakje `Exporteren` in eerste frame geselecteerd en zorg dat de klassenaam uniek is. Klik op `OK` en nogmaals op `OK` bij de waarschuwing dat geen klassedefinitie kan worden gevonden en dat er een wordt gemaakt.
- 9 Dubbelklik op de skin `Button_mySelectedDownSkin` in het deelvenster `Bibliotheek` om deze in `symboolbewerkmodus` te openen.
- 10 Klik op de blauwe vulling in het midden van de skin tot de kleur in de `kleurkiezer` voor de `vulkleur` in `Eigenschapcontrole` wordt weergegeven. Klik op de `kleurkiezer` en selecteer `kleur #00CC00` voor de `skinvulling`.
- 11 Klik op de knop `Terug` links van de `bewerkbalk` boven in het werkgebied om terug te gaan naar de `documentbewerkmodus`.
- 12 Klik in de `Eigenschapcontrole` op het tabblad `Parameters` voor elke knop en stel de `schakelparameter` in op `true`.
- 13 Voeg de volgende code toe aan het deelvenster `Handelingen` in frame 1 van de `tijdlijn`:

```
bButton.setStyle("selectedDownSkin", Button_mySelectedDownSkin);
bButton.setStyle("downSkin", Button_mySelectedDownSkin);
```



14 Selecteer Besturing > Film testen.

15 Klik op elke knop De skin voor Omlaag (geselecteerd en niet geselecteerd) voor het object `bButton` gebruikt het nieuwe skinsymbool.

## De component Button aanpassen

U kunt een component Button horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of elke toepasselijke eigenschap van de klasse Button gebruiken, zoals `height` en `width`, `scaleX` en `scaleY`.

Het wijzigen van de grootte van de knop verandert niets aan de grootte van het pictogram of label. Het selectiekader van een Button komt overeen met de rand van de Button en geeft ook de raakgebied aan voor de instantie. Wanneer u de grootte van de instantie vergroot, vergroot u ook de grootte van het raakgebied. Wanneer het selectiekader te klein is voor het label, wordt het label op maat geknipt.

Als de Button een pictogram heeft dat groter is dan de Button, steekt het pictogram buiten de randen van de knop uit.

## Stijlen gebruiken met de component Button

De stijlen van een Button geven over het algemeen waarden op voor zijn skins, pictogrammen, tekstopmaak en opvulling wanneer Flash de component in zijn verschillende toestanden tekent.

De volgende procedure plaatst twee Buttons in het werkgebied en stelt de eigenschap `emphasized` in op waar voor beide Buttons als de gebruiker op één ervan klikt. Het stelt ook de stijl `emphasizedSkin` voor de tweede Button in op de stijl `selectedOverSkin` wanneer de gebruiker erop klikt, zodat de twee Buttons verschillende skins laten zien in dezelfde toestand.

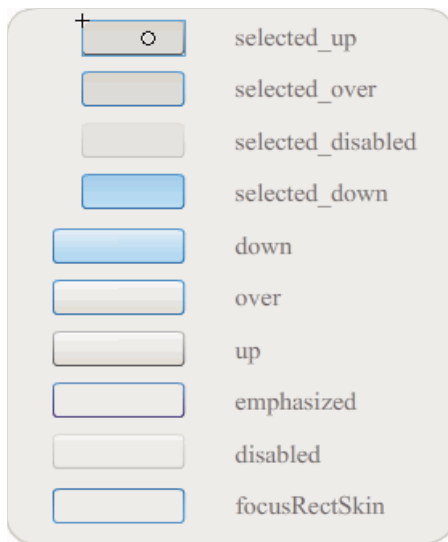
- 1 Maak een Flash-bestand (ActionScript 3.0).
- 2 Sleep twee Buttons één voor één naar het werkgebied en geef ze de instantienamen **aBtn** en **bBtn**. Geef ze op het tabblad Parameters van Eigenschapcontrole de labels Button A en Button B.
- 3 Voeg de volgende code toe aan het deelvenster Handelingen in frame 1 van de tijdlijn:

```
bBtn.emphasized = true;
aBtn.emphasized = true;
bBtn.addEventListener(MouseEvent.CLICK, Btn_handler);
function Btn_handler(evt:MouseEvent):void {
 bBtn.setStyle("emphasizedSkin", "Button_selectedOverSkin");
}
```

- 4 Selecteer Besturing > Film testen.
- 5 Klik op één van de knoppen om het effect van de stijl `emphasizedSkin` op elke knop te zien.

## Skins gebruiken met de component Button

De component Button gebruikt de volgende skins die overeenkomen met zijn verschillende toestanden. Wanneer u een of meer skins wilt bewerken om de weergave van de Button te wijzigen, dubbelklikt u op de instantie Button in het werkgebied om een palet van zijn skins te openen, zoals in de volgende afbeelding wordt getoond:

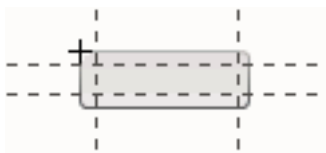


*Skins van Buttons*

Als een knop is ingeschakeld, geeft het de toestand Boven weer, wanneer de aanwijzer erboven beweegt. De knop ontvangt invoerfocus en geeft de toestand Omlaag weer wanneer deze wordt ingedrukt. De knop keert terug naar de toestand Boven wanneer de muis wordt losgelaten. Wanneer de aanwijzer van de knop af beweegt, wanneer de muis is ingedrukt, keert de knop terug naar de oorspronkelijke toestand. Als de schakelparameter is ingesteld op `true`, wordt de toestand Omlaag weergegeven met de `selectedDownSkin`, de toestand Omhoog met de `selectedUpSkin` en de toestand Boven met de `selectedOverSkin`.

Wanneer een knop is uitgeschakeld, wordt de uitgeschakelde toestand weergegeven, ongeacht gebruikersinteractie.

Wanneer u een van de skins wilt bewerken, kunt u erop dubbelklikken om deze te openen in symboolbewerkmodus, zoals in de volgende afbeelding wordt getoond:



*Knop in symboolbewerkmodus*

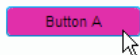
Op dit punt kunt u Flash-ontwerpgereedschap gebruiken om de skin naar believen te bewerken.

De volgende procedure wijzigt de kleur van de skin van de Button `selected_over`.

- 1 Maak een nieuw Flash-bestand (ActionScript 3.0).
- 2 Sleep een Button van het deelvenster Componenten naar het werkgebied. Stel in het tabblad Parameters de schakelparameter in op `waar`.
- 3 Dubbelklik op de Button om het palet van zijn skins te openen.
- 4 Dubbelklik op de skin `selected_over` om het te openen in symboolbewerkmodus.
- 5 Stel het zoombesturingselement op 400% in om het pictogram te vergroten voor bewerken.
- 6 Dubbelklik op de achtergrond tot de kleur ervan in de vulkleurkiezer in Eigenschapcontrole wordt weergegeven.
- 7 Selecteer kleur `#CC0099` in de kleurkiezer voor de vulkleur om deze toe te passen op de achtergrond van de skin `selected_over`.

- 8 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbewerkmodus.
- 9 Selecteer Besturing > Film testen.
- 10 Klik op de knop om deze in de geselecteerde toestand te plaatsen.

Wanneer u de muisaanwijzer boven de Button plaatst, wordt de toestand `selected_over` weergegeven, zoals in de onderstaande afbeelding.



*Knop met skin `selected_over` met gewijzigde kleur*

## De component CheckBox aanpassen

U kunt een component `CheckBox` horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of toepasbare eigenschappen van de klasse `CheckBox` gebruiken. U kunt bijvoorbeeld de grootte van een `CheckBox` wijzigen door de eigenschappen `height` en `width`, `scaleX` en `scaleY` ervan in te stellen. Het wijzigen van de grootte van de `CheckBox` verandert niets aan de grootte van het label of selectievakpictogram, het wijzigt alleen de grootte van het selectiekader.

Het selectiekader van een instantie `CheckBox` is onzichtbaar en geeft ook het raakgebied aan voor de instantie. Wanneer u de grootte van de instantie vergroot, vergroot u ook de grootte van het raakgebied. Wanneer het selectiekader te klein is voor het label, wordt het label op maat geknipt.

### Stijlen gebruiken met de CheckBox

U kunt stijleigenschappen instellen om de weergave van een instantie `CheckBox` te wijzigen. De volgende procedure wijzigt bijvoorbeeld de grootte en kleur van een label van een `CheckBox`.

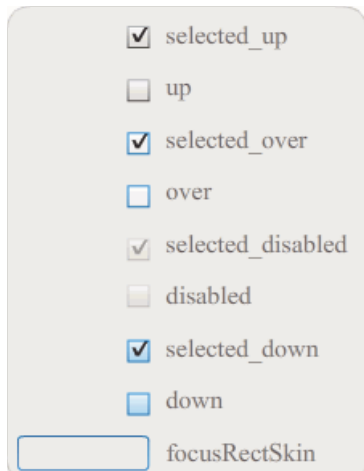
- 1 Sleep de component `CheckBox` van het deelvenster Componenten naar het werkgebied en geef deze de instantienaam `myCb`.
- 2 Klik op het tabblad Parameters in Eigenschapcontrole en voer de volgende waarde in voor de labelparameter:  
**Minder dan \$500?**
- 3 Voer in frame 1 van de hoofdtijdlijn de volgende code in het deelvenster Handelingen in:

```
var myTf:TextFormat = new TextFormat();
myCb.setSize(150, 22);
myTf.size = 16;
myTf.color = 0xFF0000;
myCb.setStyle("textFormat", myTf);
```

Zie “[Stijlen instellen](#)” op pagina 103 voor meer informatie. Zie “[Een nieuwe skin maken](#)” op pagina 107 en “[Skins gebruiken met de CheckBox](#)” op pagina 111 voor informatie over het instellen van stijleigenschappen om de pictogrammen en skins van de component te wijzigen.

### Skins gebruiken met de CheckBox

De component `CheckBox` heeft de volgende skins, die u kunt bewerken om de weergave ervan te wijzigen.



*Skins van een CheckBox*

In dit voorbeeld wordt de contour- en achtergrondkleur gewijzigd van de component in de toestanden `up` en `selectedUp`. U kunt dezelfde stappen volgen om de skins voor andere toestanden te wijzigen.

- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep de component `CheckBox` naar het werkgebied, waarmee het ook in de Bibliotheek wordt geplaatst met een map met zijn elementen.
- 3 Dubbelklik op de component `CheckBox` in het werkgebied om het deelvenster met skinpictogrammen te openen.
- 4 Dubbelklik op het pictogram `selected_up` om het te openen in symboolbewerkmodus.
- 5 Stel het zoombesturingselement op 800% in om het pictogram te vergroten voor bewerken.
- 6 Klik op de rand van de `CheckBox` om deze te selecteren. Gebruik de kleurkiezer voor de vulkleur in Eigenschapcontrole om de kleur `#0033FF` te selecteren en deze op de rand toe te passen.
- 7 Dubbelklik op de achtergrond van de `CheckBox` om deze te selecteren en gebruik de kleurkiezer voor de vulkleur opnieuw om de kleur van de achtergrond in te stellen op `#00CCFF`.
- 8 Herhaal stappen 4 t/m 8 voor de skin `Up` van `CheckBox`.
- 9 Selecteer Besturing > Film testen.

## De component `ColorPicker` aanpassen

U kunt het formaat van een `ColorPicker` alleen wijzigen via de stijlen: `swatchWidth`, `swatchHeight`, `backgroundPadding`, `textFieldWidth` en `textFieldHeight`. Als u probeert de grootte van de `ColorPicker` te wijzigen met Transformeren of met ActionScript door middel van de methode `setSize()` of de eigenschappen `width`, `height`, `scaleX` of `scaleY`, dan worden deze waarden genegeerd wanneer u het SWF-bestand maakt, en wordt de `ColorPicker` weergegeven op standaardgrootte. De achtergrond van het palet wijzigt van grootte om overeen te komen met het aantal kolommen dat is ingesteld met `setStyle()` voor de stijl `columnCount`. Het standaardaantal kolommen is 18. U kunt aangepaste kleuren instellen tot 1024 en het palet past zijn grootte verticaal aan om overeen te komen met het aantal stalen.

## Stijlen gebruiken met de component ColorPicker

U kunt verschillende stijlen instellen om de weergave van de component ColorPicker te wijzigen. De volgende procedure wijzigt bijvoorbeeld het aantal kolommen (`columnCount`) in de ColorPicker naar 12, wijzigt de hoogte (`swatchHeight`) en breedte (`swatchWidth`) van de kleurstalen en wijzigt de opvulling voor het tekstveld (`textPadding`) en de achtergrond (`backgroundPadding`).

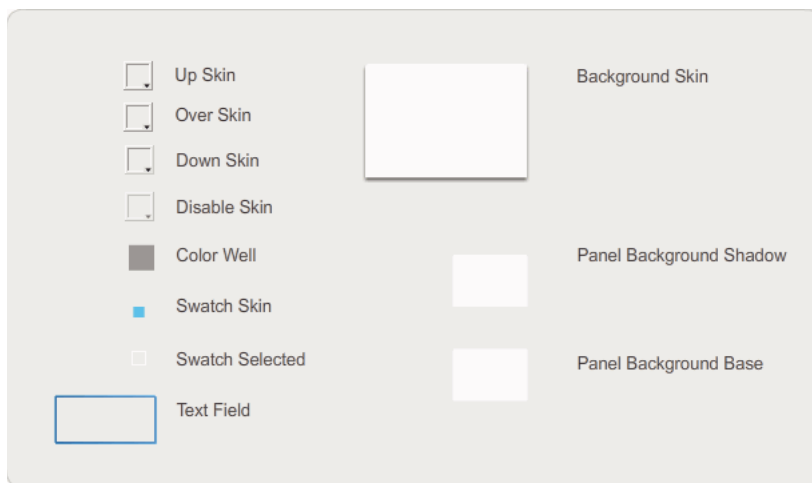
- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep een component ColorPicker naar het werkgebied en geef deze de instantienaam **aCp**.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voer de volgende code in:

```
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 Selecteer Besturing > Film testen.
- 5 Klik op de ColorPicker om deze te openen en kijk hoe deze instellingen de weergave hebben veranderd.

## Skins gebruiken met de component ColorPicker

De component ColorPicker gebruikt de volgende skins om zijn visuele status te vertegenwoordigen.



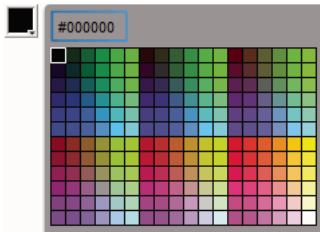
*Skins voor een ColorPicker*

U kunt de kleur van de skin Background wijzigen om de kleur van de achtergrond van het palet te wijzigen.

- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep de component ColorPicker naar het werkgebied.
- 3 Dubbelklik erop om het palet met skins te openen.
- 4 Dubbelklik op de skin Background totdat deze is geselecteerd en de kleurkiezer voor de vulkleur wordt weergegeven in Eigenschapcontrole.
- 5 Selecteer kleur #999999 met behulp van de vulkleurkiezer om het op de skin Background toe te passen.

- 6 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbewerkmodus.
- 7 Selecteer Besturing > Film testen.

Wanneer u op de ColorPicker klikt, zou de achtergrond van het palet grijs moeten zijn, zoals in de onderstaande afbeelding wordt getoond.



*ColorPicker met donkergrijze skin Background*

## De component ComboBox aanpassen

U kunt een component ComboBox horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of de toepasbare eigenschappen van de klasse `ComboBox` gebruiken, zoals `height` en `width`, `scaleX` en `scaleY`.

De ComboBox verandert van grootte zodat de opgegeven hoogte en breedte passen. De lijst verandert van grootte om overeen te komen met de breedte van de component, tenzij de eigenschap `dropdownWidth` is ingesteld.

Wanneer de tekst te lang is voor de ComboBox, wordt de tekst op maat geknipt. U moet de grootte van de ComboBox aanpassen en de eigenschap `dropdownWidth` instellen zodat de tekst past.

### Stijlen gebruiken met de component ComboBox

U kunt stijleigenschappen instellen om de weergave van een component ComboBox te wijzigen. De stijlen geven waarden op voor de skins, celrenderer, opvulling en knopbreedte van de component. In het volgende voorbeeld worden de stijlen `buttonWidth` en `textPadding` ingesteld. De stijl `buttonWidth` stelt de breedte van het raakgebied van de knop in en wordt gebruikt wanneer de ComboBox bewerkbaar is en u alleen op de knop kunt drukken om de vervolgkeuzelijst te openen. De stijl `textPadding` geeft de hoeveelheid ruimte op tussen de buitenste randen van het tekstveld en de tekst. Het is nuttig voor het verticaal centreren van de tekst in het tekstveld wanneer u de ComboBox hoger maakt. Anders zou de tekst bovenin het tekstveld worden weergegeven.

- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep een component ComboBox naar het werkgebied en geef deze de instantienaam **aCb**.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voer de volgende code in:

```
import fl.data.DataProvider;

aCb.setSize(150, 35);
aCb.setStyle("textPadding", 10);
aCb.setStyle("buttonWidth", 10);
aCb.editable = true;

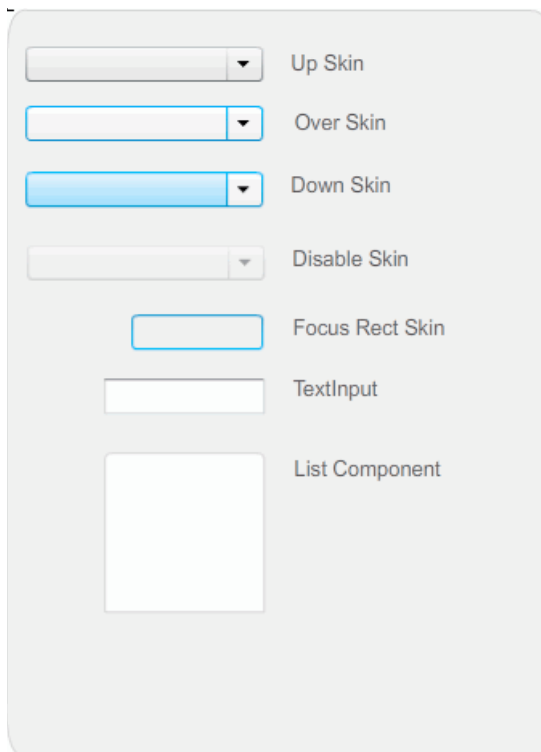
var items:Array = [
 {label:"San Francisco", data:"601 Townsend St."},
 {label:"San Jose", data:"345 Park Ave."},
 {label:"San Diego", data:"10590 West Ocean Air Drive, Suite 100"},
 {label:"Santa Rosa", data:"2235 Mercury Way, Suite 105"},
 {label:"San Luis Obispo", data:"3220 South Higuera Street, Suite 311"}
];
aCb.dataProvider = new DataProvider(items);
```

#### 4 Selecteer Besturing > Film testen.

Het gebied van de knop waarop u kunt klikken om de vervolgkeuzelijst te openen is slechts een smal gedeelte aan de rechterkant. De tekst is verticaal gecentreerd in het tekstveld. U kunt het voorbeeld testen zonder de twee instructies `setStyle()` om hun effect te zien.

## Skins gebruiken met de ComboBox

De ComboBox gebruikt de volgende skins om zijn visuele status te vertegenwoordigen.



*Skins van een ComboBox*

U kunt de kleur van de skin Up wijzigen om de kleur van de component in de inactieve toestand in het werkgebied te wijzigen.

- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep de component ComboBox naar het werkgebied.
- 3 Dubbelklik erop om het palet met skins te openen.
- 4 Dubbelklik op de skin Up tot deze is geselecteerd en open voor bewerking.
- 5 Stel het zoombesturingselement in op 400%.
- 6 Klik op het middengebied van de skin tot de kleur ervan in de kleurkiezer voor de vulkleur in Eigenschapcontrole wordt weergegeven.
- 7 Selecteer kleur #33FF99 met behulp van de kleurkiezer voor de vulkleur om het op de skin Up toe te passen.
- 8 Klik op de knop Terug links van de werkbalk boven in het werkgebied om terug te gaan naar de documentbewerkmodus.
- 9 Selecteer Besturing > Film testen.

De ComboBox wordt weergegeven in het werkgebied, zoals in onderstaande afbeelding wordt getoond.



*ComboBox met aangepaste kleur voor skin Background*

## De component DataGrid aanpassen

U kunt een component DataGrid horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of de toepasselijke eigenschappen gebruiken, zoals `width`, `height`, `scaleX` en `scaleY`. Als er geen horizontale schuifbalk is, wordt de kolombreedte proportioneel aangepast. Wanneer de grootte van een kolom (en dus cel) wordt aangepast, kan de tekst in de cellen worden afgekapt.

### Stijlen gebruiken met de component DataGrid

U kunt stijleigenschappen instellen om de weergave van een component DataGrid te wijzigen. De component DataGrid overerft stijlen van de component List. (Zie “[Stijlen gebruiken met de component List](#)” op pagina 122.)

#### Stijlen instellen voor een afzonderlijke kolom

Een object DataGrid kan meerdere kolommen bevatten en u kunt verschillende celrenderers opgeven voor elke kolom. Elke kolom van een DataGrid wordt vertegenwoordigd door een object DataGridColumn en de klasse DataGridColumn bevat de eigenschap `cellRenderer` waarvoor u de CellRenderer voor de kolom kunt definiëren.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component DataGrid naar het deelvenster Bibliotheek.
- 3 Voeg de volgende code toe aan het deelvenster Handelingen in frame 1 van de tijdlijn. Met deze code wordt een DataGrid gemaakt met een lange reeks tekst in de derde kolom. Uiteindelijk wordt de eigenschap `cellRenderer` van de kolom ingesteld op de naam van een celrenderer die een cel van meerdere regels rendeert.



```
/* 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 Sla het FLA-bestand op als MultiLineGrid fla.
- 5 Maak een nieuw ActionScript-bestand.
- 6 Kopieer de volgende ActionScript-code in het Script-venster:

```
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 Sla het ActionScript-bestand op als MultiLineCell.as in dezelfde map als u de MultiLineGrid.fla hebt opgeslagen.
- 8 Keer terug naar de toepassing MultiLineGrid.fla en selecteer Besturing > Film Testen.

De DataGrid moet er als volgt uitzien:

| 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  |

DataGrid voor de toepassing MultiLineGrid.fla

## Kopstijlen instellen

U kunt de tekststijl voor een koprij instellen met behulp van de stijl `headerTextFormat`. In het volgende voorbeeld wordt het object `TextFormat` gebruikt om de stijl `headerTextFormat` in te stellen op het lettertype Arial, de kleur rood, tekengrootte 14 en cursief.

- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep een component DataGrid naar het werkgebied en geef deze de instantienaam `aDg`.
- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voer de volgende code in:

```
import fl.data.DataProvider;
import fl.controls.dataGridClasses.DataGridColumn;

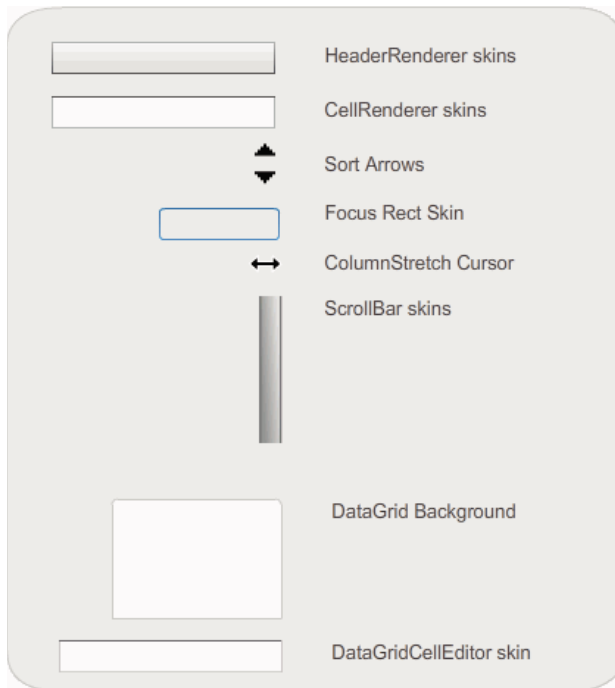
var myDP:Array = new Array();
myDP = [{FirstName:"Winston", LastName:"Elstad"},
 {FirstName:"Ric", LastName:"Dietrich"},
 {FirstName:"Ewing", LastName:"Canepa"},
 {FirstName:"Kevin", LastName:"Wade"},
 {FirstName:"Kimberly", LastName:"Dietrich"},
 {FirstName:"AJ", LastName:"Bilow"},
 {FirstName:"Chuck", 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 Selecteer Besturing > Film testen om de toepassing uit te voeren.

## Skins gebruiken met de component DataGrid

De component DataGrid gebruikt de volgende skins om zijn visuele status te vertegenwoordigen:



*Skins voor een DataGrid*

De skin CellRenderer is de skin die wordt gebruikt voor de cel met de hoofdtekst van de DataGrid, terwijl de skin HeaderRenderer wordt gebruikt voor de koprij. De volgende procedure wijzigt de achtergrondkleur van de koprij maar u kunt hetzelfde proces gebruiken om de achtergrondkleur van de cel van de hoofdtekst van de DataGrid wijzigen door de skin CellRenderer te bewerken.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component DataGrid naar het werkgebied en geef deze de instantienaam **aDg**.
- 3 Dubbelklik op de component om het palet met skins te openen.
- 4 Stel het zoombesturingselement in op 400% om de pictogrammen te vergroten voor bewerking.
- 5 Dubbelklik op de skin HeaderRenderer om het palet van skins HeaderRenderer te openen.
- 6 Dubbelklik op Up\_Skin om deze in de symboolbepijning te openen en klik op de achtergrond totdat deze is geselecteerd en de kleurkiezer voor de vulkleur wordt weergegeven in Eigenschapcontrole.
- 7 Selecteer kleur #00CC00 in de kleurkiezer voor de vulkleur om deze toe te passen op de achtergrond van de skin Up\_Skin van HeaderRenderer.
- 8 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbepijning.
- 9 Voeg de volgende code in het deelvenster Handelingen toe aan frame 1 van de tijdslijn om gegevens aan de DataGrid toe te voegen:

```
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 Selecteer Besturing > Film testen om de toepassing te testen.

De DataGrid wordt weergegeven met een groene achtergrond in de koprij, zoals in de volgende afbeelding wordt getoond.

| Name            | Home         |
|-----------------|--------------|
| Wilma Carter    | Redlands, CA |
| Sue Pennypacker | Athens, GA   |
| Jill Smithfield | Spokane, WA  |
| Shirley Goth    | Carson, NV   |
| Jennifer Dunbar | Seaside, CA  |

*DataGrid met aangepaste achtergrond van de koprij*

## De component Label aanpassen

U kunt een component Label horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. U kunt ook de ontwerpparameter `autoSize` instellen; het instellen van deze parameter verandert het selectiekader niet in de live voorvertoning, maar de grootte van het Label wordt gewijzigd. De grootte van het Label wordt gewijzigd op de parameter `wordWrap`. Als de parameter `true` is, wordt de grootte van het Label verticaal gewijzigd tot de tekst erin past. Is de parameter `false`, dan wordt de grootte van het Label horizontaal aangepast. Gebruik de methode `setSize()` bij uitvoering. Zie de methode `Label.setSize()` en de eigenschap `Label.autoSize` in de *Naslaggids voor ActionScript® 3.0 voor Adobe® Flash® Professional CS5*. Zie ook “[Een toepassing met de component Label maken](#)” op pagina 66.

## Stijlen gebruiken met de component Label

U kunt stijleigenschappen instellen om de weergave van een labelinstantie te wijzigen. Alle tekst in een componentinstantie Label moet dezelfde stijl gebruiken. De component Label heeft een stijl `textFormat`, die dezelfde attributen heeft als het object `TextFormat`, waarmee u dezelfde eigenschappen voor de inhoud van `Label.text` kunt instellen als voor een normaal Flash `TextField`. In het volgende voorbeeld wordt de kleur van de tekst in een label ingesteld op rood.

- 1 Sleep de component Label van het deelvenster Componenten naar het werkgebied en geef deze de instantienaam `a_label`.
- 2 Klik op het tabblad Parameters en vervang de waarde van de teksteigenschap door tekst.

### Kleur me rood

- 3 Open het deelvenster Handelingen, selecteer frame 1 van de hoofdtijdlijn en voer de volgende code in:

```
/* 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 Selecteer Besturing > Film testen.

Voor meer informatie over Label-stijlen raadpleegt u de Label-klasse in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#).

## Skins en de Label

De component Label heeft geen visuele elementen waarop een skin kan worden toegepast.

## De component List aanpassen

U kunt een component List horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of de toepasbare eigenschappen van de klasse List gebruiken, zoals `height` en `width`, `scaleX` en `scaleY`.

Wanneer de grootte van een lijst wordt gewijzigd, worden de rijen horizontaal kleiner, waardoor de tekst erin wordt afgekapt. De lijst voegt rijen toe of verwijdert deze, waar nodig. Schuifbalken worden, indien nodig, automatisch geplaatst.

## Stijlen gebruiken met de component List

U kunt stijleigenschappen instellen om de weergave van een component List te wijzigen. De stijlen geven waarden op voor de skins en opvulling van de component wanneer de component wordt getekend.

Met de verschillende stijlen van skins kunt u verschillende klassen opgeven die de skin kan gebruiken. Zie “Skins” op pagina 106 voor meer informatie over het gebruik van skinstijlen.

In de volgende procedure wordt de waarde van de stijl `contentPadding` voor de component `List` ingesteld. De waarde van deze instelling wordt afgetrokken van de grootte van de `List` om de opvulling rond de inhoud te bereiken, die u misschien nodig hebt om de grootte van de `List` vergroten om te voorkomen dat de tekst in de `List` wordt afgekapt.

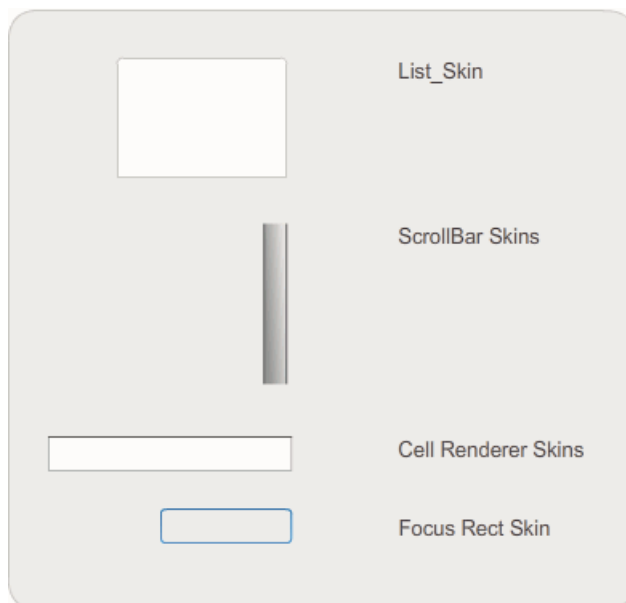
- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep de component `List` van het deelvenster Componenten naar het werkgebied en geef deze de instantienaam **aList**.
- 3 Selecteer frame 1 in de hoofdtijdlijn, open het deelvenster Handelingen en voer de volgende code in, waarmee de stijl `contentPadding` wordt ingesteld en gegevens aan de `List` worden toegevoegd:

```
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 Selecteer Besturing > Film testen.

## Skins gebruiken met de component `List`

De component `List` gebruikt de volgende skins om zijn visuele status te vertegenwoordigen:

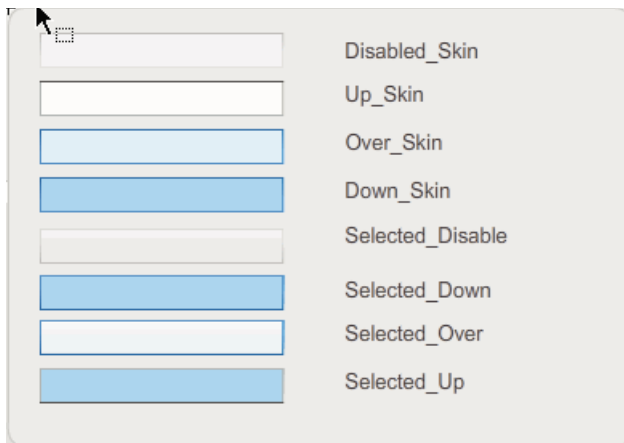


*Skins voor een List*

Zie “[De component `UIScrollBar` aanpassen](#)” op pagina 138 voor meer informatie over het toewijzen van een skin aan de `ScrollBar`. Zie “[De component `TextArea` aanpassen](#)” op pagina 132 voor meer informatie over het toewijzen van de skin `Focus Rect`.

**Opmerking:** Het wijzigen van de skin `ScrollBar` in een component wijzigt deze voor alle andere componenten die de `ScrollBar` gebruiken.

Dubbelklik op de skin `Cell Renderer` om een tweede palet met skins te openen voor de andere toestanden van een cel in een `List`.



*Skins voor een List Cell Renderer*

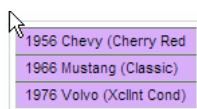
U kunt de weergave van de cellen in de List wijzigen door deze skins te bewerken. In de volgende procedure wordt de kleur van de skin Up gewijzigd om de weergave van de List in de normale inactieve toestand te wijzigen.

- 1 Maak een nieuw document van het type Flash-bestand (ActionScript 3.0).
- 2 Sleep de component List van het deelvenster Componenten naar het werkgebied en geef deze de instantienaam **aList**.
- 3 Dubbelklik op List om het palet met skins te openen.
- 4 Dubbelklik op de skin Cell Renderer om het palet met skins van Cell Renderer te openen.
- 5 Dubbelklik op de skin Up\_Skin om deze te openen voor bewerking.
- 6 Klik op het vulgebied van de skin om dit te selecteren. Een kleurkiezer voor de vulkleur wordt weergegeven in Eigenschapcontrole met de huidige vulkleur van de skin.
- 7 Selecteer kleur #CC66FF in de kleurkiezer voor de vulkleur om deze toe te passen op de vulling van de skin Up\_Skin.
- 8 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbewerkmodus.
- 9 Voeg de volgende code in het deelvenster Handelingen toe aan frame 1 van de tijdlijn om gegevens aan de List toe te voegen:

```
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 Selecteer Besturing > Film testen.

De List wordt weergegeven als in de volgende afbeelding:



*Cellen List met aangepaste kleur Up\_Skin*



Het kader is het resultaat van het instellen van de stijl `contentPadding`.

## De component `NumericStepper` aanpassen

U kunt een component `NumericStepper` horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of alle toepasselijke eigenschappen van de klasse `NumericStepper` gebruiken, zoals `width`, `height`, `scaleX` en `scaleY`.

Het wijzigen van de grootte van de component `NumericStepper` verandert de breedte van de pijltoetsen omhoog en omlaag niet. Als de grootte van de stapfunctie groter wordt dan de standaardhoogte, zet het standaardgedrag de pijltoetsen vast aan de boven- en onderkant van de component. Anders bepaalt 9-delige schaling hoe de knoppen worden getekend. De pijlknoppen worden altijd rechts van het tekstvak weergegeven.

### Stijlen en de component `NumericStepper`

U kunt de stijleigenschappen van de component `NumericStepper` instellen om de weergave te wijzigen. De stijlen geven waarden op voor de skins, opvulling en tekstopmaak van de component wanneer de component wordt getekend. Met de stijl `textFormat` kunt u de grootte en weergave van de waarde van de `NumericStepper` wijzigen. Met de verschillende stijlen van skins kunt u verschillende klassen opgeven die de skin kan gebruiken. Zie “Skins” op pagina 106 voor meer informatie over het gebruik van skinstijlen.

In deze procedure wordt de stijl `textFormat` gebruikt om de weergave te wijzigen van de waarde die de `NumericStepper` weergeeft.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component `NumericStepper` van het deelvenster Componenten naar het werkgebied en geef deze de instantienaam `myNs`.
- 3 Voeg de volgende code toe aan het deelvenster Handelingen in frame 1 van de hoofdtijdlijn:

```
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 Selecteer Besturing > Film testen.

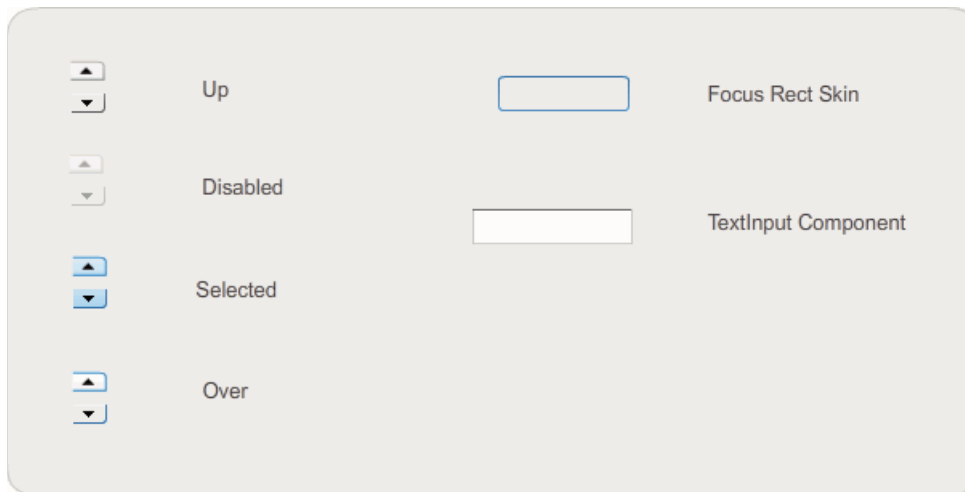
### Stijlen en de component `NumericStepper`

De component `NumericStepper` heeft skins die de toestanden Omhoog, Omlaag, Uitgeschakeld en Geselecteerd van zijn knoppen vertegenwoordigen.

Wanneer een stapfunctie is ingeschakeld, geven de knoppen Omlaag en Omhoog de toestand Boven weer als de muisaanwijzer erover beweegt. De knoppen geven hun toestand Omlaag weer wanneer ze worden ingedrukt. De knoppen keren terug naar de toestand Boven wanneer de muis wordt losgelaten. Wanneer de aanwijzer van de knoppen af beweegt, terwijl de muis is ingedrukt, keren de knoppen terug naar hun oorspronkelijke toestand.

Wanneer een stapfunctie is uitgeschakeld, wordt de uitgeschakelde toestand weergegeven, ongeacht gebruikersinteractie.

Een component NumericStepper bevat de volgende skins:



*Skins voor een NumericStepper*

- 1 Maak een nieuw FLA-bestand.
- 2 Sleep de component NumericStepper naar het werkgebied.
- 3 Stel het zoombesturingselement op 400% in om de afbeelding te vergroten voor bewerking.
- 4 Dubbelklik op de achtergrond van de skin TextInput in het deelvenster met skins tot u doordringt tot het groepsniveau en de achtergrondkleur wordt weergegeven in kleurkiezer voor de vulkleur in Eigenschapcontrole.
- 5 Selecteer met de kleurkiezer voor de vulkleur in Eigenschapcontrole de kleur #9999FF om deze op de achtergrond van de skin TextInput toe te passen.
- 6 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbeworkmodus.
- 7 Dubbelklik opnieuw op de NumericStepper om het deelvenster met skins te openen.
- 8 Dubbelklik op de achtergrond van de pijlknop Omhoog in de groep Omhoog totdat de achtergrond is geselecteerd en de kleur ervan wordt weergegeven in de kleurkiezer voor de vulkleur in Eigenschapcontrole.
- 9 Selecteer kleur #9966FF om deze op de achtergrond van de pijltoets Omhoog toe te passen.
- 10 Herhaal de stappen 8 en 9 voor pijl Omlaag in de groep Omhoog.
- 11 Selecteer Besturing > Film testen.

De instantie NumericStepper wordt weergegeven, zoals in de volgende afbeelding wordt getoond:



## De component ProgressBar aanpassen

U kunt een component ProgressBar horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of de toepasbare eigenschappen van de klasse ProgressBar gebruiken, zoals `height`, `width`, `scaleX` en `scaleY`.

De ProgressBar heeft drie skins: een skin voor het voorgangsvak, een skin voor een balk en een onbepaalde skin. Het gebruikt 9-delige schaling om de elementen te schalen.

## Stijlen en de component ProgressBar

U kunt stijleigenschappen instellen om de weergave van een instantie ProgressBar te wijzigen. De stijlen van de ProgressBar geven waarden op voor zijn skin en opvulling wanneer de component wordt getekend. In het volgende voorbeeld wordt een instantie ProgressBar vergroot en de stijl `barPadding` ervan ingesteld.

- 1 Maak een nieuw FLA-bestand.
- 2 Sleep de component ProgressBar van het deelvenster Componenten naar het werkgebied en geef deze de instantienaam **myPb**.
- 3 Voer in frame 1 van de hoofdtijdlijn de volgende code in het deelvenster Handelingen in:

```
myPb.width = 300;
myPb.height = 30;

myPb.setStyle("barPadding", 3);
```

- 4 Selecteer Besturing > Film testen.

Zie “Skins” op pagina 106 voor meer informatie over het instellen van skinstijlen.

## Skins en de component ProgressBar

De component ProgressBar gebruikt skins om het vak van de voortgangsbalk, de voltooide balk en een onbepaalde balk weer te geven, zoals getoond in de volgende afbeelding.



*Skins voor een ProgressBar*

De balk wordt over de skin van het regelvak geplaatst met behulp van de `barPadding` om de positie te bepalen. De elementen worden geschaald met 9-delige schaling.

De onbepaalde balk wordt gebruikt wanneer de eigenschap `indeterminate` van de instantie van de ProgressBar is ingesteld op `true`. De grootte van de skin wordt horizontaal en verticaal aangepast aan de grootte van de ProgressBar.

U kunt deze skins bewerken om de weergave van de ProgressBar te wijzigen. In het volgende voorbeeld wordt bijvoorbeeld de kleur van de onbepaalde balk gewijzigd.

- 1 Maak een nieuw FLA-bestand.
- 2 Sleep een component ProgressBar naar het werkgebied en dubbelklik erop om het deelvenster met skinpictogrammen te openen.
- 3 Dubbelklik op de skin van de onbepaalde balk.
- 4 Stel het zoombesturingselement op 400% in om het pictogram te vergroten voor bewerken.

- 5 Dubbelklik op één van de diagonale balken en houd vervolgens Shift ingedrukt terwijl u op de andere klikt. De huidige kleur wordt weergegeven in de kleurkiezer voor de vulkleur in Eigenschapcontrole.
- 6 Klik op de kleurkiezer voor de vulkleur in Eigenschapcontrole om deze te openen en selecteer kleur #00CC00 om deze toe te passen op de geselecteerde diagonale balken.
- 7 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbewerkmodus.
- 8 Selecteer Besturing > Film testen.

De ProgressBar wordt weergegeven, zoals in de volgende afbeelding wordt getoond.



## De component RadioButton aanpassen

U kunt een component RadioButton horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Gebruik de methode `setSize()` bij uitvoering.

Het selectiekader van een component RadioButton is onzichtbaar en geeft ook het raakgebied aan voor de component. Wanneer u de grootte van de component vergroot, vergroot u ook de grootte van het raakgebied.

Wanneer het selectiekader van de component te klein voor het componentlabel is, wordt het label op maat geknipt.

## Stijlen gebruiken met de component RadioButton

U kunt stijleigenschappen instellen om de weergave van een RadioButton te wijzigen. De stijleigenschappen van de RadioButton geven waarden op voor de skins, pictogrammen en tekstopmaak wanneer de component wordt getekend. De stijlen van de RadioButton geven waarden op voor de skins en opvulling, wanneer de component wordt getekend.

In het volgende voorbeeld wordt de stijl `textFormat` opgehaald uit een component CheckBox en toegevoegd aan een RadioButton om de stijl van hun labels identiek te maken.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component CheckBox naar het werkgebied en geef deze de instantienaam **myCh** in Eigenschapcontrole.
- 3 Sleep een component RadioButton naar het werkgebied en geef deze de instantienaam **myRb** in Eigenschapcontrole.
- 4 Voeg de volgende code toe aan het deelvenster Handelingen in frame 1 van de tijdlijn.

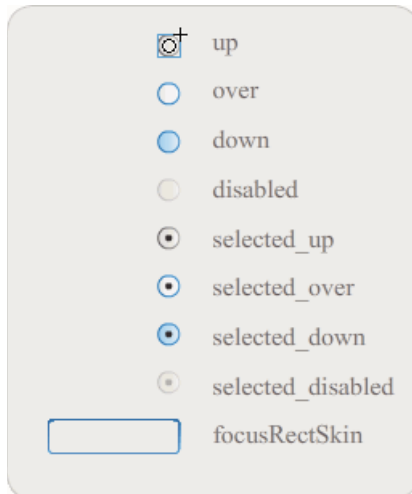
```
var tf:TextFormat = new TextFormat();
tf.color = 0x00FF00;
tf.font = "Georgia";
tf.size = 18;
myCh.setStyle("textFormat", tf);
myRb.setStyle("textFormat", myCh.getStyle("textFormat"));
```

Met deze code wordt de stijl `textFormat` voor de CheckBox ingesteld en vervolgens wordt deze op RadioButton toegepast door de methode `getStyle()` aan te roepen op de CheckBox.

- 5 Selecteer Besturing > Film testen.

## Skins en de component RadioButton

De RadioButton heeft de volgende skins, die u kunt bewerken om de weergave ervan te wijzigen.



*Skins voor een RadioButton*

Wanneer een RadioButton is ingeschakeld en niet geselecteerd, wordt de skin *Boven* weergegeven wanneer de gebruiker de aanwijzer erboven beweegt. Wanneer de gebruiker op een RadioButton klikt, ontvangt het invoerfocus en geeft het zijn skin *selected\_down* weer. Wanneer de gebruiker de muis loslaat, geeft het RadioButton de skin *selected\_up* weer. Wanneer de gebruiker de aanwijzer uit het raakgebied van RadioButton beweegt terwijl de muis wordt ingedrukt, keert de weergave van het RadioButton terug naar zijn skin *Up*.

Wanneer een RadioButton is uitgeschakeld, wordt de uitgeschakelde toestand weergegeven, ongeacht gebruikersinteractie.

In het volgende voorbeeld wordt de skin *selected\_up* vervangen, die geselecteerde toestand aangeeft.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component RadioButton naar het werkgebied en dubbelklik erop om het te openen in zijn deelvenster met skins.
- 3 Stel het zoombesturingselement op 800% in om het pictogram te vergroten voor bewerken.
- 4 Dubbelklik op de skin *selected\_up* om deze te selecteren en druk op *Delete* om deze te verwijderen.
- 5 Selecteer de rechthoek in het deelvenster *Gereedschappen*.
- 6 Stel in *Eigenschapcontrole* de lijnkleur in op rood (#FF0000) en de vulkleur op zwart (#000000).
- 7 Klik en sleep de aanwijzer om een rechthoek te tekenen en begin bij het kruisdraad dat het registratiepunt (ook *uitgangspunt* of *nulpunt* genoemd) van het symbool aangeeft.
- 8 Klik op de knop *Terug* links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbewerkmodus.
- 9 Selecteer *Besturing > Film testen*.
- 10 Klik op het keuzerondje om het te selecteren.

Het geselecteerde keuzerondje lijkt nu op de onderstaande afbeelding.



## De component ScrollPane aanpassen

U kunt een component ScrollPane horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of alle toepasselijke eigenschappen en methodes van de klasse ScrollPane gebruiken, zoals `height`, `width`, `scaleX` en `scaleY`.

De component ScrollPane moet de volgende grafische kenmerken hebben:

- Het registratiepunt (ook *uitgangspunt* of *nulpunt* genoemd) van de inhoud bevindt zich in de linkerbovenhoek van het veld.
- Wanneer de horizontale schuifbalk uitgeschakeld is, wordt de verticale schuifbalk rechts in het schuifvenster van boven naar beneden weergegeven. Wanneer de verticale schuifbalk uitgeschakeld is, wordt de horizontale schuifbalk onder in het schuifvenster van links naar rechts weergegeven. U kunt ook beide schuifbalken uitschakelen.
- Wanneer het schuifvenster te klein is, wordt de inhoud mogelijk niet correct weergegeven.
- Wanneer de grootte van het schuifvenster wordt gewijzigd, worden het schuifvak en het schuifblokje uitgevouwen of samengevouwen en wordt de grootte van hun raakgebied aangepast. De knoppen behouden hun grootte.

### Stijlen gebruiken met de component ScrollPane

Met de stijleigenschappen van de component ScrollPane worden waarden voor skins en opvulling voor de layout opgegeven wanneer de component wordt getekend. Met de diverse skinstijlen kunt u verschillende klassen opgeven die voor de skins van de component kunnen worden gebruikt. Zie “Skins” op pagina 106 voor meer informatie over het gebruik van skinstijlen.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component ScrollPane naar het werkgebied en geef deze de instantienaam **mySp**.
- 3 Klik op de tab Parameters in Eigenschapcontrole en voer de volgende waarde in voor de parameter `source`:  
**<http://www.helpexamples.com/flash/images/image1.jpg>**.
- 4 Voeg aan frame 1 van de hoofdtijdlijn de volgende code toe in het deelvenster Handelingen.  

```
mySp.setStyle("contentPadding", 5);
```

De opvulling wordt toegepast tussen de rand en de inhoud van de component, aan de buitenkant van de schuifbalken.
- 5 Selecteer Besturing > Film testen.

### Skins en de ScrollPane

De component ScrollPane bevat een rand en schuifbalken als schuifelement. Zie voor meer informatie over het toewijzen van skins aan schuifbalken “Skins gebruiken met de component UIScrollBar” op pagina 138.

## De component Slider aanpassen

U kunt een component Slider horizontaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Gebruik bij uitvoering de methode `setSize()` of alle toepasbare eigenschappen van de klasse `Slider`, zoals `width` en `scaleX`.

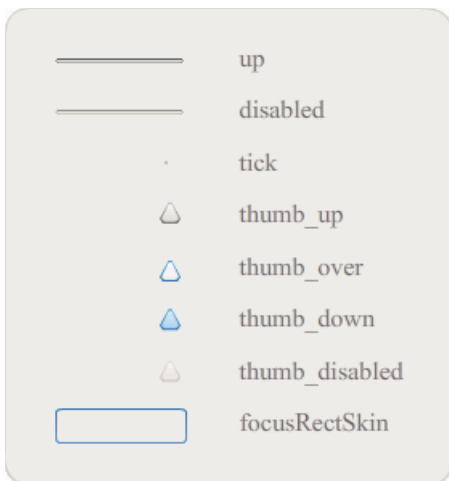
U kunt een schuifregelaar alleen langer maken. U kunt de hoogte ervan niet aanpassen. Flash negeert de eigenschap `height` en de parameter `height` van de methode `setSize()`. U kunt echter wel een verticale schuifregelaar maken en deze verticaal langer maken.

### Stijlen en de component Slider

Met de stijlen van de schuifregelaar worden alleen de klassen voor de skins opgegeven en een waarde voor `FocusRectPadding`, waarmee het aantal pixels wordt bepaald voor de opvulling tussen het selectiekader van de component en de buitenste begrenzing. Zie “Skins” op pagina 106 voor meer informatie over het gebruik van skinstijlen.

### Skins en de component Slider

De component Slider gebruikt de volgende skins, die u kunt bewerken om het uiterlijk te wijzigen.



*Skins voor Slider*

In het volgende voorbeeld wordt het vak in de toestand Omhoog bewerkt om de kleur in blauw te wijzigen.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component Slider van het deelvenster Componenten naar het werkgebied.
- 3 Dubbelklik op de component Slider om het deelvenster met skins te openen.
- 4 Dubbelklik op het registratiepunt van het vak in de toestand Omhoog om het vak in de symboolbewerkmodus te openen.
- 5 Stel het zoombesturingselement op 800% in om het pictogram te vergroten voor bewerken. Het Slider-vak bestaat uit drie balken.

- 6 Klik op de bovenste balk om deze te selecteren. Wanneer de balk is geselecteerd, wordt de kleur ervan weergegeven in de kleurkiezer voor de vulling in Eigenschapcontrole.
- 7 Selecteer met behulp van de kleurkiezer voor de vulling in Eigenschapcontrole kleur #000066 om deze toe te passen op de bovenste balk van het Slider-vak.
- 8 Klik op de middelste balk van het Slider-vak om deze te selecteren. Wanneer de balk is geselecteerd, wordt de kleur ervan weergegeven in de vulkleurkiezer in Eigenschapcontrole.
- 9 Selecteer met behulp van de kleurkiezer voor de vulling in Eigenschapcontrole kleur #0066FF om deze toe te passen op de middelste balk van het Slider-vak.
- 10 Klik op de onderste balk van het Slider-vak om deze te selecteren. Wanneer de balk is geselecteerd, wordt de kleur ervan weergegeven in de vulkleurkiezer in Eigenschapcontrole.
- 11 Selecteer met behulp van de kleurkiezer voor de vulling in Eigenschapcontrole kleur #00CCFF om deze toe te passen op de onderste balk van het Slider-vak.
- 12 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbewerkmodus.
- 13 Selecteer Besturing > Film testen.

De schuifregelaar lijkt nu op de onderstaande afbeelding.



## De component TextArea aanpassen

U kunt een component TextArea horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of alle toepasbare eigenschappen van de klasse TextArea gebruiken, zoals `height`, `width`, `scaleX` en `scaleY`.

Wanneer de grootte van een tekstgebied wordt aangepast, wordt de randgrootte aangepast aan het nieuwe selectiekader. De schuifbalken worden, indien nodig, langs de onder- en rechterraand geplaatst. De grootte van het tekstgebied wordt vervolgens aangepast binnen het resterende gebied. De component TextArea bevat geen elementen met een vaste grootte. Wanneer de breedte van het tekstgebied te klein is om de volledige tekst weer te geven, wordt de tekst geknipt.

### Stijlen en de component TextArea

Met de stijlen van de component TextArea worden waarden voor skins, opvulling en tekstopmaak opgegeven wanneer de component wordt getekend. De stijlen `texFormat` en `disabledTextFormat` zijn van toepassing op de stijl van de tekst die in de TextArea wordt weergegeven. Zie “[Skins gebruiken met de component TextArea](#)” op pagina 133 voor meer informatie over de stijleigenschappen van skins.

In het volgende voorbeeld wordt de stijl `disabledTextFormat` zo ingesteld dat het uiterlijk van tekst wordt veranderd wanneer de TextArea wordt uitgeschakeld, maar hetzelfde proces is van toepassing op het instellen van de stijl `textFormat` voor een TextArea die is ingeschakeld.

- 1 Maak een nieuw Flash-bestand.
- 2 Sleep een component TextArea naar het werkgebied en geef deze de instantienaam **myTa**.



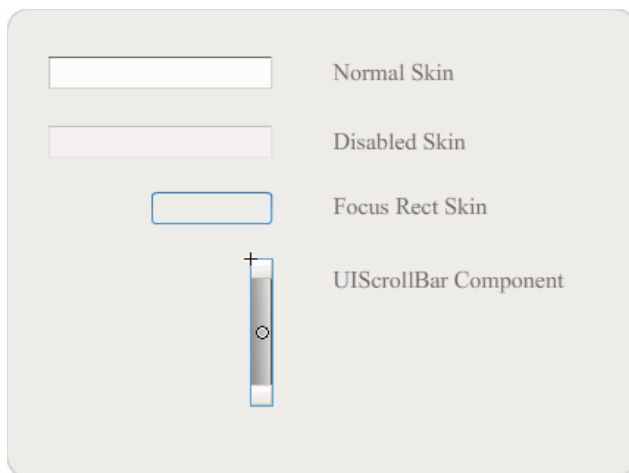
- 3 Voeg de volgende code toe aan het deelvenster Handelingen in frame 1 van de hoofdtijdlijn.

```
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 Selecteer Besturing > Film testen.

## Skins gebruiken met de component TextArea

De component TextArea gebruikt de volgende skins, die u kunt bewerken om het uiterlijk te wijzigen.



TextArea-skins

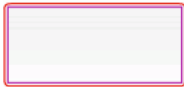
**Opmerking:** Het wijzigen van de skin ScrollBar in een component wijzigt deze voor alle andere componenten die de ScrollBar gebruiken.

De volgende procedure wijzigt de randkleuren van de skin voor de focusrechthoek, die wordt weergegeven wanneer de TextArea focus heeft en van de skin Normal.

- 1 Maak een nieuw Flash-bestand.
- 2 Sleep een component TextArea naar het werkgebied en dubbelklik erop om het deelvenster met skinpictogrammen te openen.
- 3 Dubbelklik op de skin voor de focusrechthoek.
- 4 Klik op de rand van de skin voor de focusrechthoek om deze te selecteren. Wanneer het is geselecteerd, wordt de huidige kleur ervan weergegeven in de kleurkiezer voor de vulkleur in Eigenschapcontrole.
- 5 Klik op de kleurkiezer voor de vulkleur in Eigenschapcontrole om deze te openen en selecteer kleur #CC0000 om deze op de rand toe te passen.
- 6 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbewerkmodus.
- 7 Dubbelklik op de component TextArea om het deelvenster met skinpictogrammen te openen.

- 8 Dubbelklik op de skin Up.
- 9 Selecteer telkens één kant van de rand van de skin Normal en stel de kleur in op #990099.
- 10 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbewerkmodus.
- 11 Selecteer Besturing > Film testen.

Wanneer u de TextArea selecteert om te beginnen met het invoeren van tekst, worden de randen weergegeven zoals in de volgende afbeelding wordt getoond:



De buitenste rand is de skin Focus Rect en de binnenste rand is de rand van de skin Normal.

Zie “[De component UIScrollBar aanpassen](#)” op pagina 138 voor meer informatie over het bewerken van de skin UIScrollBar.

## De component TextInput aanpassen

U kunt het formaat van een instantie TextInput wijzigen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of de toepasbare eigenschappen van de klasse TextInput gebruiken, zoals `height`, `width`, `scaleX` en `scaleY`.

Wanneer de grootte van een component TextInput wordt aangepast, wordt de randgrootte aangepast aan het nieuwe selectiekader. De component TextInput gebruikt geen schuifbalken, maar het invoegpunt schuift automatisch terwijl de gebruiker met de tekst ‘communiceert’. De grootte van het tekstveld wordt vervolgens aangepast binnen het resterende gebied. De component TextInput bevat geen elementen met een vaste grootte. Wanneer de component TextInput te klein is om de tekst weer te geven, wordt de tekst geknipt.

### Stijlen en de component TextInput

Met de stijlen van de component TextInput worden waarden voor skins, opvulling en tekstopmaak opgegeven wanneer de component wordt getekend. De stijlen `textFormat` en `disabledTextFormat` zijn van toepassing op de stijl van de tekst die in de component wordt weergegeven. Zie “[Skins en de component TextInput](#)” op pagina 135 voor meer informatie over de stijleigenschappen van skins.

In het volgende voorbeeld word de stijl `textFormat` ingesteld om het lettertype, het formaat en de kleur van de tekst in te stellen die in de component TextInput wordt weergegeven. Hetzelfde proces is van toepassing op het instellen van de stijl `disabledTextFormat` die wordt toegepast als de component is uitgeschakeld.

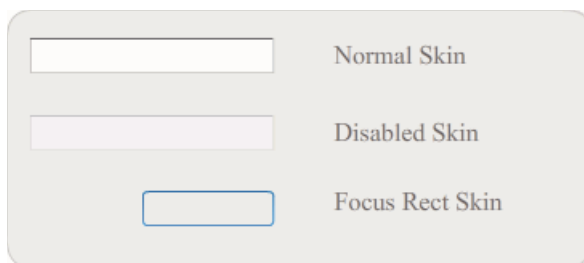
- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep een component TextInput naar het werkgebied en geef deze de instantienaam **myTi**.
- 3 Voeg de volgende code toe aan het deelvenster Handelingen in frame 1 van de hoofdtijdlijn.

```
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 Selecteer Besturing > Film testen.

## Skins en de component TextInput

De component TextInput gebruikt de volgende skins, die u kunt bewerken om het uiterlijk te wijzigen.



*Bijschrift van TextInput*

De volgende procedure wijzigt de rand- en achtergrondkleuren van een component TextInput:

- 1 Maak een nieuw Flash-bestand.
- 2 Sleep een component TextInput naar het werkgebied en dubbelklik erop om het deelvenster met skins te openen.
- 3 Dubbelklik op de skin Up.
- 4 Stel het zoombesturingselement op 800% in om het pictogram te vergroten voor bewerken.
- 5 Selecteer met één tegelijk iedere kant van de rand van de skin Normal en stel de kleur in op #993399 om dit toe te passen.
- 6 Dubbelklik op de achtergrond tot de kleur ervan in de vulkleurkiezer in Eigenschapcontrole wordt weergegeven. Selecteer de kleur #99CCCC om deze toe te passen op de achtergrond.
- 7 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied om terug te gaan naar de documentbewerkmodus.
- 8 Selecteer Besturing > Film testen.

De component TextArea ziet er nu uit zoals op de onderstaande afbeelding:



## De component `TileList` aanpassen

U kunt een component `TileList` horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of de juiste eigenschappen gebruiken, zoals `width`, `height`, `columnCount`, `rowCount`, `scaleX` en `scaleY`. De `ScrollBar`, die is opgenomen in `TileList`, wordt geschaald met de keuzelijst.

### Stijlen en de component `TileList`

Met de stijlen van `TileList` worden waarden voor skins, opvulling en tekstopmaak opgegeven wanneer de component wordt getekend. De stijlen `textFormat` en `disabledTextFormat` zijn van toepassing op de stijl van de tekst die in de component wordt weergegeven. Zie “[Skins gebruiken met de component `TileList`](#)” op pagina 136 voor meer informatie over de skinstijlen.

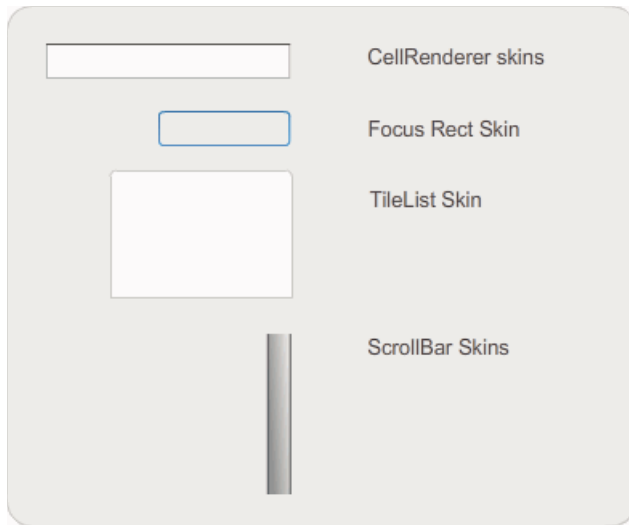
In het volgende voorbeeld wordt de methode `setRendererStyle()` aangeroepen met de stijl `textFormat` om het lettertype, het formaat, de kleur en tekstattributen van de labels in te stellen die in een instantie `TileList` worden weergegeven. Hetzelfde proces is van toepassing op het instellen van de stijl `disabledTextFormat` die wordt toegepast als de eigenschap `enabled` is ingesteld op `false`.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component `TileList` naar het werkgebied en geef deze de instantienaam **myTl**.
- 3 Voeg de volgende code toe aan het deelvenster Handelingen in frame 1 van de tijdlijn.

```
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);
```

### Skins gebruiken met de component `TileList`

De component `TileList` heeft een `TileList`-skin, een `CellRenderer`-skin en een `ScrollBar`-skin. U kunt deze skins bewerken om het uiterlijk van de `TileList` te wijzigen:



TileList-skins

**Opmerking:** Het wijzigen van de ScrollBar-skin in één component zorgt dat dit voor alle andere componenten die de ScrollBar gebruiken ook wordt gewijzigd.

De volgende procedure wijzigt de kleur van de CellRenderer-skin Selected\_Up van TileList.

- 1 Maak een Flash-document(ActionScript 3.0).
- 2 Sleep de component TileList naar het werkgebied en dubbelklik erop om het deelvenster met skins te openen.
- 3 Dubbelklik op de CellRenderer-skin en daarna op de Selected\_Up-skin, en klik daarna op de rechthoekige achtergrond.
- 4 Selecteer kleur #99FFFF met de kleurkiezer voor de vulkleur in Eigenschapcontrole om deze toe te passen op de Selected\_Up-skin.
- 5 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied totdat u terug gaat naar de documentbewerkmodus.
- 6 Dubbelklik op de tweede kolom van de rij dataProvider in het tabblad Parameters in Eigenschapcontrole om het dialoogvenster Waarden te openen. Voeg items met de volgende labels toe: 1e item, 2e item, 3e item, 4e item.
- 7 Selecteer Besturing > Film testen.
- 8 Klik op een van de cellen in de TileList om deze te selecteren en beweeg de muis daarna weg van de geselecteerde cel. De geselecteerde cel ziet er nu uit zoals op de onderstaande afbeelding.



TileList-component met aangepaste kleur voor skin Selected\_Up

## De component UILoader aanpassen

U kunt een component UILoader horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of de juiste eigenschappen gebruiken, zoals `width`, `height`, `scaleX` en `scaleY`.

Het groottegedrag van de component UILoader wordt bestuurd door de eigenschap `scaleContent`. Als `scaleContent` `true` is, wordt de inhoud zo geschaald dat deze binnen de grenzen van de lader past (en wordt herschaald wanneer `setSize()` wordt aangeroepen). Als `scaleContent` `false` is, wordt de grootte van de component aangepast aan de grootte van de inhoud en hebben `setSize()` en de eigenschappen voor grootte geen effect).

De component UILoader heeft geen elementen voor gebruikersinterface waarop u stijlen of skins kunt toepassen.

## De component UIScrollBar aanpassen

U kunt een component UIScrollBar horizontaal en verticaal aanpassen tijdens het ontwerpen en bij uitvoering. Een verticale UIScrollBar staat u echter niet toe de breedte aan te passen en een horizontale UIScrollBar staat het aanpassen van de hoogte niet toe. Selecteer tijdens het ontwerpen de component in het werkgebied en gebruik het gereedschap Vrije transformatie of een van de opdrachten Wijzigen > Transformeren. Bij uitvoering kunt u de methode `setSize()` of elke toepasbare eigenschap van de klasse UIScrollBar gebruiken, zoals `width`, `height`, `scaleX` en `scaleY`.

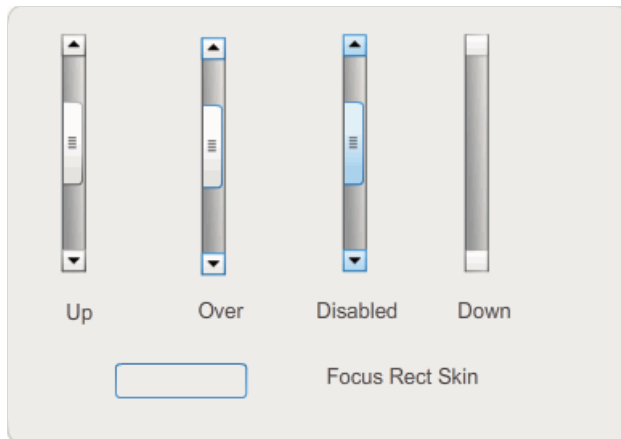
**Opmerking:** Als u de methode `setSize()` gebruikt, kunt u alleen de breedte van een horizontale schuifbalk of de hoogte van een verticale schuifbalk wijzigen. Bij het ontwerpen kunt u de hoogte van een horizontale schuifbalk of de breedte van een verticale schuifbalk instellen, maar de waarden worden opnieuw ingesteld bij het publiceren van de film. Alleen de afmeting van een schuifbalk die overeenkomt met de lengte kan gewijzigd worden.

### Stijlen gebruiken met de component UIScrollBar

Met de stijlen van de component UIScrollBar worden alleen de klassen voor de skins opgegeven en een waarde voor `FocusRectPadding`, waarmee het aantal pixels wordt bepaald voor de opvulling tussen het selectiekader van de component en de buitenste begrenzing. Zie “Skins” op pagina 106 voor meer informatie over het gebruik van skinstijlen.

### Skins gebruiken met de component UIScrollBar

De component UIScrollBar gebruikt de volgende skins.



UIScrollBar-skins

Zowel horizontale als verticale schuifbalken gebruiken dezelfde skins; bij de weergave van een horizontale schuifbalk draait de component UIScrollBar de skins op de juiste manier.

**Opmerking:** Het wijzigen van de ScrollBar-skin in één component zorgt dat dit voor alle andere componenten die de ScrollBar gebruiken ook wordt gewijzigd.

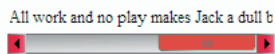
Het volgende voorbeeld toont hoe u de kleur van het blokje van de UIScrollBar en de pijlknoppen wijzigt.

- 1 Maak een nieuw Flash-document (ActionScript 3.0).
- 2 Sleep de component UIScrollBar naar het werkgebied en geef deze de instantienaam **mySb**. Stel op het tabblad Parameters de richting in op horizontaal.
- 3 Dubbelklik op de schuifbalk om het deelvenster met skins te openen.
- 4 Klik op de skin Up om deze te selecteren.
- 5 Stel het zoombesturingselement op 400% in om het pictogram te vergroten voor bewerken.
- 6 Dubbelklik op de achtergrond van de Pijl-rechts (of de Pijl-omhoog voor een verticale schuifbalk) totdat de achtergrond is geselecteerd en de kleur ervan wordt weergegeven in de kleurkiezer voor de vulkleur in Eigenschapcontrole.
- 7 Selecteer de kleur #CC0033 om deze toe te passen op de achtergrond van de knop.
- 8 Klik op de knop Terug links van de bewerkbalk boven in het werkgebied totdat u terug gaat naar de documentbewerkmodus.
- 9 Herhaal stappen 6, 7 en 8 voor de elementen blokje en Pijl-links (of Pijl-omlaag voor een verticale schuifbalk).
- 10 Voeg de volgende code in het deelvenster Handelingen toe aan frame 1 van de tijdlijn om de schuifbalk aan een TextField te koppelen.

```
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 Selecteer Besturing > Film testen.

De component UIScrollBar ziet er nu uit als op de onderstaande afbeelding.



*Horizontale ScrollBar met rood blokje en rode toetsen Pijl-links en Pijl-rechts*



# Hoofdstuk 6: De component FLVPlayback gebruiken

Met de component FLVPlayback kunt u eenvoudig een videospeler in uw CS5 Professional-toepassing opnemen om progressief gedownload videobestanden via HTTP af te spelen, of om streaming videobestanden af te spelen vanaf een Adobe's Macromedia Flash Media Server (FMS) of een Flash Video Streaming Service (FVSS).

Met de uitgave van Adobe Flash Player 9 Update 3 (versie 9.0.115.0 of hoger) zijn er belangrijke verbeteringen voor het afspelen van video-inhoud in Flash Player geïntegreerd. In deze update zijn er wijzigingen aangebracht op de component FLVPlayback die gebruikmaken van de videohardware op het systeem van de eindgebruiker om video beter te kunnen afspelen. De wijzigingen op de component FLVPlayback verhogen ook de precisie van videobestanden die in de modus Volledig scherm worden weergegeven.

Bovendien verbetert Flash Player 9 Update 3 de functionaliteit van de component FLVPlayback door ondersteuning toe te voegen voor de high-definition MPEG-4-videodelingen die gebruikmaken van de standaardcodering H.264. Tot deze indelingen behoren MP4, M4A, MOV, MP4V, 3GP en 3G2.

**Opmerking:** *Beveiligde MP4-bestanden, zoals bestanden die zijn gedownload vanaf Apple® iTunes® of die digitaal zijn gecodeerd door FairPlay®, worden niet ondersteund.*

De gebruikersvriendelijke component FLVPlayback heeft de volgende kenmerken en voordelen:

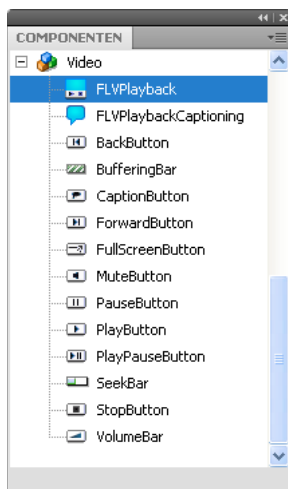
- Kan naar het werkgebied worden gesleept en snel en goed worden geïmplementeerd
- Ondersteunt volledige-schermgrootte
- Bevat een verzameling vooraf ontworpen *skins* waarmee u de weergave van de afspelbesturingselementen kunt aanpassen
- U kunt de kleur en alpha-waarden selecteren voor vooraf ontworpen skins
- Geavanceerde gebruikers kunnen hun eigen skins maken
- Live voorvertoning tijdens het ontwerpen
- Lay-outeigenschappen om het videobestand gecentreerd te houden bij wijziging van het formaat
- De mogelijkheid om het afspelen te starten wanneer voldoende van een progressief gedownload videobestand is gedownload
- Actiepunten waarmee u video met tekst, afbeeldingen en animatie kunt synchroniseren
- Het SWF-bestand behoudt een redelijke grootte

## Gebruik de component FLVPlayback

Wanneer u de component FLVPlayback gebruikt, plaatst u deze in het werkgebied en geeft u een videobestand op dat u wilt afspelen. U kunt ook diverse parameters instellen die het gedrag bepalen en het videobestand beschrijven.

De component FLVPlayback bevat tevens een ActionScript API (Application Programming Interface). De API bevat de volgende klassen, die volledig worden beschreven in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#): CuePointType, FLVPlayback, FLVPlaybackCaptioning, NCManager, NCManagerNative, VideoAlign, VideoError, VideoPlayer, VideoState en meerdere gebeurtenisklassen - AutoLayoutEvent, LayoutEvent, MetadataEvent, SkinErrorEvent, SoundEvent, VideoEvent en VideoProgressEvent.

De component FLVPlayback omvat de aangepaste UI-componenten voor het afspelen van FLV. De component FLVPlayback is een combinatie van het weergavegebied, of de videospeler, waarin het videobestand wordt weergegeven en de besturingselementen waarmee u kunt werken. De aangepaste UI-componenten voor het afspelen van FLV bieden knoppen en mechanismen waarmee u het videobestand kunt afspelen, stoppen, onderbreken en op andere wijze kunt besturen. Deze besturingselementen omvatten de BackButton, BufferingBar, CaptionButton (voor FLVPlaybackCaptioning), ForwardButton, FullScreenButton, MuteButton, PauseButton, PlayButton, PlayPauseButton, SeekBar, StopButton en VolumeBar. De component FLVPlayback en de aangepaste UI-componenten voor het afspelen van FLV worden in het deelvenster Componenten weergegeven (zie de volgende afbeelding):



FLVPlayback-componenten in het deelvenster Componenten

Het proces waarbij besturingselementen voor afspelen aan de component FLVPlayback worden toegevoegd, wordt *skins toewijzen* genoemd. De component FLVPlayback heeft een initiële standaardskin, SkinOverAll.swf, die zorgt voor de besturingselementen voor afspelen, stoppen, terug, vooruit, zoekbalk, dempen, volume, volledig scherm en ondertiteling. U kunt deze skin als volgt wijzigen:

- Selecteer een skin uit een verzameling vooraf ontworpen skins
- Maak een aangepaste skin en voeg deze toe aan de verzameling vooraf ontworpen skins
- Selecteer afzonderlijke besturingselementen uit de aangepaste UI-componenten voor het afspelen van FLV en pas deze aan

Wanneer u een vooraf ontworpen skin selecteert, kunt u de skinkleur en de alpha-waarde apart kiezen (tijdens het ontwerpen of bij uitvoering). Zie [“Een vooraf ontworpen skin selecteren”](#) op pagina 160 voor meer informatie.

Nadat u een andere skin hebt geselecteerd, wordt deze de nieuwe standaardskin.

Raadpleeg voor meer informatie over het selecteren of maken van een skin voor de component FLVPlayback [“De component FLVPlayback aanpassen”](#) op pagina 160.

## Een toepassing maken met de component FLVPlayback

U kunt op de volgende manieren de component FLVPlayback in uw toepassing opnemen:

- Sleep de component FLVPlayback van het deelvenster Componenten naar het werkgebied en geef een waarde op voor de parameter `source`.
- Gebruik de wizard Video importeren om de component in het werkgebied te maken en pas deze aan door een skin te selecteren
- Gebruik de constructor `FLVPlayback()` om dynamisch een instantie FLVPlayback in het werkgebied te maken, ervan uitgaande dat de component zich in de bibliotheek bevindt.

**Opmerking:** Als u een instantie van FLVPlayback met ActionScript maakt, moet u er ook een skin aan toewijzen door de eigenschap `skin` met ActionScript in te stellen. Wanneer u een skin op deze manier toepast, wordt de skin niet automatisch met het SWF-bestand gepubliceerd. U moet het SWF-bestand van zowel de toepassing als van de skin kopiëren naar uw toepassingsserver, anders is het SWF-bestand met skin niet beschikbaar wanneer u de toepassing uitvoert.

### De component FLVPlayback uit het deelvenster Componenten slepen

- 1 Klik op de plusknop (+) in het deelvenster Componenten om het video-item te openen.
- 2 Sleep de component FLVPlayback naar het werkgebied.
- 3 Selecteer de component FLVPlayback in het werkgebied en zoek de waardecel op voor de parameter `source` op het tabblad Parameters van Componentcontrole. Voer een tekenreeks in die een van de volgende items aangeeft:
  - Een lokaal pad naar een videobestand
  - Een URL naar een videobestand
  - Een URL naar een SMIL-bestand (Synchronized Multimedia Integration Language) dat beschrijft hoe een videobestand moet worden afgespeeldZie [“Een SMIL-bestand gebruiken”](#) op pagina 172 voor informatie over het maken van een SMIL-bestand om een of meer FVL-bestanden te beschrijven.
- 4 Klik op de waardecel voor de parameter `skin` op het tabblad Parameters in Componentcontrole, terwijl de component FLVPlayback in het werkgebied is geselecteerd.
- 5 Klik op het vergrootglas-pictogram om het dialoogvenster Skin selecteren te openen.
- 6 Selecteer een van de volgende opties:
  - Selecteer in de vervolgkeuzelijst Skin een van de vooraf ontworpen skins om een verzameling besturingselementen voor afspelen aan de component toe te wijzen.
  - Als u een aangepaste skin hebt gemaakt, selecteert u Aangepaste-skin-URL in het pop-upmenu en voert u in het vak URL de URL in voor het SWF-bestand met de skin.
  - Selecteer Geen en sleep afzonderlijke aangepaste UI-componenten voor het afspelen van FLV naar het werkgebied om besturingselementen voor het afspelen toe te voegen.

**Opmerking:** In de eerste twee gevallen wordt een voorbeeld van de skin weergegeven in het weergavevenster boven het pop-upmenu. U kunt de kleurkiezer gebruiken om de kleur van de skin te wijzigen.

U moet een aangepast UI-besturingselement aanpassen als u de kleur ervan wilt wijzigen. Zie [“Skins toewijzen aan afzonderlijke aangepaste UI-componenten voor het afspelen van FLV”](#) op pagina 162 voor meer informatie over het gebruik van aangepaste UI-besturingselementen.

- 7 Klik op OK om het dialoogvenster Skin selecteren te sluiten.

8 Selecteer Besturing > Film testen om het SWF-bestand uit te voeren en de video te starten.

In de volgende procedure wordt met de wizard Video importeren een component FLVPlayback toegevoegd:

**De wizard Video importeren gebruiken:**

1 Selecteer Bestand > Importeren > Video importeren.

2 Geef de locatie op van het videobestand door een van de volgende opties te selecteren:

- Op mijn lokale computer
- Al geïmplementeerd op een webserver, Flash Video Streaming Service of Flash Media Server

3 Afhankelijk van uw keuze voert u het pad of de URL in naar de locatie van het videobestand en klikt u op Volgende.

4 Als u een bestandspad hebt geselecteerd, wordt een dialoogvenster Implementatie geopend waarin u een van de weergegeven opties kunt selecteren en opgeven hoe u de video wilt implementeren:

- Progressieve download van een standaardwebserver
- Streamen via Flash Video Streaming Service
- Streamen via Flash Media Server
- Video insluiten in een SWF-bestand en afspelen in de tijdlijn

**Belangrijk:** U moet niet de optie Video insluiten selecteren. De component FLVPlayback speelt alleen externe streaming video af. Met deze optie wordt geen component FLVPlayback in het werkgebied geplaatst.

5 Klik op Volgende.

6 Selecteer een van de volgende opties:

- Selecteer in de vervolgkeuzelijst Skin een van de vooraf ontworpen skins om een verzameling besturingselementen voor afspelen aan de component toe te wijzen.
- Als u een aangepaste skin hebt gemaakt voor de component, selecteert u Aangepaste-skin-URL in het pop-upmenu en voert u in het vak URL de URL in voor het SWF-bestand met de skin.
- Selecteer Geen en sleep afzonderlijke aangepaste UI-componenten voor het afspelen van FLV naar het werkgebied om besturingselementen voor het afspelen toe te voegen.

**Opmerking:** In de eerste twee gevallen wordt een voorbeeld van de skin weergegeven in het weergavevenster boven het pop-upmenu.

7 Klik op OK om het dialoogvenster Skin selecteren te sluiten.

8 Kijk in het dialoogvenster Video-import voltooiën naar wat u vervolgens moet doen en klik op Voltooiën.

9 Als u het FLA-bestand nog niet hebt opgeslagen, wordt een dialoogvenster Opslaan als geopend.

10 Selecteer Besturing > Film testen om het SWF-bestand uit te voeren en de video te starten.

Met de volgende procedure wordt de component FLVPlayback toegevoegd met behulp van ActionScript.

**Op een dynamische manier een instantie maken met ActionScript:**

1 Sleep de component FLVPlayback van het deelvenster Componenten naar het deelvenster Bibliotheek (Venster > Bibliotheek).

2 Voeg de volgende code toe aan het deelvenster Handelingen in frame 1 van de tijdlijn. Wijzig *install\_drive* in het station waarop u Flash hebt geïnstalleerd en wijzig het pad om de locatie van de map Skins voor uw installatie aan te duiden.

Op een Windows-computer:

```
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";
```

Op een Macintosh-computer:

```
import fl.video.*;
var my_FLVPlybk = new FLVPlayback();
my_FLVPlybk.x = 100;
my_FLVPlybk.y = 100;
addChild(my_FLVPlybk);
my_FLVPlybk.skin = "file:///Macintosh HD:Applications:Adobe Flash
CS5:Configuration:FLVPlayback Skins:ActionScript 3.0SkinOverPlaySeekMute.swf"
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/water.flv";
```

**Opmerking:** Als u de eigenschappen *source* en *skin* niet instelt, lijkt de gegenereerde filmclip leeg.

- 3 Selecteer Besturing > Film testen om het SWF-bestand uit te voeren en het videobestand te starten.

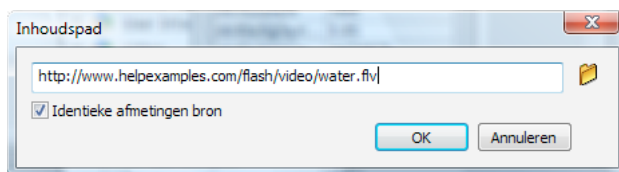
## Parameters voor de component FLVPlayback

U kunt voor elke instantie van de component FLVPlayback de volgende parameters instellen in Componentcontrole of Eigenschapscontrole: *align*, *autoPlay*, *cuePoints*, *preview*, *scaleMode*, *skin*, *skinAutoHide*, *skinBackgroundAlpha*, *skinBackgroundColor*, *source* en *volume*. Elk van deze parameters heeft een bijbehorende ActionScript-eigenschap met dezelfde naam. Wanneer u een waarde toewijst aan deze parameters, stelt u de initiële status in van de eigenschap in de toepassing. Bij het instellen van de eigenschap in ActionScript wordt de ingestelde waarde in de parameter genegeerd. Zie de klasse FLVPlayback in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over de mogelijk waarden voor deze parameters.

### De FLVPlayback-parameter source opgeven

Met de parameter *source* kunt u de naam en locatie opgeven van het videobestand, die in Flash als informatie dienen voor het afspelen van het bestand.

Open het dialoogvenster Inhoudspad door te dubbelklikken op de waardecel voor de parameter *source* in Componentcontrole.



Dialoogvenster Inhoudspad voor FLVPlayback

Het dialoogvenster Inhoudspad bevat de optie Identieke afmetingen bron-FLV, waarmee u kunt opgeven of de instantie FLVPlayback in het werkgebied overeen moet komen met de afmetingen van het videobronbestand. Het videobronbestand bevat de voorkeurshoogte en -breedte voor het afspelen. Als u deze optie selecteert, worden de afmetingen van de instantie van FLVPlayback aangepast aan deze voorkeursafmetingen.

### De bron

Voer de URL of het lokale pad in naar het videobestand of een XML-bestand dat beschrijft hoe het videobestand moet worden afgespeeld. Als u de precieze locatie van een videobestand niet weet, klikt u op het mappictogram om een dialoogvenster Bladeren te openen om de juiste locatie te vinden. Wanneer bij het bladeren naar een videobestand het bestand zich op of onder de locatie bevindt van het doel-SWF-bestand, wordt in Flash het pad automatisch relatief gemaakt ten opzichte van die locatie, zodat u het van een webserver kunt laten halen. Anders is het pad een absoluut Windows- of Macintosh-pad. U voert de naam van een lokaal XML-bestand in door het pad en de naam te typen.

Als u een HTTP-URL opgeeft, wordt het videobestand als een progressieve download afgespeeld. Als u een RTMP-URL opgeeft, streamt het videobestand vanaf Flash Media Server of FVSS. Een URL naar een XML-bestand kan ook een streaming videobestand zijn vanaf Flash Media Server of FVSS.

### Belangrijk:

U kunt ook de locatie van een SMIL-bestand opgeven dat beschrijft hoe meerdere streams van videobestanden voor meerdere bandbreedten moeten worden afgespeeld. In het bestand wordt SMIL (Synchronized Multimedia Integration Language) gebruikt om de FLV-bestanden te beschrijven. Zie “[Een SMIL-bestand gebruiken](#)” op pagina 172 voor een beschrijving van het SMIL-bestand.

U kunt ook de naam en locatie opgeven van het videobestand met de eigenschap `FLVPlayback.source` van ActionScript en de methodes `FLVPlayback.play()` en `FLVPlayback.load()`. Deze drie alternatieven hebben een hogere prioriteit dan de parameter `source` in Componentcontrole. Zie de items `FLVPlayback.source`, `FLVPlayback.play()` en `FLVPlayback.load()` voor de klasse `FLVPlayback` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#).

## Ondersteuning Volledig scherm

ActionScript 3.0 van de component `FLVPlayback` ondersteunt de modus Volledig scherm, waarvoor Flash Player 9.0.28.0 of hoger vereist is en de HTML moet zijn ingesteld op weergave in Volledig scherm. Sommige vooraf ontworpen skins zijn voorzien van een schakelknop om de weergave van het volledige scherm in en uit te schakelen. Het pictogram `FullScreenButton` wordt aan de rechterkant van de besturingsbalk weergegeven in de volgende illustratie.



Pictogram Volledig scherm op besturingsbalk

Ondersteuning voor Volledig scherm is alleen mogelijk wanneer de eigenschap `fullScreenTakeOver` is ingesteld op `true`, de standaardwaarde.

Ondersteuning voor Volledig scherm is mogelijk met of zonder ondersteuning voor hardwareversnelling. Zie “[Hardwareversnelling](#)” op pagina 149 voor informatie over ondersteuning van hardwareversnelling..

### Ondersteuning voor Volledig scherm voor FLVPlayback toepassen:

- 1 Voeg de component `FLVPlayback` toe aan uw toepassing en wijs er een videobestand aan toe.
- 2 Selecteer een skin voor de component `FLVPlayback` die over de knop Volledig scherm beschikt (zoals `SkinUnderPlaySeekFullscreen.swf`) of voeg de gebruikersinterfacecomponent `FullScreenButton` toe aan de component `FLVPlayback` vanaf het gedeelte Video in het deelvenster Componente.
- 3 Selecteer Bestand > Publicatie-instellingen.
- 4 Klik op het tabblad HTML in het dialoogvenster Publicatie-instellingen.

- 5 Selecteer op het tabblad HTML de optie Flash met ondersteuning voor volledig scherm in het pop-upmenu Sjabloon.
- 6 Selecteer op hetzelfde tabblad de optie Flash-versiedetectie en geef de versie 9.0.28 of hoger op, afhankelijk van de Flash Player-versie die u gebruikt.
- 7 Selecteer de tab Indelingen en zorg ervoor dat zowel de optie Flash (.swf) als HTML (.html) is geselecteerd. U kunt de standaard bestandsnamen vervangen.
- 8 Klik op Publiceren en vervolgens op OK.

In plaats van stap 7 kunt u ook op OK klikken en Bestand > Voorvertoning publiceren > Standaard - (HTML) selecteren om het geëxporteerde HTML-bestand in uw standaard browser te zetten. Anders opent u het HTML-bestand met uw browser om de optie Volledig scherm te testen.

Open het geëxporteerde HTML-bestand en kopieer de code waarmee het SWF-bestand in het HTML-bestand voor uw webpagina is ingesloten om de component FLVPlayback met ondersteuning voor een volledig scherm op uw webpagina te plaatsen. Deze code dient vergelijkbaar te zijn met het volgende voorbeeld:

```
//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 (
 'codebase', 'http://download.macromedia.com/pub/
 shockwave/cabs/flash/swflash.cab#version=9,0,28,0';,
 'width', '550';,
 'height', '400';,
 'src', 'fullscreen';,
 'quality', 'high';,
 'pluginspage', 'http://www.macromedia.com/go/
 getflashplayer';,
```

```
 'align';, 'middle';,
 'play';, 'true';,
 'loop';, 'true';,
 'scale';, 'showall';,
 'wmode';, 'window';,
 'devicefont';, &apos>false';,
 'id';, 'fullscreen';,
 'bgcolor';, '#ffffff';,
 'name';, 'fullscreen';,
 'menu';, 'true';,
 'allowScriptAccess';, 'sameDomain';,
 'allowFullScreen';, 'true';,
 'movie';, 'fullscreen';,
 'salign';, '';); //end AC code
 } else { // Flash is too old or we can't detect the plug-in.
 var alternateContent = 'Alternative HTML content should be placed
 here.';
 + 'This content requires Adobe Flash Player.';
 + 'Get Flash
 ';;
 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.
 Get Flash
</noscript>
```

Het is ook mogelijk het geëxporteerde HTML-bestand als sjabloon voor uw webpagina te gebruiken en er andere inhoud aan toe te voegen. Als u dit doet, moet u de naam van het HTML-bestand wel wijzigen zodat u het niet per ongeluk overschrijft door het HTML-bestand FLVPlayback weer vanuit Flash te exporteren.

U moet het in elk geval ook het bestand AC\_RunActiveContent.js naar uw webserver laden, dat naar dezelfde map wordt geëxporteerd als het HTML-bestand.

De ActionScript-ondersteuning voor de modus Volledig scherm bevat de eigenschappen

`fullScreenBackgroundColor`, `fullScreenSkinDelay`, en `fullScreenTakeOver` en de methode `enterFullScreenDisplayState()`. Zie de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor informatie over deze ActionScript-elementen.

## enterFullScreenDisplayState() gebruiken

U kunt de modus Volledig scherm ook oproepen door ActionScript-methode `enterFullScreenDisplayState()` te activeren, zoals in het onderstaande voorbeeld wordt weergegeven.

```
function handleClick(e:MouseEvent):void {
 myFLVPlybk.enterFullScreenDisplayState();
}
myButton.addEventListener(MouseEvent.CLICK, handleClick);
```



In dit voorbeeld wordt de modus Volledig scherm *niet* opgeroepen door op de schakelknop voor volledig scherm op een FLVPlayback-skin te klikken, maar door op een knop (MyButton) te klikken die de maker van de webpagina heeft toegevoegd om de modus Volledig scherm te activeren. Door op deze knop te klikken wordt de gebeurtenishandler `handleClick` geactiveerd, waarmee u de methode `enterFullScreenDisplayState()` oproept.

Met de methode `enterFullScreenDisplayState()` stelt u de eigenschap `Stage.displayState` in op `StageDisplayState.FULL_SCREEN` en heeft daarom dezelfde beperkingen als de eigenschap `displayState`. Raadpleeg voor meer informatie over de methode `enterFullScreenDisplayState()` en de eigenschap `Stage.displayState`-eigenschap, zie de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#).

## Hardwareversnelling

Flash Player 9.0.115.0 en latere versies bevatten codering die gebruik maakt van beschikbare videohardware om de prestaties en nauwkeurigheid van FLV-bestanden te verbeteren die door FLVPlayback in de modus Volledig scherm worden afgespeeld. Als aan alle voorwaarden is voldaan en de eigenschap `fullScreenTakeOver` is ingesteld op `true`, gebruikt Flash Player hardwareversnelling om het videobestand te schalen in plaats van software. Als de component FLVPlayback wordt uitgevoerd in een eerdere versie van Flash Player, of als er geen voorwaarden voor hardwareversnelling zijn, schaal Flash Player het videobestand weer zelf.

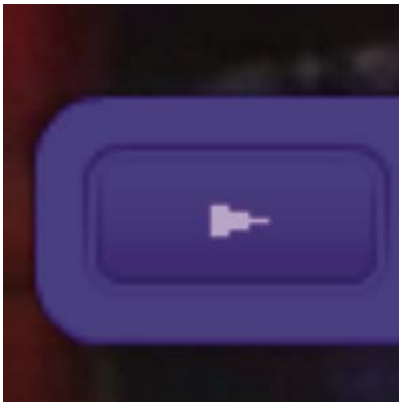
Wanneer u gebruik wilt maken van hardwareversnelling voor ondersteuning van een volledig scherm, moet uw computer over een videokaart beschikken die compatibel is met DirectX 7 met 4 MB geheugen of meer van VRAM (video RAM). Deze hardwareondersteuning is beschikbaar in Windows 2000 of Mac OS X 10.2 en latere versies van deze besturingssystemen. Direct X<sup>®</sup> biedt API's die een interface vormen tussen software en de videohardware om onder andere driedimensionale en tweedimensionale afbeeldingen te versnellen.

Als u gebruik wilt maken van de modus Hardwareversnelling, moet u de modus Volledig scherm ook op een van de volgende manieren oproepen:

- Met de schakelknop voor volledig scherm gebruiken op een FLVPlayback-skin
- Met de `FullScreenButton`-videobesturing
- Met de ActionScript-methode `enterFullScreenDisplayState()`. Zie "[enterFullScreenDisplayState\(\) gebruiken](#)" op pagina 148 voor meer informatie.

Als u de modus Volledig scherm oproept door de eigenschap `Stage.displayState` in te stellen op `StageDisplayState.FULLSCREEN`, maakt FLVPlayback geen gebruik van hardwareversnelling, zelfs niet als de videohardware en het geheugen beschikbaar zijn.

Een gevolg van het gebruik van hardwareversnelling voor ondersteuning van een volledig scherm, is dat behalve de videospeler en het videobestand ook de FLVPlayback-skins worden geschaald. De volgende afbeelding geeft het effect weer van de modus Volledig scherm met hardwareversnelling op de FLVPlayback-skin. U ziet hiervan een detail met de volledige resolutie.



*Modus Volledig scherm op een beeldscherm van 1600x1200 met een pixelvideo van 320x240*

Deze afbeelding laat het resultaat zien van het gebruik van de modus Volledig scherm op een 1600 x 1200 monitor op een videobestand met een breedte van 320 en een hoogte van 240, de standaard FLVPlayback-afmetingen. Het vervormingseffect op de skin is bij FLV-bestanden met kleinere afmetingen of op een grotere monitor duidelijker zichtbaar. Aan de andere kant is het vervormingseffect minder duidelijk bij grotere FLV-bestanden of op kleinere monitoren. Wanneer u de afmetingen verandert van 640 x 480 in 1600 x 1200, wordt de skin nog steeds groter, maar ziet er minder vervormd uit.

U kunt de eigenschap `skinScaleMaximum` zo instellen dat de schaling van de skin FLVPlayback wordt beperkt. De standaardwaarde is 4.0 of 400%. Door de schaling van de skin te beperken is er echter een combinatie van hardware en software nodig voor schaling van de FLV. Dit kan negatieve invloed hebben op de prestaties van FLV's met grote afmetingen die gecodeerd zijn bij een hoge bitsnelheid. Als de video groot is (bijvoorbeeld minimaal 640 pixels breed en 480 pixels hoog) moet u `skinScaleMaximum` niet op een kleine waarde instellen, omdat hierdoor merkbare prestatieproblemen kunnen optreden bij grote beeldschermen. Met de eigenschap `skinScaleMaximum` kunt u de afweging maken tussen prestaties en kwaliteit en het uiterlijk van een grote skin.

## De modus Volledig scherm afsluiten

U sluit de modus Volledig scherm af door opnieuw op de knop Volledig scherm te klikken of op de toets Esc te drukken.

Door de volgende eigenschappen in te stellen en de volgende methoden op te roepen kunnen lay-outveranderingen optreden waardoor de component FLVPlayback de modus Volledig scherm afsluit: `height`, `registrationHeight`, `registrationWidth`, `registrationX`, `registrationY`, `scaleX`, `scaleY`, `width`, `x`, `y`, `setScale()`, `setSize()`.

Als u de eigenschappen `align` en `scaleMode` instelt, stelt FLVPlayback deze in op `center` en `maintainAspectRatio` totdat de modus Volledig scherm wordt afgesloten.

Als u de waarde van de eigenschap `fullScreenTakeOver` wijzigt van `true` in `false` wanneer u de volledige schermweergave gebruikt, zorgt ook de modus Hardwareversnelling ervoor dat Flash de modus Volledig scherm afsluit.

## Lay-out uitlijnen voor het afspelen van meerdere FLV-bestanden

ActionScript 3.0 FLVPlayback heeft een eigenschap `align` die opgeeft of het videobestand wordt gecentreerd wanneer de grootte wordt gewijzigd of boven, onder, links of rechts in de component wordt geplaatst. Behalve de eigenschappen `x`, `y`, `width` en `height` van de component, heeft de ActionScript 3.0-component de eigenschappen `registrationX`, `registrationY`, `registrationWidth`, en `registrationHeight`. In eerste instantie komen deze overeen met de eigenschappen `x`, `y`, `width` en `height`. Bij het laden van volgende videobestanden worden deze door het automatisch opnieuw bepalen van de lay-out niet gewijzigd, zodat het nieuwe videobestand op dezelfde positie kan worden gecentreerd. Als `scaleMode = VideoScaleMode.MAINTAIN_ASPECT_RATIO`, kunnen FLV-bestanden worden ingepast in de oorspronkelijke afmetingen van de component zonder de breedte en hoogte van de component gewijzigd hoeven te worden.

## Progressief gedownload videobestanden automatisch afspelen

Bij het downloaden van een progressief gedownload videobestand wordt het afspelen van het videobestand in FLVPlayback pas gestart wanneer voldoende van het bestand is gedownload om het videobestand van begin tot eind af te spelen.

Als u het videobestand wilt afspelen voordat voldoende is gedownload, roept u de methode `play()` aan zonder parameters.

Als u de status wilt retourneren van het wachten of voldoende van het FLV-bestand is gedownload, roept u de methode `pause()` aan en vervolgens de methode `playWhenEnoughDownloaded()`.

## Actiepunten gebruiken

Een actiepunt is een punt waarop de videospeler een gebeurtenis `cuePoint` verstuurt tijdens het afspelen van een FLV-bestand. U kunt actiepunten toevoegen aan een FLV-bestand op tijdstippen waarop u een handeling wilt laten plaatsvinden voor een ander element op de webpagina. U wilt wellicht tekst of een afbeelding weergeven of synchroniseren met een Flash-animatie of het afspelen van het FLV-bestand onderbreken, zoeken tot een ander punt in de video of overschakelen naar een ander FLV-bestand. Met actiepunten kunt u met de code in ActionScript de synchronisatie besturen van punten in het FLV-bestand met andere gebeurtenissen op de webpagina.

Er zijn drie typen actiepunten: `navigation`, `event` en `ActionScript`. De actiepunten voor navigatie en gebeurtenis worden ook wel *ingesloten* actiepunten genoemd, omdat ze zijn ingesloten in de stream van het FLV-bestand en in het pakket metagegevens van het FLV-bestand.

Met een *navigatieactiepunt* kunt u naar een bepaald frame in het FLV-bestand zoeken omdat binnen het FLV-bestand een hoofdframe wordt gemaakt dat zo dicht mogelijk bij de tijd ligt die u opgeeft. Een *hoofdframe* is een gegevenssegment dat optreedt tussen afbeeldingsframes in de stream van het FLV-bestand. Wanneer u naar een navigatieactiepunt zoekt, zoekt de component naar het hoofdframe en start deze de gebeurtenis `cuePoint`.

Met een *gebeurtenisactiepunt* kunt u een punt in de tijd binnen het FLV-bestand synchroniseren met een externe gebeurtenis op de webpagina. De gebeurtenis `cuePoint` treedt precies op de opgegeven tijd op. U kunt navigatie- en gebeurtenisactiepunten in een FLV-bestand insluiten met de wizard Video importeren of Flash Video Encoder. Raadpleeg Hoofdstuk 16, 'Werken met video', in *Flash gebruiken* voor meer informatie over de wizard Video importeren en Flash Video Encoder.

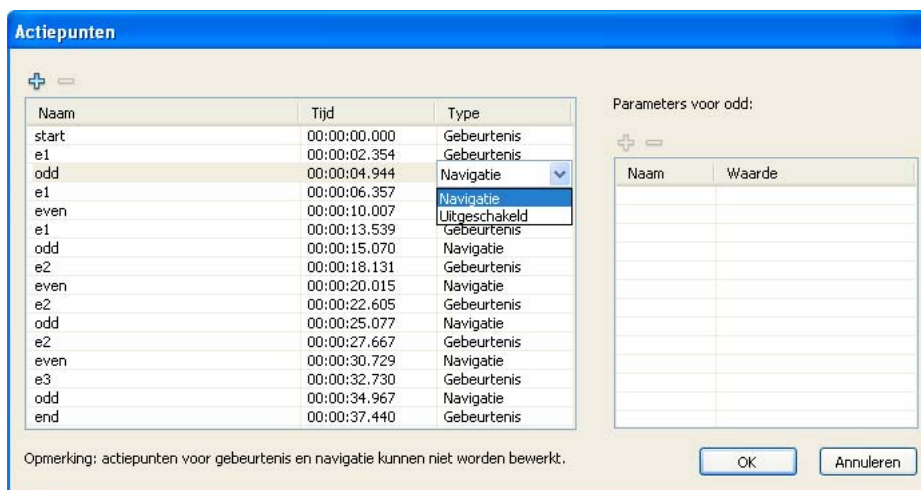
Een *ActionScript-actiepunt* is een extern actiepunt dat u kunt toevoegen via het dialoogvenster Flash Video-actiepunten van de component of via de methode `FLVPlayback.addASCuePoint()`. ActionScript-actiepunten worden in de component apart van het FLV-bestand opgeslagen en bijgehouden en zijn dan ook minder nauwkeurig dan ingesloten actiepunten. ActionScript-actiepunten zijn nauwkeurig tot op een tiende van een seconde. U kunt de nauwkeurigheid van ActionScript-actiepunten verbeteren door de waarde van de eigenschap `playheadUpdateInterval` te verlagen. De component genereert namelijk de gebeurtenis `cuePoint` voor ActionScript-actiepunten wanneer de afspreekop wordt bijgewerkt. Zie de eigenschap `FLVPlayback.playheadUpdateInterval` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#) voor meer informatie.

In ActionScript en in de metagegevens van het FLV-bestand wordt een actiepunt vertegenwoordigd door een object met de volgende eigenschappen: `name`, `time`, `type` en `parameters`. De eigenschap `name` is een tekenreeks die de toegewezen naam van het actiepunt bevat. De eigenschap `time` is een getal dat de tijd in uren, minuten, seconden en milliseconden (HH:MM:SS.mmm) vertegenwoordigt wanneer het actiepunt optreedt. De eigenschap `type` is een tekenreeks met de waarde "navigation", "event" of "actionscript", afhankelijk van het type actiepunt dat u hebt gemaakt. De eigenschap `parameters` is een array van opgegeven namen en waarden.

Wanneer een gebeurtenis `cuePoint` optreedt, is het actiepuntobject beschikbaar in het gebeurtenisobject via de eigenschap `info`.

### Het dialoogvenster Flash Video-actiepunten gebruiken

Open het dialoogvenster Flash Video-actiepunten door te dubbelklikken op de waardecel voor de parameter `cuePoints` in Componentcontrole. Het dialoogvenster ziet uit als in de volgende afbeelding:



Dialoogvenster Actiepunten

Het dialoogvenster bevat ingesloten actiepunten en ActionScript-actiepunten. U kunt dit dialoogvenster gebruiken om ActionScript-actiepunten en actiepuntparameters toe te voegen en te verwijderen. U kunt ingesloten actiepunten ook inschakelen of uitschakelen. U kunt ingesloten actiepunten echter niet toevoegen, wijzigen of verwijderen.

#### Een ActionScript-actiepunt toevoegen:

- 1 Dubbelklik op de waardecel van de parameter `cuePoints` in Componentcontrole om het dialoogvenster Flash-actiepunten te openen.
- 2 Klik op het plusteken (+) in de linkerbovenhoek, boven de lijst met actiepunten, om een standaarditem voor een ActionScript-actiepunt toe te voegen.

- 3 Klik in de kolom Naam op de tekst Nieuw actiepunt en geef het actiepunt een naam.
- 4 Klik op de waarde onder Tijd op 00:00:00:000 en voer de tijd in waarop het actiepunt moet optreden. U kunt de tijd opgeven in uren, minuten, seconden en milliseconden (HH:MM:SS.mmm).  
Bij meerdere actiepunten wordt het nieuwe actiepunt op chronologische volgorde in de lijst geplaatst.
- 5 Als u een parameter wilt toevoegen voor het geselecteerde actiepunt, klikt u op het plusteken (+) boven de sectie Parameters en voert u waarden in de kolommen Naam en Waarde in. Herhaal deze stap voor elke parameter.
- 6 Als u meer ActionScript-actiepunten wilt toevoegen, herhaalt u stap 2 tot en met 5 voor elk actiepunt.
- 7 Klik op OK om uw wijzigingen op te slaan.

#### Een ActionScript-actiepunt verwijderen:

- 1 Dubbelklik op de waardecel van de parameter `cuePoints` in Componentcontrole om het dialoogvenster Flash-actiepunten te openen.
- 2 Selecteer het actiepunt dat u wilt verwijderen.
- 3 Klik op het minteken (-) in de linkerbovenhoek, boven de lijst met actiepunten, om het actiepunt te verwijderen.
- 4 Herhaal stap 2 en 3 voor elk actiepunt dat u wilt verwijderen.
- 5 Klik op OK om uw wijzigingen op te slaan.

#### Een ingesloten actiepunt in het FLV-bestand inschakelen of uitschakelen:

- 1 Dubbelklik op de waardecel van de parameter `cuePoints` in Componentcontrole om het dialoogvenster Flash-actiepunten te openen.
- 2 Selecteer het actiepunt dat u wilt inschakelen of uitschakelen.
- 3 Klik op de waarde in de kolom Type om het pop-upmenu te openen, of klik op Pijl-omlaag.
- 4 Klik op de naam van het type actiepunt (bijvoorbeeld Gebeurtenis of Navigatie) om het in te schakelen. Klik op Uitschakelen om het uit te schakelen.
- 5 Klik op OK om uw wijzigingen op te slaan.

#### Actiepunten met ActionScript gebruiken

U kunt met ActionScript ActionScript-actiepunten toevoegen, luisteren naar gebeurtenissen van `cuePoint`, actiepunten van elk type of een bepaald type zoeken, zoeken naar een navigatieactiepunt, een actiepunt in- of uitschakelen, controleren of een actiepunt is ingeschakeld en een actiepunt verwijderen.

In de voorbeelden in deze sectie wordt een FLV-bestand met de naam `cuepoints.flv` gebruikt, dat de volgende drie actiepunten bevat:

Naam	tijd	Type
point1	00:00:00.418	Navigatie
point2	00:00:07.748	Navigatie
point3	00:00:16.020	Navigatie

### ActionScript-actiepunten toevoegen

U kunt ActionScript-actiepunten toevoegen aan een FLV-bestand met behulp van de methode `addASCuePoint()`. In het volgende voorbeeld worden twee ActionScript-actiepunten toegevoegd aan het FLV-bestand wanneer het gereed is om af te spelen. Het eerste actiepunt wordt toegevoegd met behulp van een actiepuntobject, dat de tijd, naam en het type van het actiepunt opgeeft in de eigenschappen ervan. In de tweede aanroep worden de tijd en de naam opgegeven via de parameters `time` en `name` van de methode.

```
// 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");
```

Zie de methode `FLVPlayback.addASCuePoint()` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#) voor meer informatie.

### Luisteren naar gebeurtenissen van cuePoint

Met de gebeurtenis `cuePoint` kunt u in uw ActionScript-code controle krijgen wanneer een gebeurtenis van `cuePoint` optreedt. Wanneer actiepunten in het volgende voorbeeld optreden, roept de `cuePoint` listener een gebeurtenishandlerfunctie aan die de waarde weergeeft van de eigenschap `playheadTime` en de naam en het type van het actiepunt. Gebruik dit voorbeeld in combinatie met het voorbeeld in de voorgaande sectie, ActionScript-actiepunten toevoegen, om de resultaten te bekijken.

```
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);
}
```

Zie voor meer informatie over de gebeurtenis `cuePoint` de gebeurtenis `FLVPlayback.cuePoint` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#).

### Actiepunten zoeken

Met ActionScript kunt u actiepunten van elk type zoeken, het dichtstbijzijnde actiepunt op een bepaalde tijd zoeken of het volgende actiepunt met een bepaalde naam zoeken.

De gebeurtenishandler `ready_listener()` in het volgende voorbeeld roept de methode `findCuePoint()` aan om het actiepunt `ASpt1` te zoeken en roept vervolgens de methode `findNearestCuePoint()` aan om het navigatieactiepunt te zoeken dat het dichtst bij de tijd van het actiepunt `ASpt1` ligt:

```
import fl.video.FLVPlayback;
import fl.video.CuePointType;
import fl.video.VideoEvent;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var rtn_obj:Object; //create cue point object
my_FLVPlybk.addASCuePoint(2.02, "ASpt1");//add AS cue point
function ready_listener(eventObject:VideoEvent):void {
 rtn_obj = my_FLVPlybk.findCuePoint("ASpt1", CuePointType.ACTIONSCRIPT);
 traceit(rtn_obj);
 rtn_obj = my_FLVPlybk.findNearestCuePoint(rtn_obj.time, CuePointType.NAVIGATION);
 traceit(rtn_obj);
}
my_FLVPlybk.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);
}
```

In het volgende voorbeeld zoekt de gebeurtenishandler `ready_listener()` het actiepunt `ASpt` en roept de methode `findNextCuePointWithName()` aan om het volgende actiepunt met dezelfde naam te zoeken:

```
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var rtn_obj:Object; //create cue point object
my_FLVPlybk.addASCuePoint(2.02, "ASpt");//add AS cue point
my_FLVPlybk.addASCuePoint(3.4, "ASpt");//add 2nd ASpt
my_FLVPlybk.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:VideoEvent):void {
 rtn_obj = my_FLVPlybk.findCuePoint("ASpt", CuePointType.ACTIONSCRIPT);
 traceit(rtn_obj);
 rtn_obj = my_FLVPlybk.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);
}
```

Raadpleeg voor meer informatie over Label-stijlen de methoden `FLVPlayback.findCuePoint()`, `FLVPlayback.findNearestCuePoint()` en `FLVPlayback.findNextCuePointWithName()` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#).

### Navigatieactiepunten zoeken

U kunt zoeken naar een navigatieactiepunt, zoeken naar het volgende navigatieactiepunt vanaf een bepaalde tijd en zoeken naar het vorige navigatieactiepunt vanaf een bepaalde tijd. In het volgende voorbeeld wordt het FLV-bestand `cuepoints.flv` afgespeeld en gezocht naar het actiepunt op 7,748 wanneer de gebeurtenis `ready` optreedt. Wanneer de gebeurtenis `cuePoint` optreedt, wordt in het voorbeeld de methode `seekToPrevNavCuePoint()` aangeroepen om naar het eerste actiepunt te zoeken. Wanneer die gebeurtenis `cuePoint` optreedt, wordt in het voorbeeld de methode `seekToNextNavCuePoint()` aangeroepen om naar het laatste actiepunt te zoeken door 10 seconden toe te voegen aan `eventObject.info.time`, de tijd van het huidige actiepunt.

```
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";
```

Zie de methoden `FLVPlayback.seekToNavCuePoint()`, `FLVPlayback.seekToNextNavCuePoint()` en `FLVPlayback.seekToPrevNavCuePoint()` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#).

### Ingesloten actiepunten in het FLV-bestand inschakelen en uitschakelen

U kunt ingesloten actiepunten in een FLV-bestand in- en uitschakelen met behulp van de methode `setFLVCuePointEnabled()`. Uitgeschakelde actiepunten activeren geen gebeurtenissen van `cuePoint` en werken niet met de methoden `seekToCuePoint()`, `seekToNextNavCuePoint()` of `seekToPrevNavCuePoint()`. U kunt uitgeschakelde actiepunten echter wel zoeken met de methoden `findCuePoint()`, `findNearestCuePoint()` en `findNextCuePointWithName()`.

U kunt testen of een ingesloten actiepunt in een FLV-bestand is ingeschakeld met behulp van de methode `isFLVCuePointEnabled()`. In het volgende voorbeeld worden de ingesloten actiepunten `point2` en `point3` uitgeschakeld wanneer de video gereed is om af te spelen. Wanneer de eerste gebeurtenis `cuePoint` optreedt, zal de gebeurtenishandler echter testen of het actiepunt `point3` is uitgeschakeld en, zo ja, het actiepunt inschakelen.

```
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");
 }
}
```

Zie de methoden `FLVPlayback.isFLVCuePointEnabled()` en `FLVPlayback.setFLVCuePointEnabled()` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#).

### Een ActionScript-actiepunt verwijderen

U kunt een ActionScript-actiepunt verwijderen met behulp van de methode `removeASCuePoint()`. In het volgende voorbeeld wordt het actiepunt `ASpt2` verwijderd wanneer het actiepunt `ASpt1` optreedt:



```
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(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
 trace("Cue point name is: " + eventObject.info.name);
 if (eventObject.info.name == "ASpt1") {
 my_FLVPlybk.removeASCuePoint("ASpt2");
 trace("Removed cue point ASpt2");
 }
}
```

Raadpleeg `FLVPlayback.removeASCuePoint()` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#).

## Meerdere videobestanden afspelen

U kunt FLV-bestanden achter elkaar afspelen in een instantie van `FLVPlayback` door gewoon een nieuwe URL te laden in de eigenschap `source` wanneer het voorgaande FLV-bestand is afgespeeld. Met de volgende ActionScript-code wordt bijvoorbeeld geluisterd naar de gebeurtenis `complete`, die optreedt wanneer een FLV-bestand is afgespeeld. Wanneer deze gebeurtenis optreedt, worden in de code de naam en locatie ingesteld van een nieuw videobestand in de eigenschap `source` en wordt de methode `play()` aangeroepen om de nieuwe video af te spelen.

```
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");
 }
};
```

## Meerdere videospelers gebruiken

U kunt ook meerdere videospelers openen binnen een enkele instantie van de component `FLVPlayback` en zo meerdere video's afspelen en er tijdens het afspelen tussen schakelen.

U kunt de eerste videospeler maken door de component `FLVPlayback` naar het werkgebied te slepen. De component wijst het nummer 0 automatisch aan de nieuwe videospeler toe en stelt deze in als de standaardspeler. Als u een extra videospeler wilt maken, zet u de eigenschap `activeVideoPlayerIndex` op een ander nummer. Door de eigenschap `activeVideoPlayerIndex` in te stellen, wordt de opgegeven videospeler tevens de *actieve* videospeler. Dat is de videospeler die door de eigenschappen en de methoden van de klasse `FLVPlayback` wordt beïnvloed. Het instellen van de eigenschap `activeVideoPlayerIndex` heeft echter niet tot gevolg dat de videospeler zichtbaar wordt. Als u de videospeler zichtbaar wilt maken, stelt u de eigenschap `visibleVideoPlayerIndex` in op het nummer van de videospeler. Zie voor meer informatie over hoe deze eigenschappen en de methoden en eigenschappen van de klasse `FLVPlayback` elkaar beïnvloeden de eigenschappen `FLVPlayback.activeVideoPlayerIndex` en `FLVPlayback.visibleVideoPlayerIndex` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#).

Met de volgende ActionScript-code wordt de eigenschap `source` geladen om een FLV-bestand in de standaard videospeler af te spelen en een actiepunt toe te voegen. Wanneer de gebeurtenis `ready` optreedt, opent de gebeurtenishandler een tweede videospeler door de eigenschap `activeVideoPlayerIndex` op nummer 1 in te stellen. Hiermee worden een FLV-bestand en actiepunt opgegeven voor de tweede videospeler en wordt de standaardspeler (0) weer de actieve videospeler.

```
/**
 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;
};
```

Als tijdens het afspelen van een FLV-bestand naar een ander FLV-bestand moet worden overgeschakeld, moet u dat in de ActionScript-code opgeven. Met actiepunten kunt u een gebeurtenis `cuePoint` gebruiken om een FLV-bestand op bepaalde punten naar een ander FLV-bestand te laten overschakelen. De volgende code maakt een listener voor de gebeurtenis `cuePoint` en roept tevens een handlerfunctie aan die de actieve videospeler (0) onderbreekt, overschakelt naar de tweede speler (1) en vervolgens het FLV-bestand van de tweede speler afspeelt:

```
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);
 }
};
```

Als u een nieuwe videospeler maakt, stelt de instantie FLVPlayback de eigenschappen van de nieuwe speler in op de waarden van de standaard videospeler, met uitzondering van de eigenschappen `source`, `totalTime`, en `isLive`. Deze eigenschappen worden door de instantie FLVPlayback altijd op de standaardwaarden ingesteld: respectievelijk lege string, 0 en `false`. De eigenschap `autoplay`, die voor de standaard videospeler standaard op `true` staat, wordt op `false` ingesteld. De eigenschap `cuePoints` heeft geen effect, ook niet als het bestand de volgende keer in de standaard videospeler wordt geladen.

De methoden en eigenschappen die het volume, de positionering, de afmetingen, de zichtbaarheid en de gebruikersinterface-elementen bepalen zijn altijd algemeen en hun gedrag wordt niet beïnvloed door het instellen van de eigenschap `activeVideoPlayerIndex`. Zie voor meer informatie over deze methoden en eigenschappen en het effect van het instellen van de eigenschap `activeVideoPlayerIndex` de eigenschap `FLVPlayback.activeVideoPlayerIndex` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#). De resterende eigenschappen en methoden hebben betrekking op de videospeler die door de waarde van de eigenschap `activeVideoPlayerIndex` wordt geïdentificeerd.

De eigenschappen en methoden die de afmetingen bepalen, *beïnvloeden* echter *wel* de eigenschap `visibleVideoPlayerIndex`. Zie de eigenschap `FLVPlayback.visibleVideoPlayerIndex` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#) voor meer informatie.

## FLV-bestanden streamen via Flash Media Server

De vereisten voor het streamen van FLV-bestanden via Flash Media Server variëren. Deze zijn namelijk afhankelijk van het feit of native bandbreedtedetectie beschikbaar is bij uw FVSS-provider (Flash Video Streaming Service). Native bandbreedtedetectie houdt in dat de bandbreedtedetectie in de streamingserver is ingebouwd hetgeen resulteert in betere prestaties. Vraag uw provider of native bandbreedtedetectie beschikbaar is.

Om toegang te krijgen tot FLV-bestanden op een Flash Media Server, gebruikt u een URL zoals `rtmp://my_servername/my_application/stream.flv`.

Bij het afspelen van live streaming met Flash Media Server, moet u de eigenschap `isLive` van FLVPlayback op `true` instellen. Zie de eigenschap `FLVPlayback.isLive` in de [Naslaggids voor ActionScript 3.0 voor Flash Professional](#) voor meer informatie.

Zie de documentatie van Flash Media Server op [www.adobe.com/support/documentation/nl/flashmediaserver/](http://www.adobe.com/support/documentation/nl/flashmediaserver/) voor meer informatie over het beheer van de Flash Media Server, inclusief informatie over het opzetten van een live stream.

## Voor native bandbreedtedetectie of geen bandbreedtedetectie

De klasse `NCManagerNative` is een subklasse van `NCManager` met ondersteuning voor native bandbreedtedetectie. De klasse wordt door sommige FVSS-providers (Flash Video Streaming Service) ondersteund. Als u `NCManagerNative` gebruikt, hebt u geen speciale bestanden op de Flash Media Server nodig. Met `NCManagerNative` kunt u daarnaast zonder het bestand `main.asc` verbinding maken met iedere versie van Flash Media Server, als bandbreedtedetectie niet is vereist.

Als u de klasse `NCManagerNative` in plaats van de standaardklasse `NCManager` wilt gebruiken, moet u de volgende coderegels aan het eerste frame van het FLA-bestand toevoegen:

```
import fl.video*;
VideoPlayer.iNCManagerClass = fl.video.NCManagerNative;
```

## Voor niet-native bandbreedtedetectie

Als native bandbreedtedetectie niet beschikbaar is bij uw FVSS-provider (Flash Video Streaming Service) maar u bandbreedtedetectie wel nodig hebt, moet u het bestand `main.asc` aan de FLV-toepassing van de Flash Media Server toevoegen. Het bestand `main.asc` is online te vinden op [www.adobe.com/go/learn\\_fl\\_samples\\_nl](http://www.adobe.com/go/learn_fl_samples_nl). Het zit in het bestand `Samples.zip` in de map `Samples\ComponentsAS2\FLVPlayback`.

### De Flash Media Server instellen voor streaming FLV-bestanden:

- 1 Maak een map in de toepassingenmap van Flash Media Server en geef de map bijvoorbeeld de naam **my\_application**.
- 2 Kopieer het bestand `main.asc` naar de map `my_application`.
- 3 Maak een map met de naam **streams** in de map `my_application`.
- 4 Maak een map met de naam **\_definst\_** in de map `streams`.
- 5 Plaats uw FLV-bestanden in de map **\_definst\_**.

## De component FLVPlayback aanpassen

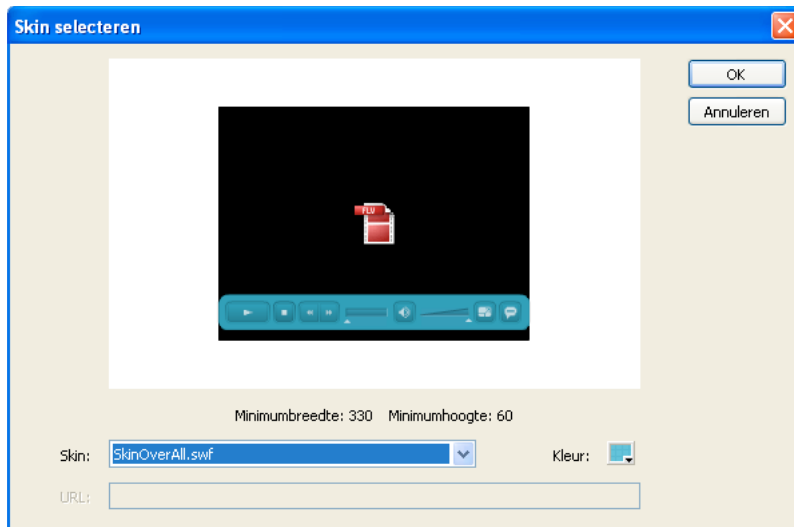
In deze sectie wordt uitgelegd hoe u de component FLVPlayback kunt aanpassen. De meeste methoden die worden gebruikt om andere componenten aan te passen werken niet met de component FLVPlayback. Voor het aanpassen van de component FLVPlayback moet u daarom alleen de technieken gebruiken die in deze sectie worden beschreven.

U hebt de volgende mogelijkheden om de component FLVPlayback aan te passen: een vooraf ontworpen skin selecteren, skins toewijzen aan afzonderlijke aangepaste UI-componenten voor het afspelen van FLV of een nieuwe skin maken. U kunt ook de eigenschappen van FLVPlayback gebruiken om het gedrag van een skin aan te passen.

**Opmerking:** U moet het SWF-bestand met skin tezamen met het SWF-toepassingenbestand naar de webserver uploaden om de skin te laten werken met de component FLVPlayback.

### Een vooraf ontworpen skin selecteren

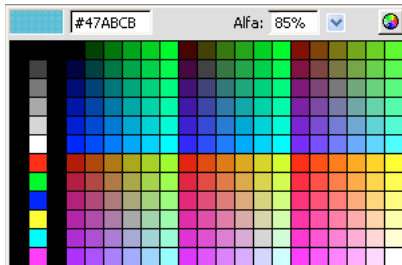
U kunt een skin selecteren voor de component FLVPlayback door in Componentcontrole op de cel `waarde` te klikken voor de parameter `skin`. Vervolgens klikt u op het vergrootglas pictogram om het volgende dialoogvenster van Skin selecteren te openen, waarin u een skin kunt selecteren of de URL kunt opgeven voor het SWF-bestand met skin.



Dialogvenster Skin selecteren voor FLVPlayback

De skins die worden vermeld in het pop-upmenu Skin bevinden zich in de Flash-toepassingsmap /Flash Configuration/FLVPlayback Skins/ActionScript 3.0. U kunt nieuwe skins beschikbaar maken voor dit dialogvenster door deze te maken en het SWF-bestand in de map te zetten. De naam verschijnt in het pop-upmenu met de extensie .swf. Zie “[Een nieuwe skin maken](#)” op pagina 167 voor meer informatie over het maken van een skinset.

Voor de skins die u toewijst door de eigenschap `skin` in te stellen, door tijdens het ontwerp de skinparameter in te stellen of met ActionScript tijdens de uitvoering, kunt u waarden opgeven voor de kleur en het transparantieniveau (alpha) die onafhankelijk zijn van de geselecteerde skin. Als u tijdens het ontwerpen waarden wilt opgeven voor de kleur en de transparantie, opent u de kleurkiezer in het dialogvenster Skin selecteren (zie hieronder).



Kleurkiezer in het dialogvenster Skin selecteren

Wanneer u een kleur wilt selecteren, klikt u op een staal in het deelvenster of geeft u de numerieke waarde van de gewenste kleur op in het tekstvak. Gebruik de schuifregelaar of typ een percentage in het tekstvak Alpha om een waarde voor de transparantie op te geven.

Als u bij uitvoering een waarde voor de kleur en de transparantie wilt opgeven, stelt u de eigenschappen `skinBackgroundColor` en `skinBackgroundAlpha` in. Stel de eigenschap `skinBackgroundColor` in op een waarde `0xRRGGBB` (rood, groen, blauw). Stel de eigenschap `skinBackgroundAlpha` in op een getal tussen 0,0 en 1,0. In het volgende voorbeeld wordt `skinBackgroundColor` op `0xFF0000` (rood) en `skinBackgroundAlpha` op 0,5 ingesteld.

```
my_FLVplybk.skinBackgroundColor = 0xFF0000;
my_FLVplybk.skinBackgroundAlpha = .5;
```

De standaardwaarden zijn de waarden die het laatst door de gebruiker zijn gekozen.

Als u voor de component FLVPlayback een skin wilt toewijzen met de aangepaste UI-componenten voor het afspelen van FLV, selecteert u Geen in het pop-upmenu.

## Skins toewijzen aan afzonderlijke aangepaste UI-componenten voor het afspelen van FLV

Met de aangepaste UI-componenten voor het afspelen van FLV kunt u de weergave van de FLVPlayback-besturingselementen in een FLA-bestand aanpassen. U kunt de resultaten bekijken als u de webpagina bekijkt. Deze componenten kunnen echter niet worden geschaald. U moet de filmclip en de bijbehorende inhoud dan ook voor de gewenste grootte maken. Daarom is het aan te bevelen om de component FLVPlayback in de gewenste grootte in het werkgebied te plaatsen en `scaleMode` op `exactFit` in te stellen.

Sleep om te beginnen de gewenste aangepaste UI-componenten voor het afspelen van FLV vanuit het deelvenster Componenten, plaats ze op de gewenste positie in het werkgebied en geef alle instanties een naam.

Deze componenten werken zonder ActionScript. Als u ze op dezelfde tijdlijn en hetzelfde frame zet als de component FLVPlayback en als er geen skin voor de component is ingesteld, maakt de component FLVPlayback de koppelingen automatisch. Als er meerdere componenten FLVPlayback in het werkgebied zijn, of als het aangepaste besturingselement en de instantie van FLVPlayback zich niet op dezelfde tijdlijn bevinden, moet u ActionScript gebruiken.

Nadat u de componenten in het werkgebied hebt geplaatst, kunt u ze net als alle andere symbolen bewerken. Als u de componenten opent, zult u zien dat ze allemaal hun eigen instellingen hebben.

## De component Button

De component Button heeft een vergelijkbare structuur. De component Button bestaat uit een BackButton, ForwardButton, MuteButton, PauseButton, PlayButton, PlayPauseButton en StopButton. De meeste hebben één filmclip in frame 1 met de instantienaam `placeholder_mc`. Doorgaans is dat een instantie van de normale toestand van de knop, maar dat hoeft niet zo te zijn. In frame 2 zijn er vier filmclips in het werkgebied voor elke weergavetoestand: normaal, over, omlaag en uitgeschakeld. (Bij uitvoering gaat de component nooit echt naar frame 2. De filmclips worden hier geplaatst zodat ze eenvoudig kunnen worden bewerkt en om te zorgen dat ze in het SWF-bestand worden geladen zonder dat de optie Exporteren in eerste frame in het dialoogvenster Symbooleigenschappen hoeft te worden geselecteerd. U moet de optie Exporteren voor ActionScript echter wel selecteren.)

Als u een skin aan een knop wilt toewijzen, kunt u deze filmclips afzonderlijk bewerken. U kunt de grootte en de weergave van de knoppen aanpassen.

In frame 1 verschijnen doorgaans enkele regels ActionScript. U hoeft dit script niet aan te passen. Dit script zorgt ervoor dat de afspreekop in frame 1 wordt gestopt en geeft aan welke filmclip voor welke status moet worden gebruikt.

## De knoppen PlayPauseButton, MuteButton, FullScreenButton en CaptionButton

De knoppen PlayPauseButton, MuteButton, FullScreenButton en CaptionButton worden op een andere manier ingesteld dan de overige knoppen. Zij bestaan slechts uit één frame met twee lagen en geen script. In dat frame bevinden zich twee knoppen, de ene knop boven de andere. Voor PlayPauseButton zijn dat de knoppen Afspelen en Pauzeren; voor MuteButton zijn dat de knoppen Dempen-aan en Dempen-uit; voor FullScreenButton zijn dat de knoppen Volledig scherm-aan en Volledig scherm-uit; voor CaptionButton zijn dat de knoppen Ondertiteling-aan en Ondertiteling-uit. Als u een skin aan elk van deze twee interne knoppen wilt toevoegen, zoals wordt beschreven in [“Skins toewijzen aan afzonderlijke aangepaste UI-componenten voor het afspelen van FLV”](#) op pagina 162; u hoeft verder niets te doen.

De knop CaptionButton heeft betrekking op de component FLVPlaybackCaptioning en moet dan ook aan deze component worden gekoppeld en niet aan de component FLVPlayback.

### De knoppen BackButton en ForwardButton

De knoppen BackButton en ForwardButton worden ook op een andere manier ingesteld dan de overige knoppen. In frame 2 hebben ze extra filmclips die u als een frame kunt gebruiken rondom een van de knoppen of beide knoppen. Deze filmclips zijn niet verplicht en hebben geen speciale mogelijkheden. Zij worden slechts toegevoegd om het bewerken eenvoudiger te maken. Als u ze wilt gebruiken, sleept u ze vanuit het deelvenster Bibliotheek naar de gewenste positie in het werkgebied. Als u ze niet wilt gebruiken, kunt u ze desgewenst uit het deelvenster Bibliotheek verwijderen.

De meeste knoppen zijn gebaseerd op een algemene set filmclips. Dat betekent dat u de weergave van alle knoppen tegelijk kunt wijzigen. U kunt gebruik maken van deze mogelijkheid, maar u kunt deze algemene clips ook vervangen en de weergave per knop instellen.

### component BufferingBar

De component bufferbalk is eenvoudig: Deze bestaat uit een animatie die zichtbaar wordt als de component de bufferstatus bereikt. U hebt geen ActionScript nodig om deze component te configureren. Standaard bestaat deze component uit een gestreepte balk die van links naar rechts beweegt met een rechthoekig masker, om het idee te geven van een gestreepte paal buiten een kapperszaak. De configuratie van deze component is eenvoudig.

Hoewel de bufferbalken in de SWF-bestanden met skin 9-delige schaling gebruiken omdat ze bij uitvoering moeten worden geschaald, maakt de aangepaste UI-component voor het afspelen van FLV met de naam BufferingBar *geen* gebruik van 9-delige schaling omdat de component geneste filmclips bevat. Als u de bufferbalken breder of langer wilt maken, moet u de inhoud wijzigen en geen schaling toepassen.

### De componenten SeekBar en VolumeBar

De componenten SeekBar (zoekbalk) en VolumeBar (volumebalk) zijn vergelijkbaar, maar ze hebben wel verschillende functies. Ze zijn beide voorzien van grepen, gebruiken hetzelfde mechanisme voor bijhouden van de grepen en ondersteunen het gebruik van geneste clips om de voortgang en de volheid bij te houden.

Op veel plaatsen gaat de ActionScript-code in de component FLVPlayback ervan uit dat het registratiepunt (ook wel *uitgangspunt* of *oorsprong*) van de component SeekBar of VolumeBar zich in de linkerbovenhoek van de inhoud bevindt. Het is daarom belangrijk dat u zich hieraan houdt. Anders krijgt u wellicht problemen met de grepen en met de filmclips voor de voortgang en volheid.

Hoewel de zoekbalken in de SWF-bestanden met skin 9-delige schaling gebruiken omdat deze bij uitvoering moeten worden geschaald, maakt de aangepaste UI-component SeekBar *geen* gebruik van 9-delige schaling omdat de component geneste filmclips bevat. Als u de zoekbalken breder of langer wilt maken, moet u de inhoud wijzigen en geen schaling toepassen.

### Greep

In frame 2 bevindt zich een instantie van de filmclip Greep. Net als bij de componenten BackButton en ForwardButton gaat deze component nooit echt naar frame 2. De filmclips worden hier geplaatst zodat ze eenvoudig kunnen worden bewerkt en om ervoor te zorgen dat ze in het SWF-bestand worden geladen zonder dat het selectievakje Exporteren in het dialoogvenster Symbooleigenschappen hoeft te worden geselecteerd. U moet de optie Exporteren voor ActionScript echter wel selecteren.

De filmclip Greep heeft een rechthoek op de achtergrond met de transparantie-instelling 0. Deze rechthoek vergroot het raakgebied van de greep, zodat de greep eenvoudiger te gebruiken is zonder dat de weergave wordt gewijzigd. Dit is vergelijkbaar met de toestand Actief van een knop. Aangezien de greep bij uitvoering dynamisch wordt gemaakt, moet deze een filmclip zijn en geen knop. De rechthoek met de transparantie-instelling 0 heeft verder geen enkele functie en u kunt aan de binnenzijde van de greep iedere gewenste afbeelding plaatsen. De greep werkt het best als u het registratiepunt horizontaal gecentreerd in het midden van de filmclip Greep plaatst.

De volgende ActionScript-code bevindt zich in frame 1 van de component SeekBar voor de besturing van de greep:

```
stop();
handleLinkageID = "SeekBarHandle";
handleLeftMargin = 2;
handleRightMargin = 2;
handleY = 11;
```

Het aanroepen van de functie `stop()` is noodzakelijk vanwege de inhoud van frame 2.

De tweede regel geeft aan welk symbool als greep moet worden gebruikt. U hoeft deze regel niet te wijzigen als u de instantie van de filmclip Greep in frame 2 bewerkt. Bij uitvoering maakt de component FLVPlayback een instantie van de opgegeven filmclip in het werkgebied op hetzelfde niveau als de instantie van de component Bar, hetgeen inhoudt dat ze allebei dezelfde bovenliggende filmclip hebben. Als uw balk zich op het hoofdniveau bevindt, moet de greep zich dus ook op het hoofdniveau bevinden.

De variabele `handleLeftMargin` bepaalt de oorspronkelijke locatie van de greep (0%) en de variabele `handleRightMargin` bepaalt waar de greep eindigt (100%). De getallen bepalen de verschuiving ten opzichte van de linker- en rechterzijde van het besturingselement voor de balk. Een positief getal geeft de limiet aan binnen de balk en een negatief getal geeft de limiet aan buiten de balk. Deze verschuivingen geven aan waar de greep naartoe kan gaan, op basis van het registratiepunt. Als u het registratiepunt in het midden van de greep plaatst, overschrijden de uiterste linker- en rechterzijde van de greep de marges. Voor een zoekbalkfilmclip moet het registratiepunt in de linkerbovenhoek van de inhoud worden geplaatst om de filmclip correct te laten werken.

The variabele `handleY` bepaalt de *y*-positie van de greep, in relatie tot de instantie van de balk. Deze is gebaseerd op de registratiepunten van iedere filmclip. Het registratiepunt in de voorbeeldgreep bevindt zich op de punt van de driehoek om dit relatief ten opzichte het zichtbare gedeelte te plaatsen, waarbij de onzichtbare rechthoek van het raakgebied wordt genegeerd. Voor de balkfilmclip moet het registratiepunt in de linkerbovenhoek van de inhoud worden geplaatst om de filmclip correct te laten werken.

Als het besturingselement van de balk bij deze limieten bijvoorbeeld op (100, 100) is ingesteld en als de balk 100 pixels breed is, kan de greep van 102 tot 198 horizontaal gaan en blijft deze op 111 verticaal. Als u `handleLeftMargin` en `handleRightMargin` wijzigt in -2 en `handleY` in -11, kan de greep van 98 tot 202 horizontaal gaan en blijft deze op 89 verticaal.

### Filmclips voor voortgang en volheid

De component SeekBar heeft een filmclip voor *voortgang* en de component VolumeBar heeft een filmclip voor *volheid*, maar in de praktijk kan iedere SeekBar of VolumeBar een van beide, geen van beide, of beide filmclips hebben. Ze zijn structureel identiek en gedragen zich op dezelfde manier, maar houden verschillende waarden bij. Een filmclip voor voortgang wordt opgevuld als het FLV-bestand bezig is met downloaden (dat is alleen zinvol bij een HTTP-download, omdat de filmclip bij streaming vanaf FMS altijd is opgevuld) en een filmclip voor volheid wordt opgevuld als de greep van links naar rechts beweegt.

De component FLVPlayback vindt deze instanties van de filmclip door naar een specifieke instantienaam te zoeken. De instantie van de filmclip voor voortgang moet daarom de balkfilmclip als bovenliggende filmclip hebben en de instantienaam `progress_mc`. De instantie van de filmclip voor volheid moet de instantienaam `fullness_mc` hebben.

U kunt de filmclips voor voortgang en volheid instellen met of zonder de geneste instantie van de filmclip `fill_mc`. De filmclip `fullness_mc` van de VolumeBar geeft de methode *met* de filmclip `fill_mc` weer en de filmclip `progress_mc` van de SeekBar geeft de methode *zonder* de filmclip `fill_mc` weer.

De methode *met* de geneste filmclip `fill_mc` is handig als u een opvulling wilt gebruiken die niet kan worden geschaald zonder dat de weergave wordt vervormd.



In de filmclip `fullness_mc` van de `VolumeBar` wordt de instantie van de geneste filmclip `fill_mc` gemaskeerd. U kunt de instantie maskeren als u de filmclip maakt of de maskering dynamisch laten plaatsvinden bij uitvoering. Als u de instantie maskeert terwijl u de filmclip maakt, geeft u de instantie de naam `mask_mc` en stelt u de instantie zodanig in dat `fill_mc` wordt weergegeven zoals `fill_mc` zou worden weergegeven al het percentage 100% is. Als u `fill_mc` niet maskeert, wordt het dynamisch gemaakte masker rechthoekig en even groot als `fill_mc` op 100%.

De filmclip `fill_mc` kan met het masker op twee manieren worden weergegeven, afhankelijk van het feit of `fill_mc.slideReveal` op `true` of `false` is ingesteld.

Als `fill_mc.slideReveal` op `true` is ingesteld, wordt `fill_mc` van links naar rechts verplaatst om door het masker zichtbaar te worden gemaakt. Bij 0% wordt `fill_mc` helemaal links geplaatst, zodat er niets door het masker wordt weergegeven. Als het percentage wordt verhoogd, wordt `fill_mc` naar rechts verplaatst totdat 100% is bereikt, waardoor de filmclip weer terug is op de positie waar deze in het werkgebied was gemaakt.

Als `fill_mc.slideReveal` `false` is of niet gedefinieerd (het standaardgedrag), wordt de grootte van het masker van links naar rechts aangepast om meer van `fill_mc` weer te geven. Bij 0% wordt het masker geschaald tot 05 horizontaal. Als het percentage wordt verhoogd, wordt `scaleX` groter totdat 100% is bereikt en `fill_mc` in zijn geheel wordt weergegeven. Dit hoeft niet per definitie `scaleX = 100` te zijn omdat `mask_mc` tijdens het maken kan zijn geschaald.

De methode zonder `fill_mc` is eenvoudiger dan de methode met `fill_mc`, maar de opvulling wordt dan wel horizontaal vervormd. Als u geen vervorming wilt, moet u `fill_mc` gebruiken. De filmclip `progress_mc` van `SeekBar` illustreert deze methode.

De filmclip voor voortgang of volheid wordt horizontaal geschaald op basis van het percentage. Bij 0% is `scaleX` van de instantie op 0 ingesteld, waardoor deze onzichtbaar is. Als het percentage toeneemt, wordt `scaleX` aangepast totdat 100% is bereikt. De clip is dan even groot als toen deze in het werkgebied werd gemaakt. Nogmaals, dit hoeft niet per definitie `scaleX = 100` te zijn omdat de instantie van de clip tijdens het maken kan zijn geschaald.

## Aangepaste UI-componenten voor het afspelen van FLV koppelen

Als u uw aangepaste UI-componenten op dezelfde tijdlijn en in hetzelfde plaatst als de component `FLVPlayback` en u de eigenschap `skin` niet hebt ingesteld, wordt `FLVPlayback` automatisch aan de componenten gekoppeld zonder dat u daarvoor ActionScript hoeft te gebruiken.

Als u meerdere componenten `FLVPlayback` in het werkgebied hebt of als het aangepaste besturingselement en `FLVPlayback` zich niet op dezelfde tijdlijn bevinden, moet u ActionScript-code schrijven om de aangepaste UI-componenten te koppelen aan de instantie van de component `FLVPlayback`. U moet de instantie van `FLVPlayback` een naam geven en vervolgens ActionScript gebruiken om de instanties van de aangepaste UI-component voor het afspelen van FLV aan de bijbehorende eigenschappen van `FLVPlayback` te koppelen. In onderstaand voorbeeld is `my_FLVPlaybk` de instantie van `FLVPlayback`. De namen van de eigenschappen van `FLVPlayback` staan achter de punten (.) en de besturingsinstanties van de aangepaste UI-component voor het afspelen van FLV staan rechts van het isgelijkteken (=):

```
//FLVPlayback instance = my_FLVPlaybk
my_FLVPlaybk.playButton = playbtn; // set playButton prop. to playbtn, etc.
my_FLVPlaybk.pauseButton = pausebtn;
my_FLVPlaybk.playPauseButton = playpausebtn;
my_FLVPlaybk.stopButton = stopbtn;
my_FLVPlaybk.muteButton = mutebtn;
my_FLVPlaybk.backButton = backbtn;
my_FLVPlaybk.forwardButton = forbtn;
my_FLVPlaybk.volumeBar = volbar;
my_FLVPlaybk.seekBar = seekbar;
my_FLVPlaybk.bufferingBar = bufbar;
```

In de volgende stappen worden de aangepaste besturingselementen StopButton, PlayPauseButton, MuteButton en SeekBar gemaakt:

- 1 Sleep de component FLVPlayback naar het werkgebied en geef de instantie de naam **my\_FLVPlybk**.
- 2 Stel de parameter `source` via Componentcontrole in op <http://www.helpexamples.com/flash/video/cuepoints.flv>.
- 3 Zet de parameter Skin op Geen.
- 4 Sleep een StopButton, een PlayPauseButton en een MuteButton naar het werkgebied en plaats ze op de instantie van FLVPlayback. Stapel ze verticaal aan de linkerzijde. Geef iedere knop een instantienaam in Eigenschapcontrole (zoals **my\_stopbtt**n, **my\_plypausbtt**n en **my\_mutebtt**n).
- 5 Open in het deelvenster Bibliotheek de map FLVPlayback Skins en open de onderliggende map SquareButton.
- 6 Selecteer de filmclip SquareBgDown en dubbelklik op de filmclip om deze in het werkgebied te openen.
- 7 Klik met de rechtermuisknop (Windows) of houd de Control-toets ingedrukt en klik (Macintosh) en kies Alles selecteren in het menu. Verwijder het symbool.
- 8 Selecteer het ovaal, teken een ovaal op dezelfde locatie en kies de vulkleur blauw (**#0033FF**).
- 9 Stel in Eigenschapscontrole de breedte (W:) in op **40** en de hoogte (H:) op **20**. Stel the x-coördinaat (X:) in op **0,0** en de -coördinaat (X:) op **0,0**.
- 10 Herhaal de stappen 6 t/m 8 voor SquareBgNormal, maar kies de vulkleur geel (**#FFFF00**).
- 11 Herhaal de stappen 6 t/m 8 voor SquareBgOver, maar kies de vulkleur groen (**#006600**).
- 12 Bewerk de filmclips voor de verschillende symboolpictogrammen in de knoppen (PauseIcon, PlayIcon, MuteOnIcon, MuteOffIcon en StopIcon). U vindt deze filmclips in het deelvenster Bibliotheek onder FLV Playback Skins/Label Button/Assets, waar *Label* de naam van de knop is, bijvoorbeeld Play, Pause, enzovoort. Volg de onderstaande stappen:
  - a Kies de optie Alles selecteren.
  - b Kies de kleur rood (**#FF0000**).
  - c Stel de schaal in op 300%.
  - d Wijzig de locatie X: van de inhoud in **7,0** om de horizontale plaatsing van het pictogram in iedere knoptoestand te wijzigen.

***Opmerking:** Door de locatie op deze manier te wijzigen hoeft u niet iedere knoptoestand afzonderlijk te openen en aan te passen.*
- 13 Klik op de blauwe pijl Terug boven de tijdlijn om terug te keren naar scène 1, frame 1.
- 14 Sleep een component SeekBar naar het werkgebied en plaats deze in de rechterbenedenhoek van de instantie FLVPlayback.
- 15 Dubbelklik in het deelvenster Bibliotheek op de SeekBar om deze in het werkgebied te openen.
- 16 Schaal deze naar 400%.
- 17 Selecteer de omtrek en kies de kleur rood (**#FF0000**).
- 18 Dubbelklik op SeekBarProgress in de map FLVPlayback Skins/Seek Bar en kies de kleur geel (**#FFFF00**).
- 19 Dubbelklik op SeekBarHandle in de map FLVPlayback Skins/Seek Bar en kies de kleur rood (**#FF0000**).
- 20 Klik op de blauwe pijl Terug boven de tijdlijn om terug te keren naar scène 1, frame 1.
- 21 Selecteer de instantie SeekBar in het werkgebied en geef deze de naam **my\_seekbar**.

**De component FLVPlayback gebruiken**

22 Voeg in het deelvenster Handelingen op frame 1 van de tijdlijn een `import` instructie toe voor de klassen Video.

Wijs de naam van de knop en van SeekBar aan de bijbehorende eigenschappen FLVPlayback toe zoals in onderstaand voorbeeld:

```
import fl.video.*;
my_FLVPlybk.stopButton = my_stopbtttn;
my_FLVPlybk.playPauseButton = my_plypausbtttn;
my_FLVPlybk.muteButton = my_mutebtttn;
my_FLVPlybk.seekBar = my_seekbar;
```

23 Druk op Control+Enter om de film te testen.

## Een nieuwe skin maken

De beste manier om een nieuw SWF-bestand met skin te maken is een van de skinbestanden die met Flash worden geleverd te kopiëren en dit te gebruiken als beginpunt. De FLA-bestanden voor deze skins bevinden zich in de Flash-toepassingsmap in Configuration/FLVPlayback Skins/FLA/ActionScript 3.0/. Als u het voltooid SWF-bestand met skin beschikbaar wilt maken als een optie in het dialoogvenster Skin selecteren, plaatst u het bestand in de map Configuration/FLVPlayback Skins/ActionScript 3.0, ofwel in de Flash-toepassingsmap ofwel in een lokale gebruikersmap Configuration/FLVPlayback Skins/ActionScript 3.0.

Aangezien u de kleur van een skin kunt instellen onafhankelijk van het kiezen van een skin, hoeft u het FLA-bestand niet te bewerken om de kleur te wijzigen. Wanneer u een skin maakt met een specifieke kleur en u niet wilt dat deze kan worden bewerkt in het dialoogvenster Skin selecteren, voegt u `this.border_mc.colorMe = false;` toe aan de ActionScript-code van het FLA-bestand met skin. Zie [“Een vooraf ontworpen skin selecteren”](#) op pagina 160 voor informatie over het instellen van de kleur van een skin.

Wanneer u de geïnstalleerde Flash-bestanden met skin (FLA) bekijkt, kunnen bepaalde onderdelen in het werkgebied onnodig lijken. Veel van deze onderdelen worden echter in geleidelagen geplaatst. Met live voorvertoning op schaal 9 kunt u snel zien wat werkelijk wordt weergegeven in het SWF-bestand.

In de volgende secties worden complexere aanpassingen en wijzigingen in de filmclips van zoekbalk, bufferbalk en volumebalk besproken.

## De skinlay-out gebruiken

Wanneer u een Flash-bestand met skin (FLA) opent, worden de lay-outs van de filmclips van de skin weergegeven op de hoofdtijdlijn. Deze clips en de ActionScript-code die zich in hetzelfde frame bevindt, definiëren de lay-out van de besturingselementen bij uitvoering.

Hoewel de lay-outlaag veel lijkt op het uiterlijk van de skin bij uitvoering, is de inhoud van deze laag niet zichtbaar bij uitvoering. Deze laag wordt alleen gebruikt om de plaatsing van de besturingselementen te berekenen. De overige besturingselementen in het werkgebied worden bij uitvoering gebruikt.

In de lay-outlaag bevindt zich een plaatsaanduiding `video_mc` voor de component FLVPlayback. De lay-out van alle overige besturingselementen wordt bepaald ten opzichte van `video_mc`. Wanneer u begint met een van de Flash-bestanden (FLA) en de grootte van de besturingselementen wijzigt, kunt u waarschijnlijk de lay-out bepalen door deze clips voor plaatsaanduiding te verplaatsen.

Elke clip voor plaatsaanduiding heeft een specifieke instantienaam. De namen van de clips voor plaatsaanduiding zijn `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` en `volumeBarHandle_mc`. Het gedeelte dat een nieuwe kleur krijgt wanneer u een skinkleur selecteert, wordt `border_mc` genoemd.

Het is niet van belang welke clip wordt gebruikt voor een besturingselement. Over het algemeen wordt voor knoppen de clip voor de normale toestand gebruikt. Voor overige besturingselementen wordt de clip voor het betreffende besturingselement gebruikt, maar dit is alleen voor het gemak. De enige belangrijke onderdelen zijn de posities  $x$  (horizontaal) en  $y$  (verticaal) en de hoogte en breedte van de plaatsaanduiding.

Naast de standaardbesturingselementen kunt u zo veel aanvullende clips gebruiken als u wenst. De enige vereiste voor deze clips is dat voor de bijbehorende bibliotheeksymbolen de optie Exporteren voor ActionScript in het dialoogvenster Koppelingseigenschappen is geselecteerd. Aangepaste clips in de lay-outlaag kunnen elke instantienaam hebben, behalve de gereserveerde instantienamen die hiervoor zijn opgesomd. Een instantienaam is alleen nodig om ActionScript in de clips in te stellen om de lay-out te bepalen.

De clip `border_mc` is bijzonder. Wanneer u de eigenschap `FLVPlayback.skinAutoHide` op `true` instelt, wordt de skin weergegeven wanneer de muisaanwijzer zich boven de clip `border_mc` bevindt. Dit is belangrijk voor skins die buiten de grenzen van de videospeler worden weergegeven. Zie voor informatie over de eigenschap `skinAutoHide` "[Het gedrag van een skin aanpassen](#)" op pagina 171.

In de Flash-bestanden (FLA) wordt `border_mc` gebruikt voor chroom en voor de randen rondom de knoppen Vorige en Volgende.

De clip `border_mc` is ook het gedeelte van de skin waarvan de alpha en kleur worden gewijzigd door de eigenschappen `skinBackgroundAlpha` en `skinBackgroundColor`. Wanneer u aanpasbare kleur en alpha wilt toestaan, moet de ActionScript-code in het FLA-bestand met skin het volgende bevatten:

```
border_mc.colorMe = true;
```

### ActionScript en skinlay-out

De ActionScript-code hieronder is algemeen van toepassing op alle besturingselementen. Sommige besturingselementen hebben specifieke ActionScript die aanvullend gedrag definieert. Dit wordt beschreven in de sectie voor het betreffende besturingselement.

Het begin van de ActionScript-code bestaat uit een groot gedeelte waarin de klassenamen voor elke status van elke component worden opgegeven. Deze klassenamen bevinden zich alle in het bestand `SkinOverAll fla`. Voor de knoppen Pauzeren en Afspelen ziet de code er bijvoorbeeld als volgt uit:

```
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";
```

De klassenamen hebben geen werkelijke externe klassebestanden en worden alleen opgegeven in het dialoogvenster Koppelingseigenschappen voor alle filmclips in de bibliotheek.

In de ActionScript 2.0-component waren er filmclips in het werkgebied die werkelijk werden gebruikt bij uitvoering. In de ActionScript 3.0-component zijn deze filmclips nog steeds in het FLA-bestand opgenomen, maar alleen om het bewerken ervan te vergemakkelijken. Deze clips bevinden zich nu in geleidelagen en worden niet geëxporteerd. Alle skinelementen in de bibliotheek zijn ingesteld om te worden geëxporteerd op het eerste frame en worden dynamisch gemaakt met bijvoorbeeld de volgende code:

```
new fl.video.skin.PauseButtonDisabled();
```

Na dit gedeelte volgt ActionScript-code die de minimale breedte en hoogte voor de skin definieert. In het dialoogvenster Skin selecteren worden deze waarden weergegeven en deze worden gebruikt bij uitvoering om te voorkomen dat de skin kleiner wordt geschaald dan de minimale grootte. Wanneer u geen minimale grootte wilt opgeven, laat u deze ongedefinieerd of stelt u deze in op kleiner dan of gelijk aan nul.

```
// 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;
```

Op elke plaatsaanduiding kunnen de volgende eigenschappen worden toegepast:

Eigenschap	Beschrijving
anchorLeft	Boolean. Plaatst het besturingselement ten opzichte van de linkerkant van de instantie FLVPlayback. De standaardwaarde is <code>true</code> , tenzij <code>anchorRight</code> expliciet wordt ingesteld op <code>true</code> . In dat geval is de standaardwaarde <code>false</code> .
anchorRight	Boolean. Plaatst het besturingselement ten opzichte van de rechterkant van de instantie FLVPlayback. De standaardwaarde is <code>false</code> .
anchorBottom	Boolean. Plaatst het besturingselement ten opzichte van de onderkant van de instantie FLVPlayback. De standaardwaarde is <code>true</code> , tenzij <code>anchorTop</code> expliciet wordt ingesteld op <code>true</code> . In dat geval is de standaardwaarde <code>false</code> .
anchorTop	Boolean. Plaatst het besturingselement ten opzichte van de bovenkant van de instantie FLVPlayback. De standaardwaarde is <code>false</code> .

Wanneer de eigenschappen `anchorLeft` en `anchorRight` beide `true` zijn, wordt het besturingselement horizontaal geschaald bij uitvoering. Wanneer de eigenschappen `anchorTop` en `anchorBottom` beide `true` zijn, wordt het besturingselement verticaal geschaald bij uitvoering.

Wanneer u de effecten van deze eigenschappen wilt zien, kunt u kijken naar het gebruik ervan in de Flash-skins. De besturingselementen voor bufferbalk en zoekbalk zijn de enige met schaling. Deze worden boven elkaar geplaatst en voor beide zijn de eigenschappen `anchorLeft` en `anchorRight` ingesteld op `true`. Voor alle besturingselementen aan de linkerkant van de bufferbalk en zoekbalk is `anchorLeft` ingesteld op `true` en voor alle besturingselementen aan de rechterkant is `anchorRight` ingesteld op `true`. Voor alle besturingselementen is `anchorBottom` ingesteld op `true`.

U kunt proberen de filmclips in de lay-outlaag te bewerken om een skin te maken waar de besturingselementen zich aan de bovenkant bevinden in plaats van aan de onderkant. U hoeft alleen de besturingselementen aan de bovenkant te verplaatsen ten opzichte van `video_mc` en `anchorTop` in te stellen op `true` voor alle besturingselementen.

## Bufferbalk

De bufferbalk heeft twee filmclips: `bufferingBar_mc` en `bufferingBarFill_mc`. De positie in het werkgebied van elk van beide clips ten opzichte van de andere clip is van belang aangezien deze relatieve plaatsing wordt behouden. De bufferbalk gebruikt twee afzonderlijke clips omdat de component wel `bufferingBar_mc` schaaft, maar niet `bufferingBarFill_mc`.

Op de clip `bufferingBar_mc` wordt 9-delige schaling toegepast, dus de randen worden niet vervormd tijdens het schalen. De clip `bufferingBarFill_mc` is zeer breed, zodat deze altijd breed genoeg is en schalen niet nodig is. De clip wordt bij uitvoering automatisch gemaskeerd zodat alleen het gedeelte boven de uitgerekte `bufferingBar_mc` wordt weergegeven. De exacte afmetingen van het masker houden standaard links en rechts een gelijke marge aan binnen `bufferingBar_mc`, op basis van het verschil tussen de *x*-posities (horizontaal) van `bufferingBar_mc` en `bufferingBarFill_mc`. U kunt de plaatsing aanpassen met ActionScript-code.

Wanneer uw bufferbalk niet hoeft te schalen of geen 9-delige schaling gebruikt, kunt u deze instellen zoals de aangepaste UI-component bufferbalk van FLVPlayback. Zie “[component BufferingBar](#)” op pagina 163 voor meer informatie.

De bufferbalk heeft de volgende aanvullende eigenschap:

Eigenschap	Beschrijving
<code>fill_mc:MovieClip</code>	Geeft de instantienaam aan van de vulling van de bufferbalk. De standaardwaarde is <code>bufferingBarFill_mc</code> .

### Zoekbalk en volumebalk

De zoekbalk heeft ook twee filmclips: `seekBar_mc` en `seekBarProgress_mc`. De positie in de lay-outlaag van elk van beide clips ten opzichte van de andere clip is van belang aangezien deze relatieve plaatsing wordt behouden. Hoewel beide clips kunnen schalen, kan `seekBarProgress_mc` niet worden genest binnen `seekBar_mc`, aangezien `seekBar_mc` 9-delige schaling gebruikt, wat niet goed werkt met geneste filmclips.

Op de clip `seekBar_mc` wordt 9-delige schaling toegepast, dus de randen worden niet vervormd tijdens het schalen. De clip `seekBarProgress_mc` schaalt ook, maar deze kan vervormen. De clip gebruikt geen 9-delige schaling aangezien deze een vulling is, die er goed uitziet wanneer deze wordt vervormd.

De clip `seekBarProgress_mc` werkt zonder `fill_mc`, op nagenoeg dezelfde manier als een clip `progress_mc` clip in aangepaste UI-componenten van FLVPlayback. Met andere woorden, de clip wordt niet gemaskeerd en wordt horizontaal geschaald. De exacte afmetingen van `seekBarProgress_mc` op 100% worden gedefinieerd door linker- en rechtermarges binnen de clip `seekBarProgress_mc`. Deze afmetingen zijn standaard gelijk aan elkaar en gebaseerd op het verschil tussen de *x*-posities (horizontaal) van `seekBar_mc` en `seekBarProgress_mc`. U kunt de afmetingen met ActionScript aanpassen in de filmclip van de zoekbalk, zoals wordt getoond in het volgende voorbeeld:

```
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;
```

U kunt deze code plaatsen op de tijdlijn van de filmclip van de zoekbalk, of samen met de overige ActionScript-code op de hoofdtijdlijn. Wanneer u aanpassingen aanbrengt met code in plaats van de lay-out te wijzigen, hoeft de vulling zich niet in het werkgebied te bevinden. Deze moet zich alleen in de bibliotheek bevinden en zijn ingesteld op Exporteren voor ActionScript in frame 1 met de juiste klassenaam.

Net als met de aangepaste UI-component `SeekBar` van FLVPlayback, is het mogelijk om een volheidsfilmclip voor de zoekbalk te maken. Wanneer uw zoekbalk niet hoeft te schalen, of wel schaalt, maar geen 9-delige schaling gebruikt, kunt u `progress_mc` of `fullness_mc` zo instellen dat deze gebruikmaakt van een van de methoden die worden gebruikt voor aangepaste UI-componenten van FLVPlayback. Zie voor meer informatie.

Aangezien de volumebalk in de Flash-skins niet schaalt, is deze op dezelfde manier opgebouwd als de aangepaste UI-component volumebalk van FLVPlayback. Zie “[De componenten SeekBar en VolumeBar](#)” op pagina 163 voor meer informatie. De uitzondering is dat de greep anders is geïmplementeerd.

### Grepen van zoekbalk en volumebalk

De grepen van de zoekbalk en volumebalk zijn op de lay-out naast de balk geplaatst. De waarden voor de linkermarge, rechtermarge en *y*-as van de greep worden standaard ingesteld door de positie ten opzichte van de filmclip van de balk. De linkermarge wordt ingesteld door het verschil tussen de *x*-locatie (horizontaal) van de greep en de *x*-locatie (horizontaal) van de balk. De rechtermarge is gelijk aan de linkermarge. U kunt deze waarden aanpassen met ActionScript in de filmclip voor de zoekbalk of volumebalk. Het volgende voorbeeld bevat dezelfde ActionScript-code die wordt gebruikt met de aangepaste UI-componenten van FLVPlayback:

```
this.seekBar_mc.handleLeftMargin = 2;
this.seekBar_mc.handleRightMargin = 2;
this.seekBar_mc.handleY = 11;
```

U kunt deze code plaatsen op de tijdlijn van de filmclip van de zoekbalk, of samen met de overige ActionScript-code op de hoofdtijdlijn. Wanneer u aanpassingen aanbrengt met code in plaats van de lay-out te wijzigen, hoeft de greep zich niet in het werkgebied te bevinden. Deze moet zich alleen in de bibliotheek bevinden en zijn ingesteld op Exporteren voor ActionScript in frame 1 met de juiste klassenaam.

Buiten deze eigenschappen zijn grepen eenvoudige filmclips, die op dezelfde manier worden ingesteld als in de aangepaste UI-componenten van FLVPlayback. Beide hebben een rechthoekige achtergrond met de eigenschap `alpha` ingesteld op 0. Deze is alleen aanwezig om het raakgebied te vergroten en is niet verplicht.

### Achtergrond- en voorgrondclips

De filmclips `chrome_mc` en `forwardBackBorder_mc` zijn geïmplementeerd als achtergrondclips.

Van de filmclips `ForwardBackBorder`, `ForwardBorder` en `BackBorder` in het werkgebied en de plaatsaanduidingen voor de knoppen `Vorige` en `Volgende`, is `ForwardBackBorder` het enige onderdeel dat zich niet op een geleidelaag bevindt. Dit bevindt zich alleen in de skins die werkelijk gebruikmaken van de knoppen `Vorige` en `Volgende`.

De enige vereiste voor deze clips is dat deze zijn ingesteld op Exporteren voor ActionScript in frame 1 in de bibliotheek.

### Het gedrag van een skin aanpassen

Met de eigenschappen `bufferingBarHidesAndDisablesOthers` en `skinAutoHide` kunt u het gedrag van een FLVPlayback-skin aanpassen.

Wanneer u de eigenschap `bufferingBarHidesAndDisablesOthers` instelt op `true`, worden de zoekbalk en de greep verborgen door de component FLVPlayback en de knoppen `Pauzeren` en `Afspelen` uitgeschakeld wanneer de bufferstatus van de component wordt ingeschakeld. Dit kan handig zijn wanneer een FLV-bestand bezig is met streamen vanaf FMS bij een trage verbinding met een hoge instelling voor de eigenschap `bufferTime` (bijvoorbeeld 10). In deze situatie zou een ongeduldige gebruiker kunnen gaan zoeken door op de knoppen `Pauzeren` en `Afspelen` te klikken, hetgeen het afspelen van het bestand nog verder kan vertragen. U kunt deze activiteit verhinderen door `bufferingBarHidesAndDisablesOthers` in te stellen op `true` en het element zoekbalk en de knoppen `Pauzeren` en `Afspelen` uit te schakelen terwijl de component zich in de bufferstatus bevindt.

De eigenschap `skinAutoHide` heeft uitsluitend invloed op vooraf ontworpen SWF-bestanden met skin en niet op besturingselementen die zijn gemaakt van de aangepaste UI-componenten van FLVPlayback. Wanneer deze eigenschap is ingesteld op `true`, wordt de skin verborgen door de component FLVPlayback wanneer de muisaanwijzer zich niet boven het weergavegebied bevindt. De standaardwaarde van deze eigenschap is `false`.

## Een SMIL-bestand gebruiken

Voor het afhandelen van meerdere streams voor meerdere bandbreedten gebruikt de klasse VideoPlayer een hulpklasse (NCManager) die een subset van SMIL ondersteunt. SMIL wordt gebruikt voor het identificeren van de locatie van de videostream, de lay-out (breedte en hoogte) van het FLV-bestand en de FLV-bronbestanden die overeenkomen met de verschillende bandbreedten. SMIL kan ook worden gebruikt om de bitsnelheid en duur van het FLV-bestand op te geven.

Gebruik de parameter `source` of de eigenschap `FLVPlayback` (ActionScript) om de locatie van een SMIL-bestand op te geven. Zie en de eigenschap `FLVPlayback.source` in de [Naslaggids voor ActionScript 3.0 voor Adobe Flash Professional CS5](#) voor meer informatie.

In het volgende voorbeeld wordt een SMIL-bestand getoond dat FLV-bestanden met meerdere bandbreedten streamt vanuit een FMS via RTMP:

```
<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>
```

De tag `<head>` kan de tags `<meta>` en `<layout>` bevatten. De tag `<meta>` ondersteunt alleen het attribuut `base`, dat wordt gebruikt om de URL op te geven van de streaming video (RTMP vanuit een FMS).

De tag `<layout>` ondersteunt alleen het element `root-layout`, dat wordt gebruikt om de attributen `height` en `width` in te stellen, en bepaalt daarmee de grootte van het venster waarin het FLV-bestand wordt gerenderd. Deze attributen accepteren alleen pixelwaarden, dus geen percentages.

Binnen de hoofdtekst van het SMIL-bestand kunt u een enkele koppeling naar een FLV-bronbestand opnemen, of kunt u, wanneer u meerdere bestanden voor meerdere bandbreedten streamt vanuit een FMS (zoals in het vorige voorbeeld), de tag `<switch>` gebruiken om een lijst met de bronbestanden weer te geven.

De tags `video` en `ref` binnen de tag `<switch>` zijn gelijk aan elkaar. Beide kunnen het attribuut `src` gebruiken om FLV-bestanden op te geven. Verder kunnen beide tags de attributen `region`, `system-bitrate` en `dur` gebruiken om het gebied, de minimaal vereiste bandbreedte en de duur van het FLV-bestand op te geven.

Binnen de tag `<body>` mag slechts een van de tags `<video>`, `<src>` of `<switch>` voorkomen.

In het volgende voorbeeld wordt een progressieve download getoond voor een enkel FLV-bestand dat geen bandbreedtedetectie gebruikt:



```
<smil>
 <head>
 <layout>
 <root-layout width="240" height="180" />
 </layout>
 </head>
 <body>
 <video src="myvideo.flv" />
 </body>
</smil>
```

## <smil>

### Beschikbaarheid

Flash Professional 8.

### Gebruik

```
<smil>
...
child tags
...
</smil>
```

### Attributen

Geen.

### Onderliggende tags

<head>, <body>

### Bovenliggende tag

Geen.

### Beschrijving

Tag op hoofdniveau, die een SMIL-bestand identificeert.

### Voorbeeld

In het volgende voorbeeld wordt een SMIL-bestand weergegeven dat drie FLV-bestanden opgeeft:

```
<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>

### Beschikbaarheid

Flash Professional 8.

### Gebruik

```
<head>
...
child tags
...
</head>
```

### Attributen

Geen.

### Onderliggende tags

<meta>, <layout>

### Bovenliggende tag

<smil>

### Beschrijving

Ondersteunt de tags <meta> en <layout> en geeft de locatie en standaardlay-out (hoogte en breedte) van de FLV-bronbestanden op.

### Voorbeeld

In het volgende voorbeeld wordt de hoofdlay-out op 240 pixels bij 180 pixels ingesteld:

```
<head>
 <meta base="rtmp://myserver/myapp/" />
 <layout>
 <root-layout width="240" height="180" />
 </layout>
</head>
```

## <meta>

### Beschikbaarheid

Flash Professional 8.

### Gebruik

```
<meta/>
```

### Attributen

`base`

### Onderliggende tags

```
<layout>
```

### Bovenliggende tag

Geen.

### Beschrijving

Bevat het attribuut `base` dat de locatie (RTMP-URL) van de FLV-bronbestanden opgeeft.

### Voorbeeld

In het volgende voorbeeld wordt een tag `meta` voor een basislocatie op `myserver` getoond:

```
<meta base="rtmp://myserver/myapp/" />
```

## <layout>

### Beschikbaarheid

Flash Professional 8.

### Gebruik

```
<layout>
...
child tags
...
</layout>
```

### Attributen

Geen.

### Onderliggende tags

```
<root-layout>
```

### Bovenliggende tag

```
<meta>
```

### Beschrijving

Geeft de breedte en hoogte van het FLV-bestand op.

### Voorbeeld

In het volgende voorbeeld wordt de lay-out ingesteld op 240 pixels bij 180 pixels:

```
<layout>
 <root-layout width="240" height="180" />
</layout>
```

## <root-layout>

### Beschikbaarheid

Flash Professional 8.

### Gebruik

```
<root-layout...attributes.../>
```

### Attributen

width, height

### Onderliggende tags

Geen.

### Bovenliggende tag

```
<layout>
```

### Beschrijving

Geeft de breedte en hoogte van het FLV-bestand op.

### Voorbeeld

In het volgende voorbeeld wordt de lay-out ingesteld op 240 pixels bij 180 pixels:

```
<root-layout width="240" height="180" />
```

## <body>

### Beschikbaarheid

Flash Professional 8.

### Gebruik

```
<body>
...
 child tags
...
</body>
```

### Attributen

Geen.

### Onderliggende tags

<video>, <ref>, <switch>

### Bovenliggende tag

<smil>

### Beschrijving

Bevat de tags <video>, <ref> en <switch>, die de naam van het FLV-bronbestand, de minimale bandbreedte en de duur van het FLV-bestand opgeven. Het attribuut `system-bitrate` wordt alleen ondersteund wanneer de tag <switch> wordt gebruikt. Binnen de tag <body> mag slechts één instantie van de tags <switch>, <video> of <ref> voorkomen.

### Voorbeeld

In het volgende voorbeeld worden drie FLV-bestanden opgegeven, waarvan twee de tag `video` gebruiken en één de tag `ref`:

```
<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>

### Beschikbaarheid

Flash Professional 8.

### Gebruik

```
<video...attributes.../>
```

### Attributen

`src`, `system-bitrate`, `dur`

### Onderliggende tags

Geen.

### Bovenliggende tag

<body>

### Beschrijving

Gelijk aan de tag <ref>. Ondersteunt de attributen `src` en `dur`, die de naam van het FLV-bronbestand en de duur ervan opgeven. Het attribuut `dur` ondersteunt de volledige (00:03:00:01) en gedeeltelijke (03:00:01) tijdnotatie.

### Voorbeeld

In het volgende voorbeeld worden de bron en duur voor een video ingesteld:

```
<video src="myvideo_mdm.flv" dur="3:00.1"/>
```

## <ref>

### Beschikbaarheid

Flash Professional 8.

### Gebruik

```
<ref...attributes.../>
```

### Attributen

`src`, `system-bitrate`, `dur`

### Onderliggende tags

Geen.

### Bovenliggende tag

```
<body>
```

### Beschrijving

Gelijk aan de tag `<video>`. Ondersteunt de attributen `src` en `dur`, die de naam van het FLV-bronbestand en de duur ervan opgeven. Het attribuut `dur` ondersteunt de volledige (00:03:00:01) en gedeeltelijke (03:00:01) tijdnotatie.

### Voorbeeld

In het volgende voorbeeld worden de bron en duur voor een video ingesteld:

```
<ref src="myvideo_cable.flv" dur="3:00.1"/>
```

## <switch>

### Beschikbaarheid

Flash Professional 8.

### Gebruik

```
<switch>
...
child tags
...
</switch/>
```

### Attributen

Geen.

### Onderliggende tags

`<video>`, `<ref>`

### Bovenliggende tag

```
<body>
```

### Beschrijving

Wordt gebruikt met de onderliggende tag `<video>` of `<ref>` om een lijst met de FLV-bestanden voor videostreams met meerdere bandbreedten weer te geven. De tag `<switch>` ondersteunt het attribuut `system-bitrate`, dat zowel de minimale bandbreedte als de attributen `src` en `dur` opgeeft.

### Voorbeeld

In het volgende voorbeeld worden drie FLV-bestanden opgegeven, waarvan twee de tag `video` gebruiken en één de tag `ref`:

```
<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>
```

# Hoofdstuk 7: De component FLVPlaybackCaptioning gebruiken

Met de component FLVPlayback kunt u een videospeler in uw Adobe Flash CS5 Professional-toepassing opnemen om gedownloade Adobe Flash-videobestanden (FLV of F4V) en streaming FLV- of F4V-bestanden te spelen. Zie “[De component FLVPlayback gebruiken](#)” op pagina 141 voor meer informatie over FLVPlayback.

Met de component FLVPlaybackCaptioning kunt u ondersteuning van ondertiteling voor uw video's opnemen. De ondertitelingscomponent ondersteunt XML-bestanden met getimede tekst volgens de W3C-standaard en bevat de volgende functies:

**Ondertiteling met ingesloten gebeurtenisactiepunten** U kunt ingesloten gebeurtenisactiepunten in een FLV-bestand met XML koppelen om ondertiteling te bieden in plaats van een XML-bestand met getimede tekst te gebruiken.

**Meerdere FLVPlaybackCaptioning-instanties** U kunt meerdere FLVPlayback-ondertitelingsinstanties maken voor meerdere FLVPlayback-instanties.

**Schakelknop** U kunt gebruikersinteractie met de ondertiteling bieden door een schakelknop voor ondertiteling te gebruiken.

## De component FLVPlaybackCaptioning gebruiken

U kunt de component FLVPlaybackCaptioning met een of meer componenten FLVPlayback gebruiken. In het eenvoudigste scenario sleept u een component FLVPlayback en een component FLVPlaybackCaptioning naar hetzelfde werkgebied, identificeert u de URL van de ondertiteling en stelt u in dat ondertiteling wordt weergegeven. U kunt ook verschillende parameters instellen om de ondertiteling in FLVPlayback aan te passen.

### Ondertiteling toevoegen aan de component FLVPlayback

U kunt de component FLVPlaybackCaptioning aan elke component FLVPlayback toevoegen. Zie “[Een toepassing maken met de component FLVPlayback](#)” op pagina 143 voor meer informatie over het toevoegen van FLVPlayback-onderdelen aan uw toepassing.

#### De component FLVPlaybackCaptioning vanuit het deelvenster Componenten toevoegen:

- 1 Open de map Video in het deelvenster Componenten.
- 2 Sleep (of dubbelklik op) de component FLVPlaybackCaptioning en voeg deze toe aan hetzelfde werkgebied als dat van de component FLVPlayback waaraan u ondertiteling wilt toevoegen.

**Opmerking:** Adobe biedt twee bestanden om u te helpen de component FLVPlaybackCaptioning snel te leren gebruiken: *caption\_video.flv* (een FLVPlayback-voorbeeld) en *caption\_video.xml* (een ondertitelingsvoorbeeld). Deze bestanden vindt u op [www.helpexamples.com/flash/video/caption\\_video.flv](http://www.helpexamples.com/flash/video/caption_video.flv) en [www.helpexamples.com/flash/video/caption\\_video.xml](http://www.helpexamples.com/flash/video/caption_video.xml).


- 3 (Optioneel) Sleep de component CaptionButton naar hetzelfde werkgebied als dat van de componenten FLVPlayback en FLVPlaybackCaptioning. Met de component CaptionButton kan de gebruiker ondertiteling in- en uitschakelen.



**Opmerking:** Sleep de component `CaptionButton` naar hetzelfde werkgebied als dat van de componenten `FLVPlayback` en `FLVPlaybackCaptioning` om deze te kunnen gebruiken.

- 4 Wanneer de component `FLVPlayback` in het werkgebied is geselecteerd, geeft u de volgende vereiste informatie op het tabblad `Parameters` in `Eigenschapcontrole` op:

- Stel `showCaptions` in op `true`.
- Geef de `source` op van het XML-bestand met getimedede tekst dat moet worden gedownload.

 Wanneer u in `Flash` werkt om ondertitels te testen, moet u de eigenschap `showCaptions` instellen op `true`. Als u echter de component `CaptionButton` toevoegt om gebruikers de mogelijkheid te bieden ondertiteling in of uit te schakelen, moet u de eigenschap `showCaptions` instellen op `false`.

Er zijn andere parameters beschikbaar waarmee u de component `FLVPlaybackCaptioning` kunt aanpassen. Zie “[De component FLVPlaybackCaptioning aanpassen](#)” op pagina 191 en de *Naslaggids voor ActionScript® 3.0 voor Adobe® Flash® Professional CS5* voor meer informatie.

- 5 Selecteer `Besturing > Film testen` om de video te starten.

### Op een dynamische manier een instantie maken met ActionScript:

- 1 Sleep de component `FLVPlayback` van het deelvenster `Componenten` naar het deelvenster `Bibliotheek` (`Windows > Bibliotheek`).
- 2 Sleep de component `FLVPlaybackCaptioning` van het deelvenster `Componenten` naar het deelvenster `Bibliotheek`.
- 3 Voeg de volgende code toe aan het deelvenster `Handelingen` in frame 1 van de tijdlijn.

```
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 Wijzig `install_drive` in het station waarop u `Flash` hebt geïnstalleerd en wijzig het pad om de locatie van de map `Skins` voor uw installatie aan te duiden.

**Opmerking:** Wanneer u een instantie `FLVPlayback` maakt met `ActionScript`, moet u er ook dynamisch een skin aan toewijzen door de eigenschap `skin` met `ActionScript` in te stellen. Wanneer u een skin toepast met `ActionScript`, wordt deze niet automatisch met het `SWF`-bestand gepubliceerd. Kopieer de `SWF`-bestanden van de skin en van de toepassing naar uw server, anders is het `SWF`-bestand van de skin niet beschikbaar wanneer de gebruiker deze uitvoert.

## De FLVPlaybackCaptioning-componentparameters instellen

Als u een component verder wilt aanpassen, kunt u de volgende parameters in Eigenschapcontrole of Componentcontrole instellen voor elke instantie van de component FLVPlaybackCaptioning. De eigenschappen worden in de volgende lijst genoemd en beknopt beschreven:

**autoLayout** Hiermee wordt bepaald of de component FLVPlaybackCaptioning de grootte van het ondertitelingsgebied bepaalt. De standaardwaarde is `true`.

**captionTargetName** Hiermee wordt de instantienaam geïdentificeerd van een tekstveld dat of filmclip die ondertiteling bevat. De standaardwaarde is `auto`.

**flvPlaybackName** Hiermee wordt de instantienaam van de FLVPlayback geïdentificeerd die u wilt ondertitelen. De standaardwaarde is `auto`.

**simpleFormatting** Hiermee worden de indelingsinstructies van het XML-bestand met getimedede tekst beperkt wanneer het is ingesteld op `true`. De standaardwaarde is `false`.

**showCaptions** Hiermee wordt bepaald of ondertiteling wordt weergegeven. De standaardwaarde is `true`.

**source** Hiermee wordt de locatie van het XML-bestand met getimedede tekst geïdentificeerd.

Zie de *Naslaggids voor ActionScript® 3.0 voor Adobe® Flash® Professional CS5* voor meer informatie over alle parameters voor FLVPlaybackCaptioning.

### De parameter source opgeven

U kunt de parameter `source` gebruiken om de naam en locatie op te geven van het XML-bestand met getimedede tekst dat de ondertiteling van uw film bevat. Voer het URL-pad rechtstreeks in de broncel in Componentcontrole in.

### Ondertiteling weergeven

U moet de parameter `showCaptions` instellen op `true` om de ondertiteling weer te geven.

Zie de *Naslaggids voor ActionScript® 3.0 voor Adobe® Flash® Professional CS5* voor meer informatie over alle parameters voor de component FLVPlaybackCaptioning.

In de vorige voorbeelden hebt u geleerd hoe u de component FLVPlaybackCaptioning kunt maken en inschakelen om ondertiteling weer te geven. Er zijn twee bronnen die u voor de ondertiteling kunt gebruiken: (1) een XML-bestand met getimedede tekst dat de ondertitels bevat of (2) een XML-bestand met ondertitelingstekst die u kunt koppelen aan ingesloten gebeurtenisactiepunten.

## Ondertiteling met getimedede tekst gebruiken

Met de component FLVPlaybackCaptioning kunt u ondertiteling aan de gekoppelde component FLVPlayback toevoegen door een XML-bestand met getimedede tekst (TT) te downloaden. Raadpleeg de pagina's over het Timed Text Authoring Format op [www.w3.org](http://www.w3.org) voor meer informatie over getimedede tekst.

Deze sectie biedt een overzicht van de ondersteunde Timed Text-tags, de vereiste tags voor de ondertitelingsbestanden en een voorbeeld van een XML-bestand met getimedede tekst. Zie “[Timed Text-tags](#)” op pagina 184 voor meer informatie over alle ondersteunde Timed Text-tags.

De component FLVPlaybackCaptioning ondersteunt de volgende Timed Text-tags:

Categorie	Taak
Ondersteuning voor alineapmaak	Rechts of links uitlijnen of centreren
Ondersteuning van tekstopmaak	<ul style="list-style-type: none"><li>• De grootte van de tekst instellen in absolute pixelgrootte of deltastijl (bijvoorbeeld +2 of -4)</li><li>• Tekstkleur en lettertype instellen</li><li>• Tekst vet of cursief maken</li><li>• Tekstuitvulling instellen</li></ul>
Ondersteuning van andere opmaak	<ul style="list-style-type: none"><li>• De achtergrondkleur van het tekstveld voor ondertiteling instellen</li><li>• De achtergrondkleur van het tekstveld voor ondertiteling op transparant (alpha 0) instellen</li><li>• Tekstomloop van het tekstveld voor ondertiteling instellen (aan of uit)</li></ul>

De component FLVPlaybackCaptioning stemt overeen met de tijdcode van het FLV-bestand. Elke ondertitel moet een attribuut `begin` hebben, dat bepaalt wanneer de ondertitel moet worden weergegeven. Als de ondertitel geen attribuut `dur` of `end` heeft, verdwijnt de ondertitel wanneer de volgende ondertitel wordt weergegeven of wanneer het FLV-bestand wordt beëindigd.

Het volgende voorbeeld toont een XML-bestand met getimedede tekst. Dit bestand (`caption_video.xml`) biedt ondertiteling voor het bestand `caption_video.flv`. Deze bestanden vindt u op [www.helpexamples.com/flash/video/caption\\_video.flv](http://www.helpexamples.com/flash/video/caption_video.flv) en [www.helpexamples.com/flash/video/caption\\_video.xml](http://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 Macromedia
 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 FutureSplash.</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 EU-pin computer,</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 easy
 as drawing on paper.</p>
 </div>
 </body>
</tt>
```

## Timed Text-tags

De component FLVPlaybackCaptioning ondersteunt Timed Text-tags voor het ondertitelen van XML-bestanden. Raadpleeg de pagina's over het Timed Text Authoring Format op [www.w3.org](http://www.w3.org) voor meer informatie over Timed Text-tags voor audio en video. In de volgende tabel worden de tags weergegeven die wel en niet worden ondersteund.

Functie	Tag/waarde	Gebruik/beschrijving	Voorbeeld
Genegeerde tags	metagegevens	Genegeerd / toegestaan op elk niveau van het document	
	set	Genegeerd / toegestaan op elk niveau van het document	
	xml:lang	Genegeerd	
	xml:space	Genegeerd / Gedrag negeert als: xml:space="default"	

Functie	Tag/waarde	Gebruik/beschrijving	Voorbeeld
	layout	Genegeerd / inclusief alle tags region in een tagsectie layout	
	De tag br	Alle attributen en inhoud worden genegeerd.	
Media-timing voor ondertitels	attributen begin	Alleen in tags p toegestaan. Vereist voor implementatie van media-timing voor ondertitels.	<p begin="3s">
	attributen dur	Alleen in tags p toegestaan. Aanbevolen. Indien niet opgenomen eindigt de ondertitel met het FLV-bestand of wanneer de volgende ondertitel begint.	
	attributen end	Alleen in tags p toegestaan. Aanbevolen. Indien niet opgenomen eindigt de ondertitel met het FLV-bestand of wanneer de volgende ondertitel begint.	
Klok-timing voor ondertitels	00:03:00.1	Volledige kloknotatie	
	03:00.1	Gedeeltelijke kloknotatie	
	10	Verschuivingstijden zonder eenheden. Verschuiving staat voor seconden.	
	00:03:00:05 00:03:00:05.1 30f 30t	Niet ondersteund. Tijdsnotaties die frames of tikken bevatten worden niet ondersteund.	
Tags voor hoofdtekst	body	Vereist / Slechts één tag body wordt ondersteund.	<body ><div >... </div ></body>
Tags voor inhoud	tag div	Nul of meer toegestaan. De eerste tag wordt gebruikt.	
	tag p	Nul of meer toegestaan.	
	De tag span	Een logische container voor een reeks eenheden met tekstinhoud. Geneste bereiken worden niet ondersteund. Tags voor attributen style worden ondersteund.	
	De tag br	Geeft een expliciete regelterugloop aan.	
Tags voor stijlen (Alle stijltags worden binnen de tag p gebruikt)	style	Verwijst een of meer stijlelementen. Kan als een tag en als een attribuut worden gebruikt. Als tag, is een attribuut ID vereist (de stijl kan in het document opnieuw worden gebruikt). Een of meer tags style binnenin de tag style worden ondersteund.	
	tts:backgroundColor	Geeft een stijleigenschap op die de achtergrondkleur van een gebied definieert. Alpha wordt genegeerd, tenzij ingesteld op nul (alpha 0) om de achtergrond transparant te maken. De kleurnotatie is #RRGGBBAA.	

Functie	Tag/waarde	Gebruik/beschrijving	Voorbeeld
	tts:color	Geeft een stijleigenschap op die de voorgrondkleur definieert. Alpha wordt voor geen enkele kleur ondersteund. Waarde <code>transparent</code> wordt naar zwart vertaald.	<pre>&lt;style id="3" style="2" tts:backgroundColor="white"/&gt; "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	Geeft een stijleigenschap op die de lettertypefamilie definieert.	<pre>"default" = _serif "monospace" = _typewriter "sansSerif" = _sans "serif" = _serif "monospaceSansSerif" = _typewriter "monospaceSerif" = _typewriter "proportionalSansSerif" = _sans</pre>
	tts:fontSize	Geeft een stijleigenschap op die het lettertype definieert. Alleen de eerste (verticale) waarde wordt gebruikt wanneer er twee zijn opgegeven. Procentuele waarden en eenheden worden genegeerd. Absolute pixelgrootte (bijvoorbeeld 12) en relatieve stijlgrootte (bijvoorbeeld +2) worden ondersteund.	
	tts:fontStyle	Geeft een stijleigenschap op die de lettertypestijl definieert.	<pre>"normal" "italic" "inherit"* * Standaardgedrag; overerft de stijl van de omsluitende tag.</pre>

Functie	Tag/waarde	Gebruik/beschrijving	Voorbeeld
	tts:fontWeight	Geeft een stijleigenschap op die het lettertypedikte definieert.	"normal" "bold" "inherit"* * Standaardgedrag; overerft de stijl van de omsluitende tag.
	tts:textAlign	Geeft een stijleigenschap op die definieert hoe gebieden inline binnen een omsluitend blokgebied moeten worden uitgelijnd.	"left" "right" "center" "start" ("left") "end" ("right") "inherit"* *Overerft de stijl van de omsluitende tag. Als geen tag textAlign is ingesteld, is de standaardinstelling "left".
	tts:wrapOption	Geeft een stijleigenschap op die definieert of automatische regelterugloop wordt toegepast binnen de context van het element dat wordt beïnvloed. Deze instelling beïnvloedt alle alinea's in het ondertitelingselement.	"wrap" "noWrap" "inherit"* *Overerft de stijl van de omsluitende tag. Als geen tag wrapOption is ingesteld, is de standaardinstelling "wrap".
Attributen die niet worden ondersteund	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		

## Actiepunten met ondertiteling gebruiken

Actiepunten maken interactie met een video mogelijk; u kunt bijvoorbeeld het afspelen van een FLV-bestand beïnvloeden of tekst weergeven op bepaalde momenten in de video. Als u geen XML-bestand met getimedede tekst hebt dat u met een FLV-bestand kunt gebruiken, kunt u gebeurtenisactiepunten insluiten in een FLV-bestand en die vervolgens koppelen aan tekst. Deze sectie biedt informatie over de actiepunstandaarden van de component FLVPlaybackCaptioning en een kort overzicht van hoe u die actiepunten voor ondertiteling aan tekst kunt koppelen. Raadpleeg hoofdstuk 16, 'Werken met video,' in *Flash gebruiken* voor meer informatie over hoe u gebeurtenisactiepunten met de wizard Video importeren of de Flash Video Encoder kunt insluiten.

### Actiepunstandaarden voor FLVPlaybackCaptioning

Binnen de metagegevens van het FLV-bestand wordt een actiepunt vertegenwoordigd als een object met de volgende eigenschappen: `name`, `time`, `type` en `parameters`. ActionScript-actiepunten voor FLVPlaybackCaptioning bevatten de volgende attributen:

**name** De eigenschap `name` is een tekenreeks die de toegewezen naam van het actiepunt bevat. De eigenschap `name` moet beginnen met het voorvoegsel `fl.video.caption.2.0.`, dat moet worden gevolgd door een tekenreeks. De tekenreeks is een serie positieve gehele getallen die telkens worden verhoogd om elke naam uniek te houden. Het voorvoegsel bevat het versienummer dat ook overeenkomt met het versienummer van de FLVPlayback. Voor Adobe Flash CS4 en hoger moet u het versienummer instellen op `2.0`.

**time** De eigenschap `time` is de tijd waarop de ondertitel wordt weergegeven.

**type** De eigenschap `type` is een tekenreeks waarvan de waarde 'event' is.

**parameters** De eigenschap `parameters` is een array die de volgende naam-waardeparen ondersteunt:

- **text:String** De tekst met HTML-opmaak voor de ondertiteling. Deze tekst wordt rechtstreeks aan de eigenschap `TextField.htmlText` doorgegeven. De component FLVPlaybackCaptioning ondersteunt een optionele eigenschap `text:n`, die het gebruik van meerdere geluidssporen voor taal ondersteunt. Zie "[Meerdere geluidssporen voor taal met ingesloten actiepunten ondersteunen](#)" op pagina 190 voor meer informatie.
- **endTime:Number** De tijd waarop de ondertitel moet verdwijnen. Als u deze eigenschap *niet opgeeft*, neemt de component FLVPlaybackCaptioning aan dat het geen getal is (NaN) en wordt de ondertiteling weergegeven totdat het FLV-bestand is voltooid (de instantie FLVPlayback verzendt de gebeurtenis `VideoEvent.COMPLETE`). Geef de eigenschap `endTime:Number` op in seconden.
- **backgroundColor:uint** Deze parameter stelt de `TextField.backgroundColor` in. Deze eigenschap is optioneel.
- **backgroundColorAlpha:Boolean** Als de `backgroundColor` een alpha van 0% heeft, stelt de parameter `TextField.background = !backgroundColor` in. Deze eigenschap is optioneel.
- **wrapOption:Boolean** Deze parameter stelt de `TextField.wordWrap` in. Deze eigenschap is optioneel.

### Ondertiteling voor ingesloten gebeurtenisactiepunten

Als u geen XML-bestand met getimedede tekst hebt dat ondertitels voor uw FLV-bestand bevat, kunt u ondertiteling maken door een XML-bestand met ondertitels te koppelen aan ingesloten gebeurtenisactiepunten. In het XML-voorbeeld wordt aangenomen dat u de volgende stappen hebt uitgevoerd om ingesloten gebeurtenisactiepunten in uw video te maken.

- Voeg de gebeurtenisactiepunten toe (volgens de FLVPlaybackCaptioning-standaarden) en codeer de video.
- Sleep in Flash een component FLVPlayback en een component FLVPlaybackCaptioning naar het werkgebied.



- Stel de eigenschappen source van de componenten FLVPlayback en FLVPlaybackCaptioning in (de locaties van het FLV-bestand en het XML-bestand).
- Publiceer.

In het volgende voorbeeld wordt XML in het coderingsprogramma geïmporteerd:

```
<?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>
```

De component FLVPlaybackCaptioning ondersteunt ook meerdere geluidssporen voor taal met ingesloten actiepunten. Zie [“Meerdere geluidssporen voor taal met ingesloten actiepunten ondersteunen”](#) op pagina 190 voor meer informatie.

## Meerdere geluidssporen voor taal met ingesloten actiepunten ondersteunen

De FLVPlaybackCaptioning-eigenschap `track` ondersteunt meerdere geluidssporen voor taal met ingesloten actiepunten, zolang het XML-bestand met getimedede tekst voldoet aan de actiepuntestandaarden van FLVPlaybackCaptioning. (Zie [“Actiepuntestandaarden voor FLVPlaybackCaptioning”](#) op pagina 188 voor meer informatie.) De component FLVPlaybackCaptioning ondersteunt echter niet meerdere geluidssporen voor taal in afzonderlijke XML-bestanden. Als u de eigenschap `track` wilt gebruiken stelt u de eigenschap in op een waarde die niet gelijk is aan 0. Wanneer u bijvoorbeeld de eigenschap `track` instelt op 1 (`track == 1`), zoekt de component FLVPlaybackCaptioning de actiepuntparameters. Wanneer geen overeenkomst wordt gevonden, wordt de eigenschap `text` in de actiepuntparameters gebruikt. Zie de eigenschap `track` in de *Naslaggids voor ActionScript® 3.0 voor Adobe® Flash® Professional CS5* voor meer informatie.

## Meerdere FLV-bestanden met ondertiteling afspelen

U kunt meerdere videospelers openen binnen één instantie van de component FLVPlayback om meerdere video's af te spelen en er tussen te schakelen terwijl ze worden afgespeeld. U kunt ook ondertiteling koppelen aan elke videospeler binnen de component FLVPlayback. Zie [“Meerdere videospelers gebruiken”](#) op pagina 157 voor informatie over hoe u meerdere videospelers opent. Als u ondertiteling wilt gebruiken in meerdere videospelers, maakt u een instantie van de component FLVPlaybackCaptioning voor elke `VideoPlayer` en stelt u de `videoPlayerIndex` van FLVPlaybackCaptioning in op de overeenkomstige index. De `VideoPlayer`-index staat standaard op 0 wanneer er maar één `VideoPlayer` voorkomt.

Het volgende voorbeeld toont code die unieke ondertiteling aan unieke video's toewijst. Wanneer u dit voorbeeld wilt uitvoeren, moet u de fictieve URL's in het voorbeeld vervangen door werkende URL's.

```
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");
```

## De component FLVPlaybackCaptioning aanpassen

Als u de component FLVPlaybackCaptioning snel wilt gebruiken, kunt u de standaardinstellingen van FLVPlaybackCaptioning gebruiken waarmee de ondertiteling rechtstreeks boven de component FLVPlayback wordt geplaatst. Het kan ook zijn dat u de component FLVPlaybackCaptioning wilt aanpassen door de ondertiteling van de video vandaan te plaatsen.

De volgende code toont hoe u dynamisch een object FLVPlayback kunt maken met de schakelknop voor ondertiteling:

- 1 Plaats de component FLVPlayback in het werkgebied op 0,0 en geef deze de instantienaam **player**.
- 2 Plaats de component FLVPlaybackCaptioning in het werkgebied op 0,0 en geef deze de instantienaam **captioning**.
- 3 Plaats de component CaptionButton in het werkgebied.
- 4 In de volgende voorbeeldcode stelt u de variabele `testVideoPath:String` in op een FLV-bestand (met een absoluut of relatief pad).

**Opmerking:** In de voorbeeldcode wordt de variabele `testVideoPath` ingesteld op het Flash-videovoorgebeeld, `caption_video.flv`. Wijzig deze variabele in het pad van de ondertitelingsvideocomponent waaraan u een component `CaptionButton` toevoegt.

- 5 In de volgende voorbeeldcode stelt u de variabele `testCaptioningPath:String` in op een XML-bestand met getimedede tekst (met een absoluut of relatief pad).

**Opmerking:** In de voorbeeldcode wordt de variabele `testCaptioningPath` ingesteld op het XML-bestand met getimedede tekst, `caption_video.xml`. Wijzig deze variabele in het pad van het XML-bestand met getimedede tekst dat ondertitels voor uw video bevat.

- 6 Sla de volgende code op als `FLVPlaybackCaptioningExample.as` in dezelfde map als uw FLA-bestand.
- 7 Stel de DocumentClass in het FLA-bestand in op `FLVPlaybackCaptioningExample`.

```
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;
 }
 }
}
```

Zie de *Naslaggids voor ActionScript® 3.0 voor Adobe® Flash® Professional CS5* voor meer informatie over alle parameters voor FLVPlaybackCaptioning.