

Använda ACTIONSCRIPT® 3.0-komponenter

Juridiska meddelanden

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

Innehåll

Kapitel 1: Inledning

Tänkta användare	1
Systemkrav	1
Om dokumentationen	1
Typografiska konventioner	2
Uttryck som används i handboken	2
Ytterligare resurser	2

Kapitel 2: Om ActionScript 3.0-komponenter

Fördelar med att använda komponenter	3
Komponenttyper	4
Lägga till och ta bort från ett dokument	6
Söka efter komponentens versionsnummer	8
Modell för händelsehantering i ActionScript 3.0	9
Ett enkelt program	9

Kapitel 3: Arbeta med komponenter

Komponentarkitektur	16
Arbeta med komponentfiler	18
Felsöka komponentprogram	20
Ställa in parametrar och egenskaper	20
Biblioteket	21
Ändra storlek på komponenter	22
Direktförhandsvisning	22
Hantera händelser	23
Arbeta med visningslistan	24
Arbeta med FocusManager	26
Arbeta med List-baserade komponenter	28
Arbeta med en DataProvider	28
Arbeta med en CellRenderer	36
Göra komponenter åtkomliga	43

Kapitel 4: Använda UI-komponenterna

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

Innehåll

Använda komponenten ScrollPane	78
Använda komponenten Slider	81
Använda komponenten TextArea	84
Använda komponenten TextInput	87
Använda komponenten TileList	90
Använda komponenten UILoader	93
Använda komponenten UIScrollBar	95
 Kapitel 5: Anpassa UI-komponenterna	
Om anpassning av UI-komponenter	98
Ställa in stilar	98
Om skal	101
Anpassa komponenten Button	103
Anpassa komponenten CheckBox	105
Anpassa komponenten ColorPicker	107
Anpassa komponenten ComboBox	108
Anpassa komponenten DataGrid	111
Anpassa komponenten Label	115
Anpassa komponenten List	116
Anpassa komponenten NumericStepper	118
Anpassa komponenten ProgressBar	120
Anpassa komponenten RadioButton	121
Anpassa komponenten ScrollPane	123
Anpassa komponenten Slider	124
Anpassa komponenten TextArea	126
Anpassa komponenten TextInput	128
Anpassa komponenten TileList	129
Anpassa komponenten UILoader	131
Anpassa komponenten UIScrollBar	131
 Kapitel 6: Använda komponenten FLVPlayback	
Använda FLVPlayback-komponenten	134
Anpassa komponenten FLVPlayback	152
Använda en SMIL-fil	162
 Kapitel 7: Använda komponenten FLVPlayback Captioning	
Använda komponenten FLVPlaybackCaptioning	170
Använda Timed Text-bildtexter	172
Använda referenspunkter med bildtexter	177
Spela upp flera videofiler med bildtext	180
Anpassa komponenten FLVPlaybackCaptioning	180

Kapitel 1: Inledning

Adobe® Flash® CS5 Professional är standardutvecklingsverktyget för framställning av högkvalitativa webbplatser. Komponenterna är byggstenarna för de högfunktionella Internet-program som skapar webbplatserna. En *komponent* är ett filmklipp med parametrar som du använder för att anpassa komponenten vid utvecklingen i Flash eller vid körning med metoder, egenskaper och händelser i Adobe® ActionScript®. Komponenterna är utformade så att utvecklarna kan återanvända och dela kod, och kapsla in komplexa funktioner som designers kan använda och anpassa utan att använda ActionScript.

Med komponenter kan du snabbt och enkelt bygga robusta program med konsekvent utseende och beteende. I den här handboken beskrivs hur du skapar program med Adobe ActionScript 3.0-komponenter. I *Språkreferens för komponenter i Adobe® ActionScript® 3.0* beskrivs alla komponents API (Application Programming Interface).

Du kan använda komponenter som har skapats av Adobe®, ladda ned komponenter som har skapats av andra utvecklare eller skapa egna komponenter.

Tänkta användare

Handboken är avsedd för utvecklare som skapar Flash-program och vill använda komponenter för att det ska gå fortare. Du bör vara van vid att utveckla program i Flash och att skriva ActionScript.

Om du inte är så van vid att skriva ActionScript kan du lägga till komponenter i ett dokument, ställa in parametrarna i egenskapsinspektören eller komponentinspektören och hantera händelserna i beteendepanelen. Du kan till exempel koppla ett Gå till webbsida-beteende till en Button-komponent som öppnar en URL i en webbläsare när någon klickar på knappen, utan att skriva någon ActionScript-kod.

Om du är programmerare och vill skapa mer robusta program kan du skapa komponenter dynamiskt, använda ActionScript för att ställa in egenskaper och anropa metoder vid körning, och hantera händelser med händelseavlyssnarmodellen.

Mer information finns i ”[Arbeta med komponenter](#)” på sidan 16.

Systemkrav

Flash-komponenter har inga systemkrav utöver systemkraven för Flash.

Alla SWF-filer som använder Flash CS3-komponenter eller senare måste visas med Adobe® Flash® Player 9.0.28.0 eller senare och måste publiceras för ActionScript 3.0 (du kan ange det här via Arkiv > Publiceringsinställningar, på fliken Flash).

Om dokumentationen

I dokumentet förklaras ingående hur du utvecklar Flash-program med komponenter. Du förutsätts ha allmän kunskap om Flash och ActionScript 3.0. Specifik dokumentation om Flash och relaterade produkter finns tillgängliga separat.

Det här dokumentet finns som en PDF-fil och som online-hjälp. Om du vill se onlinehjälp startar du Flash och väljer Hjälp > Flash-hjälp > Använda ActionScript 3.0-komponenter.

Information om Flash finns i följande dokument:

- *Använda Flash*
- *Utvecklarhandbok för ActionScript 3.0*
- *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*

Typografiska konventioner

Följande typografiska konventioner används i handboken:

- *Kursiv anger ett värde som ska bytas ut (till exempel i en katalogsväg).*
- *Kod anger ActionScript-kod, inklusive namn på metoder och egenskaper.*
- *Kursiv kod anger ett kodobjekt som ska bytas ut (till exempel en ActionScript-parameter).*
- **Fetstil anger ett värde som du ska ange.**

Uttryck som används i handboken

Följande uttryck används i handboken:

vid körning När koden körs i Flash Player.

vid redigering När du arbetar i utvecklingsmiljön i Flash.

Ytterligare resurser

Förutom innehållet i dessa handböcker ger Adobe regelbundet ut uppdaterade artiklar, designidéer och exempel på Adobe Developer Center och Adobe Design Center.

Ytterligare komponentexempel finns på www.adobe.com/go/learn_fl_samples_se.

Adobe Developer Center

Adobe Developer Center är en resurs för uppdaterad information om ActionScript, artiklar om verklig programutveckling och information om viktiga frågor. Mer information om Developer Center finns på www.adobe.com/go/flash_devcenter_se.

Adobe Design Center

Lär dig det senaste inom digital design och rörlig grafik. Titta igenom arbeten av ledande artister, upptäck nya designtrender och öka dina kunskaper med självstudiekurser, arbetsflöden och avancerad teknik. Gå dit ett par gånger i månaden och titta efter nya kurser och artiklar samt inspirerande galleriobjekt. Mer information om Design Center finns på www.adobe.com/go/fl_designcenter_se.

Kapitel 2: Om ActionScript 3.0-komponenter

Adobe® Flash® Professional CS5-komponenterna är filmklipp med parametrar som gör att du kan ändra deras utseende och beteende. En komponent kan vara en enkel gränssnittskontroll, t.ex. en RadioButton eller en CheckBox, eller ha innehåll som en List eller DataGrid.

Med komponenter kan du snabbt och enkelt bygga robusta Flash-program med konsekvent beteende och utseende. I stället för att skapa anpassade knappar, kombinationsrutor och listor kan du använda Flash-komponenterna som implementerar de här kontrollerna. Du drar dem helt enkelt från panelen Komponenter till programdokumentet. Du kan också enkelt anpassa komponenternas grafiska utseende och känsla så att de passar programdesignen.

Du kan göra allt det här utan djupare kunskap i ActionScript, men du kan också använda ActionScript 3.0 för att ändra en komponents beteende eller implementera nytt beteende. Varje komponent har en unik uppsättning ActionScript-metoder, -egenskaper och -händelser som utgör dess API (Application Programming Interface). Med API kan du skapa och ändra komponenter medan programmet körs.

Med API kan du också skapa nya, anpassade komponenter. Du kan ladda ned komponenter som andra Flash-användare har skapat på Adobe Exchange på www.adobe.com/go/flash_exchange_se. Mer information om att skapa en komponent finns på www.adobe.com/go/learn_fl_creating_components_se.

Komponentarkitekturern ActionScript 3.0 innehåller klasser som alla komponenter baseras på, skal och format som du använder för att anpassa utseenden, en modell för händelsehantering, fokushantering, ett hjälpmedelsgränssnitt och mycket mer.

***Obs!** Adobe Flash CS5 innehåller både ActionScript 2.0-komponenter och ActionScript 3.0-komponenter. Du kan inte blanda dessa två uppsättningar med komponenter. Du måste använda en av uppsättningarna för ett program. Flash CS5 visar antingen ActionScript 2.0- eller ActionScript 3.0-komponenter baserat på om du öppnar en ActionScript 2.0- eller en ActionScript 3.0-fil. När du skapar ett nytt Flash-dokument måste du ange antingen Flash-fil (ActionScript 3.0) eller Flash-fil (ActionScript 2.0). När du öppnar ett befintligt dokument undersöker Flash publiceringsinställningarna för att avgöra vilken uppsättning med komponenter som ska användas. Information om ActionScript 2.0-komponenter finns i Använda Adobe® ActionScript® 2.0-komponenter.*

En fullständig lista med Flash ActionScript 3.0-komponenter finns i ”Komponenttyper” på sidan 4.

Fördelar med att använda komponenter

Med komponenter kan du separera processen att utforma ditt program från kodningsprocessen. De gör att utvecklare kan skapa funktionalitet som designers kan använda i program. Utvecklare kan kapsla in ofta använda funktioner i komponenter, och designers kan anpassa komponenternas storlek, plats och beteende genom att ändra deras parametrar. De kan också ändra komponenternas utseende genom att redigera deras grafiska element, eller skal.

Komponenterna delar grundfunktioner som format, skal och fokushantering. När du lägger till den första komponenten i ett program upptar de här grundfunktionerna omkring 20 kilobyte av dess storlek. När du lägger till andra komponenter delas den här initiala minnesallokeringen av de komponenter som läggs till, och på så sätt minskas programmets storleksökning.

I det här avsnittet beskrivs några av fördelarna med ActionScript 3.0-komponenterna.

Funktionerna i ActionScript 3.0 ger ett kraftfullt, objektorienterat programmeringsspråk som är ett viktigt steg i utvecklingen av Flash Player-funktioner. Språket är utformat för att bygga högfunktionella Internet-program på en grund av återanvändbar kod. ActionScript 3.0 är baserat på ECMAScript, det internationella och standardiserade skriptspråket, och är kompatibelt med ECMAScript (ECMA-262) version 3-specifikationen. Du hittar en grundlig introduktion till ActionScript 3.0 i *Utvecklarhandbok för ActionScript 3.0*. Referensinformation om språket finns i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

FLA-baserade komponenter för användargränssnitt ger enkel åtkomst till skal för enkel anpassning vid utveckling. De här komponenterna tillhandahåller också format, inklusive skalformat, som du använder till att anpassa olika aspekter av komponenternas utseende, och till att läsa in skal vid körning. Du hittar mer information i ["Anpassa UI-komponenterna"](#) på sidan 98 och i [Referenshandbok för ActionScript® 3.0 i Adobe Flash Professional](#).

Den nya FVLPlayback-komponenten lägger till komponenten FLVPlaybackCaptioning tillsammans med stöd för helskärmsläge, förbättrad direktförhandsvisning, skal som används för att lägga till färg och alfainställningar, samt förbättrade funktioner för FLV-hämtning och layout.

Med egenskapsinspektören och komponentinspektören kan du ändra komponentparametrarna medan du utvecklar i Flash. Mer information finns i ["Arbeta med komponentfiler"](#) på sidan 18 och ["Ställa in parametrar och egenskaper"](#) på sidan 20.

En ny samlingsdialogruta för komponenterna ComboBox, List och TileList gör att du kan fylla i deras egenskap `dataProvider` via användargränssnittet. Mer information finns i ["Skapa en DataProvider"](#) på sidan 29.

Med händelsemodellen ActionScript 3.0 kan ditt program avlyssna händelser och anropa händelsehanterare som svarar. Mer information finns i ["Modell för händelsehantering i ActionScript 3.0"](#) på sidan 9 och ["Hantera händelser"](#) på sidan 23.

Manager-klasser är ett enkelt sätt att hantera fokus och format i ett program. Du hittar mer information i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

UIComponent-basklassen tillhandahåller kärnmetoder, egenskaper och händelser för komponenter som utökar den. Alla komponenter i användargränssnittet till ActionScript 3.0 user ärver från klassen UIComponent. Mer information finns i avsnittet om klassen UIComponent i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Användning av en SWC i de UI FLA-baserade komponenterna tillhandahåller ActionScript-definitioner som objekt i en komponents tidslinje för att påskynda kompileringen.

En klasshierarki som är enkel att utöka med ActionScript 3.0 låter dig skapa unika namnutrymmen, importera klasser som du behöver och enkelt skapa underklasser för att utöka komponenterna.

Du hittar mer information i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Obs! Flash CS5 har stöd för både FLA-baserade och SWC-baserade komponenter. Mer information finns i ["Komponentarkitektur"](#) på sidan 16.

Komponenttyper

Du installerar Flash-komponenterna när du installerar Flash CS5.

ActionScript 3.0-komponenterna innehåller följande komponenter för användargränssnitt (UI):

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

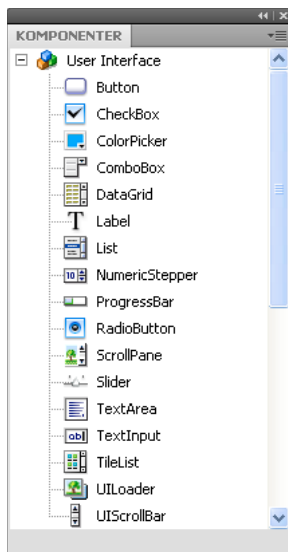
Utöver användargränssnittskomponenterna innehåller Flash ActionScript 3.0-komponenterna följande komponenter och understödjande klasser:

- FLVPlayback-komponenten (fl.video.FLVPlayback), som är en SWC-baserad komponent.
Med komponenten FLVPlayback kan du enkelt ta med en videospelare i Flash-programmet för att spela upp en direktuppspelningsvideo över HTTP, från en FVSS (Adobe® Flash® Video Streaming Service), eller från Adobes Macromedia® Flash® Media Server (FMS). Mer information finns i ”[Använda komponenten FLVPlayback](#)” på sidan 134.
- De anpassade användargränssnittskomponenterna FLVPlayback, som är FLA-baserade och fungerar med både ActionScript 2.0- och ActionScript 3.0-versionerna av FLVPlayback-komponenten. Mer information finns i ”[Använda komponenten FLVPlayback](#)” på sidan 134.
- Komponenter FLVPlayback Captioning, som ger stängd bildtext för FLVPlayback. Se ”[Använda komponenten FLVPlayback Captioning](#)” på sidan 170.
Du hittar en fullständig lista över ActionScript 3.0-komponenter och de klasser som har stöd för dem i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Visa Flash-komponenter:

Följ stegen nedan om du vill visa komponenterna i Flash ActionScript 3.0 i panelen:

- 1 Starta Flash.
- 2 Skapa en ny Flash-fil (ActionScript 3.0) eller öppna ett befintligt Flash-dokument där publiceringsinställningarna anger ActionScript 3.0.
- 3 Välj Fönster > Komponenter för att öppna panelen Komponenter, om den inte redan är öppen.



Komponentpanelen med gränssnittskomponenter

Du kan också hämta ytterligare komponenter från Adobe Exchange på www.adobe.com/go/flash_exchange_se. Om du vill installera komponenter som har hämtats från Exchange hämtar och installerar du Adobe® Extension Manager på www.adobe.com/go/exchange_se. Klicka på länken Home i Adobe Exchange och sök efter länken Extension Manager.

Alla komponenter kan visas i panelen Komponenter i Flash. Följ de här stegen för att installera komponenter på en Windows® - eller Macintosh®-dator.

Installera komponenter på en Windows-baserad dator eller Macintosh-dator:

- 1 Avsluta Flash.
- 2 Placera SWC- eller FLA-filen som innehåller komponenten i följande mapp på hårddisken:
 - I Windows:
C:\Program\Adobe\Adobe Flash CS5\språk\Configuration\Components
 - På Macintosh:
Macintosh-hårddisken:Program:Adobe Flash CS5:Configuration:Components
- 3 Starta Flash.
- 4 Välj Fönster > Komponenter för att visas komponenten i panelen Komponenter, om den inte är öppen redan.
Mer information om komponentfiler finns i ”[Arbeta med komponentfiler](#)” på sidan 18

Lägga till och ta bort från ett dokument

När du drar en FLA-baserad komponent från panelen Komponenter till scenen importerar Flash ett redigerbart filmklipp till biblioteket. När du drar en SWC-baserad komponent till scenen importerar Flash ett komplicerat klipp till biblioteket. När en komponent har importerats till biblioteket kan du dra instanser av den till scenen från antingen bibliotekspanelen eller panelen Komponenter.

Lägga till komponenter vid redigering

Du kan lägga till en komponent i ett dokument genom att dra den från panelen Komponenter. Du kan ställa in egenskaper för varje instans av en komponent i egenskapsinspektören eller på fliken Parametrar i komponentinspektören.

- 1 Välj Fönster > Komponenter.
- 2 Dubbelklicka på komponenten i panelen Komponenter eller dra komponenten till scenen.
- 3 Markera komponenten på scenen.
- 4 Om egenskapsinspektören inte visas väljer du Fönster > Egenskaper > Egenskaper.
- 5 I egenskapsinspektören anger du ett instansnamn komponentinstansen.
- 6 Välj Fönster > Komponentinspektören och välj sedan fliken Parametrar för att ange parametrar för instansen. Mer information finns i [”Ställa in parametrar och egenskaper”](#) på sidan 20.
- 7 Ändra storlek på komponenten genom att redigera värdena för bredden (W:) och höjden (H:). Mer information om att ändra storlek på vissa komponenttyper finns i [”Anpassa UI-komponenterna”](#) på sidan 98.
- 8 Välj Kontroll > Testa filmen eller tryck på Control+Retur för att kompilera dokumentet och se resultatet av inställningarna.

Du kan även ändra färg och textformatering för en komponent genom att ange dess formategenskaper, eller anpassa dess utseende genom att redigera komponentens skal. Mer information om detta finns i [”Anpassa UI-komponenterna”](#) på sidan 98.

Om du drar en komponent till scenen vid redigering kan du referera till dokumentet genom att använda dess instansnamn (till exempel `myButton`).

Lägga till komponenter vid körning med ActionScript

Om du vill lägga till en komponent i ett dokument vid körning med ActionScript måste komponenten först finnas i programmets bibliotek (Fönster > Bibliotek) när SWF-filen kompileras. Om du vill lägga till en komponent i biblioteket drar du den från panelen Komponenter till panelen Bibliotek. Mer information om biblioteket finns i [”Biblioteket”](#) på sidan 21.

Du måste också importera komponentens klassfil för att göra dess API tillgängligt för ditt program. Komponentklassfiler installeras i *paket* som innehåller en eller flera klasser. Om du vill importera en komponentklass använder du programsatsen `import` och anger paketnamnet och klassnamnet. Du importerar till exempel klassen `Button` med följande `import`-sats:

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

Information om vilket paket en komponent tillhör finns i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#). Information om placeringen av komponentkällfiler finns i [”Arbeta med komponentfiler”](#) på sidan 18.

Om du vill skapa en instans av komponenten måste du anropa komponentens ActionScript-konstruktormetod. Följande exempel skapar till exempel en instans av en `Button` med namnet `aButton`:

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

Det sista steget är att anropa den statiska metoden `addChild()` för att lägga till komponentinstansen i scen- eller programbehållaren. Följande programsats lägger till instansen `aButton`:

```
addChild(aButton);
```

Nu kan du använda komponentens API för att dynamiskt ange komponentens storlek och placering på scenen, avlyssna händelser och ange egenskaper för att ändra dess beteende. Mer information om programmeringsgränssnittet för en viss komponent finns i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Mer information om metoden `addChild()` finns i ”[Arbeta med visningslistan](#)” på sidan 24.

Ta bort en komponent

Om du vill ta bort en komponentinstans från scenen vid redigering markerar du den och trycker på Delete. Då tas instansen bort från scenen, men komponenten finns kvar i programmet.

Om du vill ta bort en komponent från Flash-dokumentet när du har placerat den på scenen eller i biblioteket måste du ta bort själva komponenten och dess associerade resurser från biblioteket. Det räcker inte att ta bort komponenten från scenen. Om du inte tar bort den från biblioteket tas den med i programmet när du kompilerar det.

- 1 Välj symbolen för komponenten i panelen Bibliotek.
- 2 Klicka på knappen Ta bort längst ned i panelen Bibliotek, eller välj Ta bort på bibliotekspanelens meny.

Upprepa stegen för att ta bort alla resurser som är associerade med komponenten.

Mer information om att ta bort en komponent från dess behållare när programmet körs finns i ”[Ta bort en komponent från visningslistan](#)” på sidan 26.

Söka efter komponentens versionsnummer

Flash ActionScript 3.0 har en versionsegenskap som du kan visa om du behöver ange den för Adobes tekniska support, eller om du måste veta vilken version av komponenten du använder.

Visa versionsnumret för en användargränssnittskomponent:

- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra komponenten till scenen och ge den ett instansnamn. Dra till exempel en ComboBox till scenen och kalla den `aCb`.
- 3 Tryck på **F9** eller välj Fönster > Åtgärder om du vill öppna panelen Åtgärder.
- 4 Klicka på bildruta 1 på huvudtidslinjen och lägg till följande kod i panelen Åtgärder:

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

Versionsnumret, som liknar det i följande bild, ska visas i utdatapanelen.

För FLVPlayback- och FLVPlaybackCaptioning-komponenterna måste du referera till klassnamnet i stället för instansnamnet, eftersom versionsnumret lagras i en klasskonstant.

Visa versionsnumret för FLVPlayback- och FLVPlaybackCaptioning-komponenter:

- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra komponenterna FLVPlayback och FLVPlaybackCaptioning till bibliotekspanelen.
- 3 Tryck på **F9** eller välj Fönster > Åtgärder om du vill öppna panelen Åtgärder.
- 4 Klicka på bildruta 1 på huvudtidslinjen och lägg till följande kod i panelen Åtgärder:

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

Versionsnumret visas i utdatapanelen.

Modell för händelsehantering i ActionScript 3.0

ActionScript 3.0 introducerar en enda modell för händelsehantering som ersätter de olika mekanismerna för händelsehantering som finns i tidigare versioner av ActionScript. Den nya händelsemodellen bygger på Document Object Model (DOM) Level 3-händelsespecifikationen.

För utvecklare med erfarenhet av att använda metoden `addListener()` i ActionScript 2.0 kan det vara till hjälp att nämna skillnaderna mellan händelseavlyssnarmodellen i ActionScript 2.0 och händelsemodellen i ActionScript 3.0. I listan nedan beskrivs några av de större skillnaderna mellan de två händelsemodellerna:

- Om du vill lägga till händelseavlyssnare i ActionScript 2.0 använder du `addListener()` i vissa fall och `addEventListener()` i andra. I ActionScript 3.0 använder du alltid `addEventListener()`.
- Det finns inget händelseflöde i ActionScript 2.0, vilket betyder att metoden `addListener()` enbart kan anropas för objektet som sänder händelsen. I ActionScript 3.0 kan metoden `addEventListener()` anropas för alla objekt som tillhör händelseflödet.
- I ActionScript 2.0 kan händelseavlyssnare vara funktioner, metoder eller objekt. I ActionScript 3.0 kan bara funktioner och metoder vara händelseavlyssnare.
- Syntaxen `on(event)` fungerar inte längre i ActionScript 3.0, så du kan inte bifoga ActionScript-händelsekod till ett filmklipp. Du kan bara använda `addEventListener()` för att lägga till en händelseavlyssnare.

Följande exempel, som avlyssnar en `MouseEvent.CLICK`-händelse för en `Button`-komponent som heter `aButton`, illustrerar den grundläggande händelsehanterarmodellen i ActionScript 3.0:

```
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);  
}
```

Mer information om händelsehantering i ActionScript 3.0 finns i *Programmering med ActionScript 3.0*. Mer information om händelsehantering för komponenter i ActionScript 3.0 finns i "[Hantera händelser](#)" på sidan 23.

Ett enkelt program

I det här avsnittet beskrivs de olika stegen som du utför för att skapa ett enkelt ActionScript 3.0-program med Flash-komponenter och utvecklingsverktyget i Flash. Exemplet finns både som en FLA-fil med ActionScript-koden på tidslinjen, och som en extern ActionScript-klassfil med en FLA-fil som enbart innehåller komponenterna i biblioteket. Vanligtvis använder du externa klassfiler när du utvecklar större program så att du kan dela koden mellan klasser och program, och så att du kan göra programmen enklare att underhålla. Mer information om programmering med ActionScript 3.0 finns i *Programmering med ActionScript 3.0*.

Utforma programmet

Vårt första exempel på ett program med ActionScript-komponenter är en variant av programmet "Hello World"-programmet, så utformningen är ganska enkel:

- Programmet kallas Greetings.

- Det använder en TextArea för att visa en hälsning som initialt är Hello World.
- Det använder en ColorPicker som gör att du kan ändra textens färg.
- Det använder tre RadioButtons som gör att du kan ställa in textens storlek till liten, större och störst.
- Det använder en ComboBox som gör att du kan välja en annan hälsning i en listruta.
- Programmet använder komponenter från panelen Komponenter och skapar också programelement med ActionScript-kod.

Med hjälp av den här definitionen kan du börja skapa programmet.

Skapa programmet Greetings

Följ stegen nedan för att skapa programmet Greetings genom att använda utvecklingsverktyget i Flash till att skapa en FLA-fil, placera komponenter på scenen och lägga till ActionScript-kod på tidslinjen.

Skapa programmet Greetings i en FLA-fil:

- 1 Välj Arkiv > Nytt.
- 2 I dialogrutan Nytt dokument väljer du Flash-fil (ActionScript 3.0) och klickar på OK.
Ett nytt Flash-fönster öppnas.
- 3 Välj Arkiv > Spara, döp Flash-filen till **Greetings fla** och klicka på knappen Spara.
- 4 I panelen Komponenter i Flash väljer du en TextArea-komponent och drar den till scenen.
- 5 I fönstret Egenskaper, med TextArea markerat på scenen, skriver du **aTa** som namn på instansen och anger följande information:
 - Ange **230** som W-värde (bredd).
 - Ange **44** som H-värde (höjd).
 - Ange **165** som X-värde (vågrät position).
 - Ange **57** som Y-värde (lodrät position).
 - Ange **Hello World!** som textparameter på fliken Parametrar.
- 6 Dra en ColorPicker-komponent till scenen, placera den till vänster om TextArea och ge den instansnamnet **txtCp**. Ange följande information i egenskapsinspektören:
 - Ange **96** som X-värde.
 - Ange **72** som Y-värde.
- 7 Dra tre RadioButton-komponenter till scenen, en i taget, och ge dem instansnamnen **smallRb**, **largerRb** och **largestRb**. Ange följande information för dem i egenskapsinspektören:
 - Ange **100** som W-värde och **22** som H-värde för var och en av dem.
 - Ange **155** som X-värde.
 - Ange **120** som Y-värde för smallRb, **148** för largerRb och **175** för largestRb.
 - Ange **fontRbGrp** som groupName-parameter för var och en av dem.
 - Ange etiketter för dem på fliken Parametrar, med texten **Liten**, **Större**, **Störst**.
- 8 Dra en ComboBox till scenen och ge den instansnamnet **msgCb**. Ange följande information för den i egenskapsinspektören:
 - Ange **130** som W-värde.

- Ange **265** som X-värde.
- Ange **120** som Y-värde.
- På fliken Parametrar anger du **Greetings** som promptparameter.
- Dubbelklicka på textfältet så att parametern dataProvider öppnar dialogrutan Värden.
- Klicka på plustecknet och byt ut etikettens värde mot **Hello World!**
- Upprepa det föregående steget för att lägga till etikettsvärdena **Ha en bra dag!** och **Härlig morgon!**
- Klicka på OK för att stänga dialogrutan Värden.

9 Spara filen.

10 Om den inte redan är öppen, öppnar du åtgärdspanelen genom att trycka på **F9** eller välja Åtgärder på menyn Fönster. Klicka på bildruta 1 på huvudtidslinjen och ange följande kod i panelen Åtgärder:

```
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);
```

De tre första raderna importerar händelseklasserna som programmet använder. En händelse inträffar när en användare interagerar med en av komponenterna. De följande fem raderna registrerar händelsehanterare för händelserna som programmet vill avlyssna. En `click`-händelse inträffar för en `RadioButton` när en användare klickar på den. En `change`-händelse inträffar när en användare väljer en annan färg i `ColorPicker`. En `change`-händelse inträffar för en `ComboBox` när en användare väljer en annan hälsning i listrutan.

Den fjärde raden importerar klassen `RadioButtonGroup` så att programmet kan tilldela en händelseavlyssnare till gruppen med `RadioButtons`, i stället för att tilldela avlyssnaren till varje enskild knapp.

11 Lägg till följande rad med kod i åtgärdspanelen för att skapa `TextFormat`-objektet `tf`, som programmet använder för att ändra textens formategenskaper för `storlek` och `färg` i `TextArea`.

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

12 Lägg till följande kod för att skapa händelsehanteringsfunktionen `rbHandler`. Den här funktionen hanterar en `click`-händelse när en användare klickar på en av `RadioButton`-komponenterna.

```
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);
}
```

Den här funktionen använder en `switch`-sats för att undersöka egenskapen `target` i `event`-objektet för att avgöra vilken `RadioButton` som utlöste händelsen. Egenskapen `currentTarget` innehåller namnet på objektet som utlöste händelsen. Beroende på vilken `RadioButton` som användaren klickade på ändrar programmet storleken på texten i `TextArea` till 14, 18 eller 24 punkter.

- 13 Lägg till följande kod för att implementera funktionen `cpHandler()` som hanterar en ändring av värdet i `ColorPicker`:

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

Den här funktionen ställer in den färg som valdes i färgväljaren för egenskapen `color` för `TextFormat`-objektet `tf`, och sedan anropar den `setStyle()` för att använda den på texten i `TextArea`-instansen `aTa`.

- 14 Lägg till följande kod för att implementera funktionen `cbHandler()` som hanterar en ändring av urvalet i `ComboBox`:

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

Den här funktionen byter helt enkelt ut texten i `TextArea` mot den valda texten i `ComboBox`, `event.target.selectedItem.label`.

- 15 Välj **Kontroll > Testa filmen** eller tryck på **Control+Enter** om du vill kompilera koden och testa programmet `Greetings`.

I följande avsnitt visas hur du skapar samma program med en extern `ActionScript`-klass och en `FLA`-fil som enbart har de komponenter som krävs i biblioteket.

Skapa programmet `Greetings2` med en extern klassfil:

- 1 Välj **Arkiv > Nytt**.
- 2 I dialogrutan **Nytt dokument** väljer du **Flash-fil (ActionScript 3.0)** och klickar på **OK**.
Ett nytt **Flash-fönster** öppnas.
- 3 Välj **Arkiv > Spara**, döpt **Flash-filen** till **`Greetings2 fla`** och klicka på knappen **Spara**.
- 4 Dra alla komponenter nedan från panelen **Komponenter** till biblioteket:
 - `ColorPicker`
 - `ComboBox`
 - `RadioButton`
 - `TextArea`

Den kompillerade `SWF`-filen använder alla dessa resurser, så du måste lägga till dem i biblioteket. Dra komponenterna från den nedersta delen av bibliotekpanelen. När du lägger till de här komponenterna i biblioteket läggs andra resurser (till exempel `List`, `TextInput`, och `UI ScrollBox`) till automatiskt.

- 5 I fönstret **Egenskaper för dokumentklassen** skriver du **`Greetings2`**.

Om **Flash** visar varningsmeddelandet ”ingen definition för dokumentklassen hittades” ignorerar du det. I följande steg definierar du `Greetings2`-klassen. Den här klassen definierar programmets huvudfunktioner.

- 6 Spara filen `Greetings2 fla`.
- 7 Välj **Arkiv > Nytt**.

8 I dialogrutan Nytt dokument väljer du ActionScript-fil och klickar på OK.

Ett nytt skriptfönster öppnas.

9 Lägg till följande kod i skriptfönstret:

```
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() {
```

Skriptet definierar en ActionScript 3.0-klass med namnet Greetings2. Skriptet fungerar på följande sätt:

- Det importerar klasser som vi ska använda i filen. Vanligtvis lägger du till de här `import`-satserna när du refererar till olika klasser i koden, men i det här exemplet importeras alla i ett steg.
- Det deklarerar variabler som representerar de olika typerna av komponentobjekt som vi ska lägga till i koden. En annan variabel skapar `TextFormat`-objektet `tf`.
- Den definierar en konstruktorfunktion, `Greetings2()`, för klassen. Vi ska lägga till rader i den här funktionen, och lägga till andra metoder till klassen i följande steg.

10 Välj Arkiv > Spara, döp Flash-filen till **Greetings2.as** och klicka på knappen Spara.

11 Lägg till följande rader med kod i funktionen `Greeting2()`:

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

Nu ska funktionen se ut så här:

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

12 Lägg till följande rader med kod efter högerparentesen i metoden `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);
}
```

De här raderna gör följande:

- Skapar instanser av komponenterna i programmet.
- Anger varje komponents storlek, position och egenskaper.

- Lägger till varje komponent till scenen med metoden `addChild()`.

13 Lägg till följande kod för metoden `setUpHandlers()` efter högerparentesen i metoden `bldRadioButtons()`:

```
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;
}
}
```

De här funktionerna definierar händelseavlyssnare för komponenterna.

14 Välj Arkiv > Spara för att spara filen.

15 Välj Kontroll > Testa filmen eller tryck på Control+Enter om du vill kompilera koden och testa programmet Greetings2.

Utveckla och kör efterföljande exempel

När du har utvecklat och kört programmet Greetings bör du ha den grundläggande kunskap du behöver för att köra de andra kodexemplen som beskrivs i den här boken. Den relevanta ActionScript 3.0-koden i varje exempel markeras och diskuteras, och du kan klippa ut och klistra in varje exempel i boken i en FLA-fil, kompilera och köra den.

Kapitel 3: Arbeta med komponenter

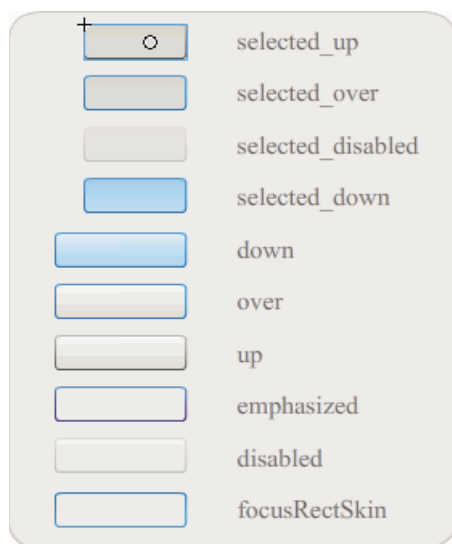
Komponentarkitektur

Adobe® ActionScript® 3.0-komponenter stöds av Adobe® Flash Player version 9.0.28.0 och senare. Komponenterna är inte kompatibla med komponenter som har byggts före Flash CS4. Information om hur du använder Adobe® ActionScript® 2.0-komponenter finns i *Använda Adobe® ActionScript® 2.0-komponenter* och *Språkpreferens för komponenter i Adobe® ActionScript® 2.0*.

Gränssnittskomponenterna (UI) i Adobe ActionScript 3.0 implementeras som FLA-baserade komponenter, men Flash CS5 stöder både SWC- och FLA-baserade komponenter. Komponenterna FLVPlayback och FLVPlaybackCaptioning är till exempel SWC-baserade komponenter. Du kan placera båda typerna av komponenter i mappen Komponenter om du vill att de ska visas i panelen Komponenter. De här två typerna av komponenter är uppbyggda på olika sätt, så de beskrivs var för sig här.

FLA-baserade komponenter i ActionScript 3.0

Användargränssnittskomponenterna i ActionScript 3.0 är FLA-baserade (.fla) filer med inbyggda skal som du når för redigering genom att dubbelklicka på komponenten på scenen. Komponentens skal och andra resurser är placerade i bildruta 2 på tidslinjen. När du dubbelklickar på komponenten går Flash automatiskt till bildruta 2 och öppnar en palett med komponentens skal. I följande bild visas paletten med skal som visas för Button-komponenten.



Skal för komponenten Button

Mer information om komponentskal och hur man anpassar komponenter finns i ["Anpassa UI-komponenterna"](#) på sidan 98 och ["Anpassa komponenten FLVPlayback"](#) på sidan 152.

För att påskynda kompileringen av programmen och undvika konflikter med ActionScript 3.0-inställningarna innehåller de FLA-baserade UI-komponenterna i Flash CS5 också en SWC med komponentens färdigkompileerade ActionScript-kod. ComponentShim-SWC placeras på scenen i bildruta två i alla användargränssnittskomponenter för att de förkompilerade definitionerna ska bli tillgängliga. För att bli tillgänglig för ActionScript måste en komponent antingen finnas på scenen eller i biblioteket med alternativet Exportera i första bildrutan markerat i dess länkningssegenskaper. Om du vill skapa en komponent med ActionScript måste du också importera klassen med en `import`-sats för att kunna få tillgång till den. Information om programsatsen `import` finns i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

SWC-baserade komponenter

SWC-baserade komponenter har också en FLA-fil och en ActionScript-klassfil, men de har kompilerats och exporterats som en SWC. En SWC-fil är ett paket med förkompilerade Flash-symboler och ActionScript-kod som gör att du undviker att kompilera om symboler och kod som inte ändras.

FLVPlayback- och FLVPlaybackCaptioning-komponenterna är SWC-baserade komponenter. De har externa skal i stället för inbyggda skal. FLVPlayback-komponenten har ett standardskal som du kan ändra genom att välja ett skal i en uppsättning med redan utformade skal, genom att anpassa kontroller från UI-kontrollerna i komponentpanelen (BackButton, BufferingBar och så vidare) eller genom att skapa ett anpassat skal. Mer information finns i ”[Anpassa komponenten FLVPlayback](#)” på sidan 152.

I Flash kan du konvertera ett filmklipp till ett kompilerat klipp enligt anvisningarna nedan:

Kompilera ett filmklipp

- Högerklicka (Windows) eller Control-klicka (Macintosh) på filmklippet i bibliotekspanelen och välj Konvertera till sammansatt klipp.

Det kompilerade klippet uppför sig precis som filmklippet som det kompilerades från, men kompilerade klipp visas och publiceras mycket fortare än vanliga filmklipp. Kompilerade klipp kan inte redigeras, men deras egenskaper kan visas i egenskapsinspektören och i komponentinspektören.

SWC-komponenter innehåller ett kompilerat klipp, komponentens förkompilerade ActionScript-definitioner och andra filer som beskriver komponenten. Om du skapar en egen komponent kan du exportera den som en SWC-fil för att distribuera den.

Exportera en SWC-fil

- Välj filmklippet i bibliotekspanelen och högerklicka (Windows) eller tryck på Ctrl samtidigt som du klickar (Macintosh), och välj sedan Exportera SWC-fil.

Obs! *Formatet på en SWC-fil i Flash CS4 eller senare är kompatibelt med Flex SWC-formatet så att SWC-filer kan bytas ut mellan de två produkterna, eventuellt med ändringar.*

Mer information om att skapa SWC-baserade komponenter finns på www.adobe.com/go/learn_fl_creating_components_se.

Komponent-API i ActionScript 3.0

Varje ActionScript 3.0-komponent är byggd på en ActionScript 3.0-klass som finns i en paketmapp och har ett namn med formatet `fl.paketnamn.klassnamn`. Komponenter som Button är till exempel en förekomst av klassen Button och har paketnamnet `fl.controls.Button`. Du måste referera till paketnamnet när du importerar en komponentklass till ditt program. Du importerar klassen Button med följande programsats:

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

Information om placeringen av komponentklassfiler finns i ”[Arbeta med komponentfiler](#)” på sidan 18.

En komponents klass definierar metoderna, egenskaperna, händelserna och formaten som gör att du kan interagera med den i ditt program. Användargränssnittskomponenterna i ActionScript 3.0 är underklasser till klasserna Sprite och UIComponent och ärver egenskaper, metoder och händelser från dem. Klassen Sprite är det grundläggande byggblocket för visningslistor och liknar ett MovieClip, men har ingen Timeline. Klassen UIComponent är basklassen för alla visuella komponenter, både interaktiva och icke-interaktiva. Arvsordningen för varje komponent, och dess egenskaper, metoder, händelser och format, beskrivs i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Alla ActionScript 3.0-komponenter använder händelsehanteringsmodellen i ActionScript 3.0. Mer information om händelsehantering finns i ”[Hantera händelser](#)” på sidan 23 och *Programmering med ActionScript 3.0*.

Arbeta med komponentfiler

I det här avsnittet beskrivs var komponentfilerna lagras, var ActionScript-källfilerna finns och hur du lägger till och tar bort komponenter från komponentpanelen.

Komponentfilernas lagringsplats

Flash-komponenter lagras i konfigurationsmappen på programnivå.

Obs! Information om de här mapparna finns i e ”[Konfigurationsmapor installerade med Flash](#)” i *Använda Flash*.

Komponenterna installeras på följande platser:

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

I komponentmappen finns användargränssnittskomponenterna (UI) i filen User Interface fla, och FLVPlayback- (FLVPlaybackAS3.swc) och FLVPlaybackCaptioning-komponenterna finns i mappen Video.

Du kan också laga komponenterna på följande användarbaserade platser:

- Windows 2000 eller Windows XP: C:\Documents and Settings\användarnamn\Local Settings\Application Data\Adobe\Adobe Flash CS5\en\Configuration\Components
- Windows Vista: C:\User\användarnamn\Local Settings\Application Data\Adobe\Adobe Flash CS5\en\Configuration\Components

Obs! I Windows döljs mappen Application Data som standard. Om du vill visa dolda mappar och filer väljer du Den här datorn för att öppna Utforskaren i Windows och väljer sedan Verktyg > Mappalternativ och fliken Visa. Markera alternativet Visa dolda filer och mappar på fliken Visa.

- Mac OS X: Macintosh-hårddisken:Users:<användarnamn>:Library:Application Support:Adobe Flash CS5:Configuration:Components

Komponentkällfilernas lagringsplats

ActionScript-klassfilerna (.as) (eller *källfilerna*) för komponenter installeras i följande programmappor för Windows 2000 och Windows XP:

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

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

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

För Mac OS X finns komponentkällfilerna här:

Användargränssnittskomponenter Macintosh-hårddisken:Applications:Adobe Flash CS5:Configuration:Component Source>ActionScript 3.0>User Interface:fl

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

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

Komponentkällfiler och Classpath

Eftersom ActionScript 3.0-komponenterna har sin kod inkompilerad ska du inte ange ActionScript-klassfilernas placering i din Classpath-variabel. Om du ändå tar med deras placering i Classpath tar det längre tid att kompilera programmen. Om Flash hittar komponentklassfiler i Classpath-inställningen åsidosätter klassfilen alltid komponentens inkompileerade kod.

Det finns ett tillfälle när du kanske vill lägga till komponentkällfilernas placering i Classpath-inställningen: när du du felsöker ett program med komponenter. Mer information finns i ”[Felsöka komponentprogram](#)” på sidan 20.

Ändra komponentfilerna

Om du måste uppdatera, lägga till eller ta bort SWC-baserade komponenter eller lägga till nya FLA-baserade komponenter i Flash, måste du läsa in dem på nytt i panelen Komponenter för att de ska bli tillgängliga. Du kan läsa in komponenterna på nytt antingen genom att starta om Flash, eller genom att välja Läs in igen på komponentpanelmenyn. Då hämtar Flash alla komponenter som du har lagt till i mappen Komponenter.

Läsa in komponenter i komponentpanelen igen när Flash körs:

- Välj Läs in igen på komponentpanelens meny.

Ta bort en komponent från komponentpanelen:

- Ta bort FLA-, SWC- eller MXP-filen från mappen Komponenter och starta om Flash eller välj Läs in igen på komponentpanelens meny. En MXP-fil är en komponentfil som har laddats ned från Adobe Exchange.

Du kan ta bort eller byta ut SWC-baserade komponenter medan Flash körs, och sedan återspeglas ändringarna när du läser in igen. Om du däremot ändrar eller tar bort FLA-baserade komponenter återspeglas inte ändringarna förrän du avslutar och startar om Flash. Du kan emellertid lägga till FLA-baserade komponenter och läsa in dem med inläsningskommandot.



Adobe rekommenderar att du först kopierar den Flash-komponentfil (.fla eller .as) som du tänker ändra. Sedan kan du återställa den om det behövs.

Felsöka komponentprogram

ActionScript 3.0-komponenterna innehåller all sin källkod för att minska kompileringstiden när du kompilerar dina program. Men Flash-felsökaren kan inte inspektera kod inuti kompilerade klipp. Därför måste du lägga till komponentens källfiler i Classpath-inställningen om du vill felsöka programmet ända in i komponentens källkod.

Placeringen av komponentpaketens mappar är relativ till källfilernas placering för komponenttypen. Om du vill referera till alla ActionScript 3.0-källfilerna för alla UI-komponenter lägger du till följande plats i din Classpath för användargränssnittspaketet:

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

Obs! Det åsidosätter den inkompileerade koden för alla UI-komponenter och ökar programmets kompileringstid. Om du av någon anledning har ändrat en komponents källfil kan komponenten få ett annat beteende.

Om du vill ange Classpath väljer du först Inställningar på menyn Redigera och sedan ActionScript i listan med kategorier. Till sist klickar du på knappen Inställningar för ActionScript 3.0. Om du vill lägga till en ny post klickar du på plustecknet ovanför fönstret som visar de aktuella inställningarna.

Variabeln `$(AppConfig)` refererar till konfigurationsmappen för Flash CS5 på den plats där du installerade Flash CS5. Vanligtvis ser sökvägen ut så här:

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

Obs! Om du måste ändra en komponents källfil rekommenderar Adobe att du kopierar den ursprungliga källfilen till en annan plats och lägger till den platsen i din Classpath.

Information om placeringen av komponentkällfiler finns i ”[Komponentkällfilernas lagringsplats](#)” på sidan 18.

Ställa in parametrar och egenskaper

Alla komponenter har parametrar som du kan ställa in för att ändra deras utseende och beteende. En parameter är en egenskap i komponentens klass och visas i egenskapsinspektören och komponentinspektören. Det mest använda egenskaperna visas som redigeringsparametrar. De övriga måste du ställa in med ActionScript. Alla parametrar som kan ställas in vid redigering kan även ställas in med ActionScript. När du ställer in en parameter med ActionScript åsidosätter den alla värden som har ställts in vid redigering.

De flesta användargränssnittskomponenter i ActionScript 3.0 ärver egenskaper och metoder från klassen `UIComponent` och från en basklass. Klasserna `Button` och `CheckBox` ärver till exempel egenskaper från både klassen `UIComponent` och klassen `BaseButton`. Du har tillgång till både en komponents ärvda egenskaper och dess egna klassegenskaper. Till exempel ärver komponenten `ProgressBar` egenskapen `ProgressBar.enabled` från `UIComponent`, men har också en egen `ProgressBar.percentComplete`-egenskap. Du har tillgång till båda dessa egenskaper när du interagerar med en förekomst av komponenten `ProgressBar`. Mer information om en komponents egenskaper finns tillsammans med informationen om dess klass i [Referenshandbok för ActionScript 3.0](#).

Du kan ställa in parametrar för en förekomst av en komponent med antingen egenskapsinspektören eller komponentinspektören.

Ange ett förekomstnamn för en komponent i egenskapsinspektören:

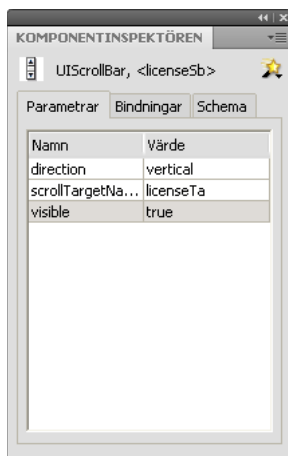
- 1 Välj Fönster°> Egenskaper°> Egenskaper.
- 2 Markera en förekomst av en komponent på scenen.

- 3 Ange ett namn på komponentförekomsten i rutan <Förekomstnamn>, som är placerad nedanför listrutan Filmklipp. Du kan också klicka på fliken Parametrar och ange namnet i rutan nedanför ordet *Komponent*. Ange värden för de parametrar som du vill ställa in.

Det är praktiskt att lägga till ett suffix till förekomstnamnet som anger vilken typ av komponent det är. Det gör det enklare att läsa ActionScript-koden. Förekomstnamnet **licenseSb** identifierar till exempel att komponenten är en rullningslist som rullar ett licensavtal i textområdet **licenseTa**.

Ange parametrar för en komponentförekomst i komponentinspektören:

- 1 Välj Fönster > Komponentinspektören.
- 2 Markera en förekomst av en komponent på scenen.
- 3 Klicka på fliken Parametrar och ange värden för parametrarna i listan.



Komponentparametrar i komponentinspektören

Ställa in komponentegenskaper i ActionScript

I ActionScript använder du en punktoperator (.) (punktsyntax) för att komma åt egenskaper eller metoder som tillhör ett objekt eller en förekomst på scenen. Ett punktsyntaxuttryck börjar med namnet på förekomsten följt av en punkt, och slutar med det element som du vill ange. Följande ActionScript-kod ställer till exempel in egenskapen *width* för CheckBox-förekomsten *aCh* så att den blir 50 pixlar bred:

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

Följande *if*-sats kontrollerar om användaren har markerat kryssrutan:

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

Biblioteket

När du för första gången lägger till en komponent i ett dokument importerar Flash den som ett filmklipp till panelen Bibliotek. Du kan också dra en komponent från komponentpanelen direkt till bibliotekspanelen och sedan lägga till en förekomst av den på scenen. I båda fallen måste du lägga till en komponent i biblioteket innan du kommer åt dess klasselement.

Om du lägger till en komponent i biblioteket och skapar en förekomst av den med ActionScript måste du först importera dess klass med satsen `import`. I satsen `import` måste du ange både komponentens paketnamn och dess klassnamn. Följande sats importerar till exempel klassen `Button`:

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

När du placerar en komponent i biblioteket importerar Flash också en mapp med dess resurser, som innehåller skalerna för dess olika lägen. En komponents *skal* innehåller den samling med symboler som utgör dess grafiska visning i programmet. Ett enda skal är den grafiska representationen, eller filmklippet, som anger ett visst läge för komponenten.

Innehållet i mappen med komponentresurser gör att du kan ändra komponentens skal om du vill. Mer information finns i ”[Anpassa UI-komponenterna](#)” på sidan 98.

När en komponent finns i biblioteket kan du lägga till ytterligare förekomster av den i dokumentet genom att dra dess ikon till scenen från komponentpanelen eller från biblioteket.

Ändra storlek på komponenter

Använd verktyget Omformat fritt eller metoden `setSize()` för att ändra storlek på komponentförekomster. Du kan anropa metoden `setSize()` från en komponentförekomst (se `UIComponent.setSize()`) om du vill ändra storlek på den. Följande kod ändrar storlek på en förekomst av `List`-komponenten till 200 pixlar bred och 300 pixlar hög:

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

Storleken på en komponent ändras inte automatiskt så att den passar på sin etikett. Om en komponentförekomst som har lagts till i ett dokument inte är stor nog för att visa etiketten beskärs etikettens text. Du måste ändra storlek på komponenten så att den passar på etiketten.

Mer information om att ändra storlek på komponenter finns i avsnitten om de enskilda komponenterna i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Direktförhandsvisning

Med funktionen Direktförhandsvisning, som är aktiv som standard, kan du visas komponenter på scenen så som de kommer att visas i det publicerade Flash-innehållet. Komponenterna visas med en ungefärlig storlek.

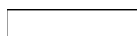
Så här aktiverar och inaktiverar du Direktförhandsvisning:

- Välj **Kontroll > Aktivera direktförhandsvisning**. En bock bredvid alternativet anger att det är aktiverat.

Direktförhandsvisningen visar olika parametrar för olika komponenter. Information om vilka komponentparametrar som visas i direktförhandsvisningen finns i avsnitten om de enskilda komponenterna i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).



En `Button`-komponent med direktförhandsvisning aktiverad



En `Button`-komponent med direktförhandsvisning inaktiverad

Komponenterna i direkthandvisning fungerar inte. Om du vill testa funktionen måste du använda kommandot Kontroll > Testa filmen.

Hantera händelser

Alla komponenter sänder händelser när en användare interagerar med dem. När en användare till exempel klickar på en Button skickar den en `MouseEvent.CLICK`-händelse, och när en användare väljer ett objekt i en List, skickar List en `Event.CHANGE`-händelse. En händelse kan också inträffa när något viktigt händer med en komponent, till exempel när innehållet har lästs in för en `UILoader`-förekomst, vilket genererar en `Event.COMPLETE`-händelse. Om du vill hantera en händelse skriver du ActionScript-kod som körs när händelsen inträffar.

En komponents händelser innehåller händelserna i alla klasser som komponenten ärver från. Det betyder att alla användargränssnittskomponenter i ActionScript 3.0 ärver händelser från klassen `UIComponent`, eftersom det är basklassen för användargränssnittskomponenterna i ActionScript 3.0. En lista med de händelser som en komponent sänder finns i avsnittet med händelser för komponentklassen i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

En fullständig förklaring av händelsehantering i ActionScript 3.0 finns i *Programmering med ActionScript 3.0*.

Om händelseavlyssnare

Följande huvudpunkter gäller för hantering av händelser för ActionScript 3.0-komponenter:

- Alla händelser sänds av en förekomst av en komponentklass. Komponentförekomsten är *utsändaren*.
- Du registrerar en *händelseavlyssnare* genom att anropa metoden `addEventListener()` för komponentförekomsten. Följande kodrad lägger till en avlyssnare för händelsen `MouseEvent.CLICK` i Button-förekomsten `aButton`:

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

Den andra parametern i metoden `addEventListener()` registrerar namnet på funktionen, `clickHandler`, som ska anropas när händelsen inträffar. Den här funktionen kallas också en *återkopplingsfunktion*.

- Du kan registrera flera avlyssnare för en komponentförekomst.

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

- Du kan registrera en avlyssnare för flera komponentförekomster.

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

- Händelsehanterarfunktionen skickas till ett händelseobjekt som innehåller information om händelsetypen och förekomsten som sänder händelsen. Mer information finns i "[Om händelseobjektet](#)" på sidan 24.
- Avlyssnaren förblir aktiv tills programmet avslutas eller du uttryckligen tar bort den med metoden `removeEventListener()`. Följande kodrad tar till exempel bort avlyssnaren för händelsen `MouseEvent.CLICK` för `aButton`:

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

Om händelseobjektet

Händelseobjektet ärver egenskaper från objektklassen Event och har egenskaper som innehåller information om händelsen som inträffade, inklusive egenskaperna `target` och `type` som tillhandahåller viktig information om händelsen:

Egenskap	Beskrivning
<code>type</code>	En sträng som anger händelsens typ.
<code>target</code>	En referens till komponentförekomsten som sänder händelsen.

Om en händelse har ytterligare egenskaper visas de i händelseklassens beskrivning i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Händelseobjektet genereras automatiskt och skickas till händelsehanterarfunktionen när en händelse inträffar.

Du kan använda händelseobjektet inuti funktionen för att komma åt namnet på händelsen som sändes, eller förekomstnamnet på komponenten som sände händelsen. Från förekomstnamnet kan du komma åt andra komponentegenskaper. Följande kod använder till exempel egenskapen `target` för Event-objektet `evtObj` för att komma åt egenskapen `label` för `aButton` och visa den i utdatapanelen:

```
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");
}
```

Arbeta med visningslistan

Alla ActionScript 3.0-komponenter ärver från klassen `DisplayObject` och kan därför komma åt dess metoder och egenskaper för att interagera med visningslistan. *Visningslistan* är hierarkin med visade objekt och visuella element i ett program. Den här hierarkin innehåller följande element:

- Scenen, som är behållaren på den översta nivån
- Visningsobjekt, bland annat former, `MovieClips` och textfält
- Behållare för visningsobjekt, som är särskilda typer av visningsobjekt som kan innehålla underordnade visningsobjekt.

Objektens ordning i visningslistan visar deras djup i den överordnade behållaren. Ett objekts djup refererar till dess position uppifrån och ned eller framifrån och bakåt på scenen eller i dess visningsbehållare. Djupordningen framgår när objekten överlappar, men finns även när de inte överlappar. Alla objekt i visningslistan har ett motsvarande djup på scenen. Om du vill ändra ett objekts djup genom att placera det framför eller flytta det bakom andra objekt, måste du ändra dess position i visningslistan. Objektens standardordning i visningslistan är den ordning som de placerades på scenen i. Position 0 i visningslistan är objektet längst ned i ordningsdjupet.

Lägga till en komponent i visningslistan

Du kan lägga till ett objekt i ett `DisplayObjectContainer`-objekt genom att anropa behållarens `addChild()`- eller `addChildAt()`-metod. När det gäller scenen kan du också lägga till ett objekt i dess visningslista vid redigering genom att skapa det, och när det gäller komponenter, genom att dra dem till scenen från komponentpanelen. Om du vill lägga till ett objekt i en behållare med `ActionScript` skapar du först en förekomst av det genom att anropa dess konstruktor med operatoren `new`, och anropar sedan någon av metoderna `addChild()` eller `addChildAt()` för att placera den på scenen och i visningslistan. Metoden `addChild()` placerar objektet i nästa position i visningslistan, och `addChildAt()` anger vilken position objektet ska läggas till i. Om du anger en position som redan är upptagen flyttas objektet på den positionen, samt objekten på högre positioner, uppåt ett steg. Egenskapen `numChildren` för ett `DisplayObjectContainer`-objekt innehåller antalet visningsobjekt som objektet innehåller. Du kan hämta ett objekt från visningslistan genom att anropa metoden `getChildAt()` och ange positionen. Om du vet namnet på objektet kan du också anropa metoden `getChildByName()`.

Obs! När du lägger till en komponent med `ActionScript` måste du tilldela ett namn till dess namnegenskap om du vill komma åt den via namnet i visningslistan.

I följande exempel visas namnen och positionerna för tre komponenter i visningslistan. Först drar du en `NumericStepper`, en `Button` och en `ComboBox` till scenen så att de överlappar varandra, och ger dem förekomstnamnen `aNs`, `aButton` och `aCb`. Sedan lägger du till följande kod i bildruta 1 på tidslinjen i panelen Åtgärder:

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

Följande rader visas i utdatapanelen:

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

Flytta en komponent i visningslistan

Du kan ändra positionen för ett objekt i visningslistan, och dess visningsdjup, genom att anropa metoden `addChildAt()`, och ange namnet på ett objekt samt positionen där du vill placera det som metodens parametrar. Lägg till exempel till följande kod i det föregående exemplet om du vill placera `NumericStepper` högst upp, och upprepa slingan för att visa komponentens nya position i visningslistan:

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

Följande rader visas i utdatapanelen:

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

`NumericStepper` ska också visas framför de andra komponenterna på skärmen.

Observera att `numChildren` är antalet objekt (från 1 till n) i visningslistan, där den första positionen i listan är 0. Om det finns tre objekt i listan är det tredje objektets indexposition alltså 2. Det betyder att du kan referera till den sista positionen i visningslistan, eller till det översta objektet när det gäller visningsdjup, som `numChildren - 1`.

Ta bort en komponent från visningslistan

Du kan ta bort en komponent från en visningsobjektbehållare och dess visningslista med metoderna `removeChild()` och `removeChildAt()`. Med följande exempel placeras tre `Button`-komponenter framför varandra på scenen, och en händelseavlyssnare läggs till för varje komponent. När du klickar på en `Button` tar händelsehanteraren bort den från visningslistan och scenen.

- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra en `Button` från panelen Komponenter till bibliotekspanelen.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och lägg till följande kod:

```
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);
}
```

En fullständig förklaring av visningslistan finns i Visningsprogrammering i *Programmering med ActionScript 3.0*.

Arbeta med FocusManager

När en användare trycker på tabbtangenten för att navigera i ett Flash-program, eller klickar i ett program, bestämmer `FocusManager`-klassen vilken komponent som får indatafokus. Du behöver inte lägga till en `FocusManager`-förekomst i ett program eller skriva någon kod för att aktivera fokushanteraren, såvida du inte skapar en komponent.

Om ett `RadioButton`-objekt får fokus analyserar `FocusManager` objektet och alla objekt med samma `groupName`-värde, och ställer in fokus på objektet som har en `selected`-egenskap med värdet `true`.

Varje modal fönsterkomponent innehåller en förekomst av `FocusManager`, så kontrollerna för fönstret utgör en egen tabbuppsättning. Det förhindrar att en användare av misstag navigerar till komponenter i andra fönster genom att trycka på tabbtangenten.

FocusManager använder djupnivån (eller z-ordningen) på elementen i behållaren som standardnavigationsschema eller *tabbslinga*. Vanligtvis navigerar du i tabbslingan med tabbtangenten, och fokus flyttas från den första komponenten som har fokus, till den sista, och tillbaka igen till den första. Djupnivåerna ställs primärt in efter den ordning som komponenterna dras till scenen. Du kan även använda kommandona Ändra > Ordna > Flytta längst fram/Flytta längst bak för att bestämma den slutliga z-ordningen. Mer information om olika djup finns i ”[Arbeta med visningslistan](#)” på sidan 24.

Du kan anropa metoden `setFocus()` om du vill ge fokus till en komponentförekomst i ett program. Följande exempel skapar till exempel en FocusManager-förekomst av den aktuella behållaren (`this`) och ger fokus till Button-förekomsten `aButton`.

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

Du kan bestämma vilken komponent som har fokus genom att anropa metoden `getFocus()`, och du kan bestämma vilken komponent i tabbslingan som sedan ska få fokus genom att anropa metoden `getNextFocusManagerComponent()`. I följande exempel finns en CheckBox, en RadioButton och en Button på scenen, och de har alla avlyssnare för `MouseEvent.CLICK`- och `FocusEvent.MOUSE_FOCUS_CHANGE`-händelser. När `MouseEvent.CLICK`-händelsen inträffar eftersom användaren klickar på komponenten, anropar `showFocus()`-funktionen metoden `getNextFocusManagerComponent()` för att bestämma vilken komponent i tabbslingan som ska få fokus sedan. Sedan anropar den metoden `setFocus()` för att ge fokus till den komponenten. När `FocusEvent.MOUSE_FOCUS_CHANGE`-händelsen inträffar visar `fc()`-funktionen namnet på komponenten som den här händelsen inträffade på. Händelsen utlöses när användaren klickar på en annan komponent än den som är nästa komponent i slingan.

```
// 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);
}
```

Om du vill skapa en Button som får fokus när användaren trycker på Enter (Windows) eller Retur (Macintosh) ställer du in egenskapen `FocusManager.defaultButton` för den Button-förekomst som du vill ha som standard-Button, som i följande kod:

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

FocusManager-klassen åsidosätter den fokusram som är standard i Flash Player och ritar en standardfokusram med rundade hörn.

Mer information om att skapa ett fokusschema i ett Flash-program finns klassen `FocusManager` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#). Om du vill skapa en anpassad fokushanterare måste du skapa en klass som implementerar gränssnittet `IFocusManager`. Mer information finns i `IFocusManager` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Arbeta med List-baserade komponenter

Komponenterna `List`, `DataGrid` och `TileList` ärver alla från basklassen `SelectableList`. Därför anses de här komponenterna vara List-baserade komponenter. En `ComboBox` består av en textruta och en `List`, så den är också en List-baserad komponent.

En `List` består av två rader. En `DataGrid` och en `TileList` består av rader som kan delas upp i flera kolumner. Det överlappande området mellan en rad och en kolumn är en cell. I en `List`, som är en enda kolumn med rader, är varje rad en cell. En cell har följande två viktiga aspekter:

- Datavärdena som cellerna innehåller kallas för objekt. Ett *objekt* är ett ActionScript-objekt som används för att spara informationsenheter i en `List`. En `List` kan betraktas som en array. Varje indexerat arrayområde är ett objekt. I en `List` är ett objekt ett objekt som oftast har en `label`-egenskap som visas och en `data`-egenskap som används för att lagra data. En `DataProvider` är en datamodell av objekten i en `List`. Med en `DataProvider` kan du fylla i en List-baserad komponent genom att tilldela den komponentens `DataProvider`-egenskap.
- En cell kan innehålla olika typer av data, från text till bilder, `MovieClips` eller en klass som du skapar. Därför måste en cell ritas och återges på ett sätt som är lämpligt för dess innehåll. Och därför har List-baserade komponenter en `cellRenderer` som återger cellerna. När det gäller `DataGrid` är varje kolumn ett `DataGridColumn`-objekt, som också har en `cellRenderer`-egenskap, så att varje kolumn kan återges på rätt sätt för dess innehåll.

Alla List-baserade komponenter har egenskaperna `cellRenderer` och `dataProvider` som du ställa in för att ladda och återge komponenternas celler. Information om hur du använder de här egenskaperna och arbetar med List-baserade komponenter finns i ”[Arbeta med en DataProvider](#)” på sidan 28 och ”[Arbeta med en CellRenderer](#)” på sidan 36.

Arbeta med en DataProvider

En `DataProvider` är en datakälla som du kan använda för att förse komponenterna `ComboBox`, `DataGrid`, `List` och `TileList` med data. Var och en av de här klasserna har en `dataProvider`-egenskap som du tilldelar ett `DataProvider`-objekt för att fylla i komponentens celler med data. Vanligtvis är en `DataProvider` en uppsättning med data, till exempel ett `Array`- eller `XML`-objekt.

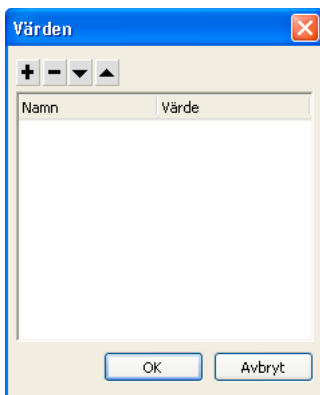
Skapa en DataProvider

För komponenterna ComboBox, List och TileList kan du skapa en DataProvider med parametern `dataProvider` i utvecklingsmiljön. Komponenten DataGrid har ingen `dataProvider`-parameter i egenskapsinspektören eftersom den kan ha flera kolumner, och dess DataProvider är därför mer komplex. Du kan också använda ActionScript för att skapa en DataProvider för de här komponenterna och för DataGrid.

Använda parametern DataProvider

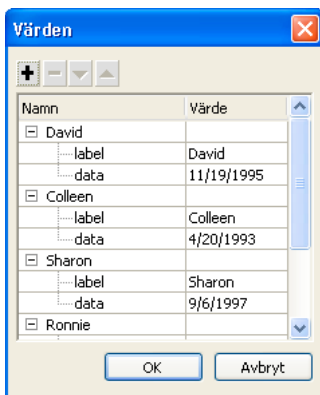
Du skapar en enkel DataProvider för komponenterna ComboBox, List och TileList genom att klicka på parametern `dataProvider` på fliken Parametrar i egenskapsinspektören eller komponentinspektören.

Om du dubbelklickar på värdecellen, som till att börja med visar en tom Array, öppnas dialogrutan Värden, där du kan ange flera etiketts- och datavärden för att skapa en DataProvider.



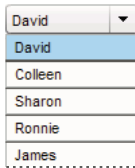
dialogrutan Värden för dataProvider

Klicka på plustecknet och lägg till ett objekt i dataProvider. Klicka på minustecknet om du vill ta bort ett objekt. Klicka på uppåtpilen för att flytta ett markerat objekt uppåt i listan, eller på nedåtpilen för att flytta ett markerat objekt nedåt i listan. I följande bild visas dialogrutan Värden som skapar en lista med barns namn och födelsedagar.



dialogrutan Värden med data

Den Array som du skapar består av par med etikett- och värdefält. Etikettfälten är `label` och `data` och värdefälten är barnens namn och födelsedagar. Etikettfältet identifierar innehållet som visas i List, i det här fallet barnens namn. Den ComboBox som blir resultatet ser ut så här:

Arbeta med komponenter

En ComboBox ifylld av DataProvider

När du har lagt till alla data klickar du på OK för att stänga dialogrutan. Den Array som finns i parametern `dataProvider` har nu fyllts i med objekten som du skapade.

Namn	Värde
<code>allowMultipleSelection</code>	<code>False</code>
<code>dataProvider</code>	<code>[{label:David,data:11/19/1995},{label:Colleen,data:4/20/1993},{label:Sharon,data:9/6/1997}]</code>
<code>enabled</code>	<code>true</code>
<code>horizontalLineScrollSize</code>	<code>1</code>
<code>horizontalPageScrollSize</code>	<code>0</code>
<code>horizontalScrollPolicy</code>	<code>auto</code>
<code>verticalLineScrollSize</code>	<code>1</code>

Parametern `dataProvider` med data

Du kan komma åt label- och datavärdena som du skapade genom att använda ActionScript för att komma åt komponentens `dataProvider`-egenskap.

Skapa en DataProvider med ActionScript

Du kan skapa en `DataProvider` genom att skapa data i ett Array- eller XML-objekt och tillhandhålla objektet som parametern `value` för `DataProvider`-konstruktorn.

Obs! I ActionScript 3.0 kan du inte tilldela ett Array- eller XML-objekt direkt till en `dataProvider`-egenskap, eftersom egenskapen är definierad som ett `DataProvider`-objekt, och bara kan ta emot ett objekt av typen `DataProvider`.

Följande exempel fyller i en `List`-komponent, som är en enda rad kolumn med rader, med namnen på flera barn och deras födelsedagar. Exemplet definierar listan i en `items`-Array och anger den som parameter när den skapar `DataProvider`-förekomsten (`ny DataProvider(objekt)`) och tilldelar den egenskapen `dataProvider` för `List`-komponenten.

```
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);
```

Denna Array består av par med etikett- och värdefält. Etikettfälten är `label` och `data` och värdefälten är barnens namn och födelsedagar. Etikettfältet identifierar innehållet som visas i `List`, i det här fallet barnens namn. Den `List` som blir resultatet ser ut så här:

Arbeta med komponenter

David
Colleen
Sharon
Ronnie
James

En List ifylld av en DataProvider

Värdet på datafältet är tillgängligt när användaren markerar ett objekt i listan genom att klicka på det, vilket orsakar en `change`-händelse. Följande exempel lägger till en `TextArea` (`aTa`) och en händelsehanterare (`changeHandler`) i föregående exempel för att visa barnets födelsedag när en användare markerar ett namn i `List`.

```
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;
};
```

När en användare nu markerar ett barns namn i `List` visas barnets födelsedag i `TextArea`, så som visas i följande bild. Det åstadkoms av funktionen `changeHandler()` när den ställer in värdet på datafältet i det markerade objektet (`event.target.selectedItem.data`) för text-egenskapen för `TextArea` (`aTa.text`). Egenskapen `event.target` är det objekt som utlöste händelsen, i det här fallet `List`.

David
Colleen
Sharon
Ronnie
James

7/6/1993

Visa datafältet från DataProvider för en List

Du kan ta med andra data än text i en `DataProvider`. I följande exempel tas filmklipp med i en `DataProvider` som förser en `TileList` med data. I exemplet byggs `DataProvider` genom att `addItem()` anropas för att lägga till varje objekt när filmklippet, en färglagd ruta, har skapats.

Arbeta med komponenter

```

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();
}

```

Du kan också använda XML-data (i stället för en array) om du vill fylla i ett DataProvider-objekt. Följande kod lagrar till exempel data i ett XML-objekt som heter `employeesXML`, och skickar sedan objektet som värdeparametern för konstruktorfunktionen `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;

```

Du kan tillhandahålla data som attribut till XML-data, som i föregående kod, eller som egenskaper för XML-data, som i följande kod:

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

DataProvider har också en uppsättning metoder och egenskaper som gör att du kan komma åt och ändra den. Du kan använda DataProvider-API för att lägga till, ta bort, byta ut, sortera och sammanfoga objekt i en DataProvider.

Ändra en DataProvider

Du kan lägga till objekt i en DataProvider med metoderna `addItem()` och `addItemAt()`. I följande exempel läggs objekt till som en användare anger i textfältet i en redigerbar ComboBox. Det förutsätts att en ComboBox har dragits in på scenen och har fått namnet `aCb`.

```
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);
}
```

Du kan också byta ut och ta bort objekt i en komponent via dess DataProvider. Följande exempel implementerar två separata List-komponenter, `listA` och `listB`, och tillhandahåller en Button med etiketten Sync. När en användare klickar på denna Button använder exemplet metoden `replaceItemAt()` för att byta ut objekten i `listB` mot objekten i `listA`. Om `listA` är längre än `listB` anropar exemplet metoden `addItem()` för att lägga till de extra objekten i `listB`. Om `listB` är längre än `listA` anropar exemplet metoden `removeItemAt()` för att ta bort de extra objekten i `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++);
    }
}
}
}
}

```

Du kan också sammanfoga med och sortera en DataProvider med metoderna `merge()`, `sort()` och `sortOn()`. Följande exempel fyller i två DataGrid-förekomster (`aDg` och `bDg`) med delvisa spelarlistor för två fotbollslag. En Button med etiketten Merge läggs till, och när användaren klickar på den sammanfogar händelsehanteraren (`mrgHandler`) spelarlistan för `bDg` med spelarlistan för `aDg` och sorterar den DataGrid som blir resultatet i namnkolumnen.

```

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");
}
```

Mer information finns i DataProvider-klassen i [Referenshandbok för ActionScript 3.0](#).

Arbeta med en CellRenderer

CellRenderer är en klass som List-baserade komponenter, till exempel List, DataGrid, TileList och ComboBox, använder för att ändra och visa anpassat cellinnehåll för sina rader. En anpassad cell kan innehålla text, en fördefinierad komponent, t.ex. en CheckBox, eller en visningsobjektclass som du skapar. Om du vill återge data med en anpassad CellRenderer kan du antingen utöka CellRenderer-klassen, eller implementera gränssnittet ICellRenderer för att skapa en egen anpassad CellRenderer-klass.

Klasserna List, DataGrid, TileList och ComboBox är underklasser till klassen SelectableList. Klassen SelectableList innehåller en cellRenderer-stil. Den här stilen definierar visningsobjektet som komponenten använder för att återge celler.

Du kan justera formateringen av stilarna som används av CellRenderer genom att anropa metoden `setRendererStyle()` för List-objektet (se ”[Formatera celler](#)” på sidan 36). Du kan också definiera en anpassad klass som du använder som CellRenderer (se ”[Definiera en anpassad CellRenderer-klass](#)” på sidan 37).

Formatera celler

Klassen CellRenderer innehåller flera stilar som du använder för att styra cellens format.

Med stilarna nedan kan du definiera skalerna som används för cellens olika lägen (inaktiverad, ned, över och upp):

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

Följande stilar anger textformateringen:

- `disabledTextFormat`

- `textFormat`
- `textPadding`

Du kan använda stilarna genom att anropa metoden `setRendererStyle()` för List-objektet, eller genom att anropa metoden `setStyle()` för objektet `CellRenderer`. Du kan hämta stilarna genom att anropa metoden `getRendererStyle()` för List-objektet eller genom att anropa metoden `getStyle()` för objektet `CellRenderer`. Du kan också komma åt ett objekt som definierar alla renderingsstilar (som namngivna egenskaper för objektet) via egenskapen `rendererStyles` för List-objektet eller metoden `getStyleDefinition()` för objektet `CellRenderer`.

Anropa metoden `clearRendererStyle()` om du vill återställa en stil till dess standardvärde.

Om du vill hämta eller ange höjden på raderna i listan använder du egenskapen `rowHeight` för List-objektet.

Definiera en anpassad `CellRenderer`-klass

Skapa en klass som utökar `CellRenderer`-klassen för att definiera en anpassad `CellRenderer`

Följande kod innehåller till exempel två klasser. Klassen `ListSample` instansierar en List-komponent och den använder den andra klassen, `CustomRenderer`, för att definiera cellrenderaren som ska användas för List-komponenten. Klassen `CustomRenderer` utökar klassen `CellRenderer`.

- 1 Välj Arkiv > Nytt.
- 2 I dialogrutan Nytt dokument som visas markerar du Flash File (ActionScript 3.0) och klickar på OK.
- 3 Välj Fönster > Komponenter så visas panelen Komponenter.
- 4 I komponentpanelen drar du en List-komponent till scenen.
- 5 Om Flash inte visar egenskapsinspektören väljer du Fönster > Egenskaper > Egenskaper.
- 6 När List-komponenten är markerad anger du egenskaperna i egenskapsinspektören:
 - Variantnamn: `myList`
 - W (bredd): 200
 - H (höjd): 300
 - X: 20
 - Y: 20
- 7 Välj bildruta 1 i lager 2 på tidslinjen och välj Fönster > Åtgärder.
- 8 Lägg till följande skript i åtgärdspanelen:

```
myList.setStyle("cellRenderer", CustomCellRenderer);
myList.addItem({label:"Burger -- $5.95"});
myList.addItem({label:"Fries -- $1.95"});
```
- 9 Välj Arkiv > Spara. Skriv ett namn på filen och klicka på OK.
- 10 Välj Arkiv > Nytt.
- 11 I dialogrutan Nytt dokument som visas markerar du ActionScript File och klickar på OK.
- 12 I skriptfönstret anger du följande kod för att definiera klassen `CustomCellRenderer`:

```

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 Välj Arkiv > Spara. Kalla filen CustomCellRenderer.as, placera den i samma katalog som FLA-filen och klicka på OK.

14 Välj Kontroll > Testa filmen.

Använda en klass som implementerar gränssnittet ICellRenderer för att definiera en anpassad CellRenderer

Du kan också definiera en CellRenderer med en klass som ärver DisplayObject-klassen och implementerar ICellRenderer-gränssnittet. Följande kod definierar till exempel två klasser. Klassen ListSample2 lägger till ett List-objekt till visningslistan och definierar dess CellRenderer för att använda CustomRenderer-klassen. Klassen CustomRenderer utökar CheckBox-klassen (som utökar DisplayObject-klassen) och implementerar gränssnittet ICellRenderer. Observera att CustomRenderer-klassen definierar get- och set-metoderna för egenskaperna data och listData som definieras i gränssnittet ICellRenderer. Andra egenskaper och metoder som är definierade i gränssnittet ICellRenderer (egenskapen selected och metoden setSize()) har redan definierats i klassen CheckBox:

1 Välj Arkiv > Nytt.

2 I dialogrutan Nytt dokument som visas markerar du Flash File (ActionScript 3.0) och klickar på OK.

3 Välj Fönster > Komponenter så visas panelen Komponenter.

4 I komponentpanelen drar du en List-komponent till scenen.

5 Om Flash inte visar egenskapsinspektören väljer du Fönster > Egenskaper > Egenskaper.

6 När List-komponenten är markerad anger du egenskaperna i egenskapsinspektören:

- Variantnamn: myList
- W (bredd): 100
- H (höjd): 300
- X: 20
- Y: 20

7 Välj bildruta 1 i lager 2 på tidslinjen och välj Fönster > Åtgärder.

8 Lägg till följande skript i åtgärdspanelen:

```

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

```

9 Välj Arkiv > Spara. Skriv ett namn på filen och klicka på OK.

10 Välj Arkiv > Nytt.

11 I dialogrutan Nytt dokument som visas markerar du ActionScript File och klickar på OK.

12 I skriptfönstret anger du följande kod för att definiera klassen CustomCellRenderer:

```
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 Välj Arkiv > Spara. Kalla den CustomCellRenderer.as, placera den i samma katalog som FLA-filen och klicka på OK.

14 Välj Kontroll > Testa filmen.

Använd en symbol för att definiera en CellRenderer

Du kan också använda en symbol i biblioteket för att definiera en CellRenderer. Symbolen måste exporteras för ActionScript, och bibliotekssymbolens klassnamn måste ha en associerad klassfil som antingen implementerar gränssnittet ICellRenderer eller utökar klassen CellRenderer (eller en av dess underklasser).

Följande exempel definierar en anpassad CellRenderer som använder en bibliotekssymbol.

- 1 Välj Arkiv > Nytt.
- 2 I dialogrutan Nytt dokument som visas markerar du Flash File (ActionScript 3.0) och klickar på OK.
- 3 Välj Fönster > Komponenter så visas panelen Komponenter.
- 4 I komponentpanelen drar du en List-komponent till scenen.
- 5 Om Flash inte visar egenskapsinspektören väljer du Fönster > Egenskaper > Egenskaper.
- 6 När List-komponenten är markerad anger du egenskaperna i egenskapsinspektören:
 - Variantnamn: myList
 - W (bredd): 100
 - H (höjd): 400
 - X: 20
 - Y: 20
- 7 Klicka på panelen Parametrar och dubbelklicka sedan på den andra kolumnen i dataProvider-raderna.

- 8 I dialogrutan Värden som visas klickar du två gånger på plustecknet för att lägga till två dataelement (med etiketterna angivna till label0 och label1). Sedan klickar du på OK.
- 9 Rita ett textfält på scenen med hjälp av textverktyget.
- 10 När textfältet är markerat anger du egenskaperna i panelen egenskapsinspektören:
 - Texttyp: Dynamisk text
 - Variantnamn: textField
 - W (bredd): 100
 - Teckensnittstorlek: 24
 - X: 0
 - Y: 0
- 11 När textfältet är markerat väljer du Ändra > Konvertera till symbol.
- 12 Gör följande inställningar i dialogrutan Konvertera till symbol och klicka sedan på OK.
 - Namn: MyCellRenderer
 - Typ: MovieClip
 - Exportera för ActionScript: Markerat
 - Exportera i första bildrutan: Markerat
 - Klass: MyCellRenderer
 - Basklass: flash.display.MovieClipOm Flash visar en ActionScript-klassvarning klickar du på OK i varningsrutan.
- 13 Ta bort förekomsten av den nya filmklippssymbolen från scenen.
- 14 Välj bildruta 1 i lager 2 på tidslinjen och välj Fönster > Åtgärder.
- 15 Lägg till följande skript i åtgärdspanelen:

```
myList.setStyle("cellRenderer", MyCellRenderer);
```
- 16 Välj Arkiv > Spara. Skriv ett namn på filen och klicka på OK.
- 17 Välj Arkiv > Nytt.
- 18 I dialogrutan Nytt dokument som visas markerar du ActionScript File och klickar på OK.
- 19 I skriptfönstret anger du följande kod för att definiera klassen MyCellRenderer:

```
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 Välj Arkiv > Spara. Kalla den MyCellRenderer.as, placera den i samma katalog som FLA-filen och klicka på OK.

21 Välj Kontroll > Testa filmen.

CellRenderers-egenskaper

Egenskapen `data` är ett objekt som innehåller alla egenskaper som är angivna för `CellRenderers`. Observera till exempel i följande klass, som definierar en anpassad `CellRenderers` som utökar klassen `CheckBox`, att `set`-funktionen för egenskapen `data` skickar värdet på `data.label` till egenskapen `label` som ärvs från klassen `CheckBox`:

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

Egenskapen `selected` anger om en cell är markerad eller inte i listan.

Använda en CellRenderers för en kolumn i ett DataGrid-objekt

Ett `DataGrid`-objekt kan ha flera kolumner och du kan ange olika cellåtergivare för varje kolumn. Varje kolumn i en `DataGrid` representeras av ett `DataGridColumn`-objekt, och klassen `DataGridColumn` innehåller en `cellRenderers`-egenskap, som du kan definiera kolumnens `CellRenderers` för.

Definiera en CellRenderers för en redigerbar cell

Klassen `DataGridCellEditor` definierar en renderare som används för redigerbara celler i ett `DataGrid`-objekt. Den blir renderare för en cell när egenskapen `editable` för `DataGrid`-objektet får värdet `true` och användaren klickar på cellen som ska redigeras. Om du vill definiera en `CellRenderers` för den redigerbara cellen anger du egenskapen `itemEditor` för varje element i `columns`-arrayen för `DataGrid`-objektet.

Använda en bild, SWF-fil eller ett filmklipp som en CellRenderers

Klassen `ImageCell`, en underklass till `CellRenderers`, definierar ett objekt som används för att återge celler där det huvudsakliga cellinnehållet är en bild, SWF-fil eller ett filmklipp. Klassen `ImageCell` innehåller följande stilar för att definiera cellens utseende:

- `imagePadding` - Den utfyllning som skiljer cellens kant från bildens kant, i pixlar
- `selectedSkin` - Det skal som används för att ange att något är markerat
- `textOverlayAlpha` - Opaciteten för överlägget bakom celletiketten
- `textPadding` - Den utfyllning som skiljer cellens kant från textens kant, i pixlar

Klassen `ImageCell` är standard-`CellRenderers` för `TileList`-klassen.

Göra komponenter åtkomliga

Du kan göra visuellt innehåll i Flash-programmen åtkomligt för användare med nedsatt syn via en skärmläsare. Med skärmläsaren får användarna en ljudbeskrivning av skärminnehållet. Information om hur du gör Flash-programmet åtkomligt för en skärmläsare finns i Kapitel 18, ”Skapa åtkomligt innehåll” i *Använda Flash*.

Om du vill göra en ActionScript 3.0-komponent åtkomlig för en skärmläsare måste du även importera dess hjälpmedelsklass och anropa den klassens `enableAccessibility()`-metod. Du kan göra följande ActionScript 3.0-komponenter åtkomliga för en skärmläsare:

Komponent	Hjälpmmedelsklass
Button	ButtonAccImpl
CheckBox	CheckBoxAccImpl
ComboBox	ComboBoxAccImpl
List	ListAccImpl
RadioButton	RadioButtonAccImpl
TileList	TileListAccImpl

Komponentens hjälpmedelsklasser finns i paketet `fl.accessibility`. Om du gör en `CheckBox` åtkomlig för till exempel en skärmläsare lägger du till följande programsatser i ditt program:

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

Du aktiverar bara en komponents åtkomlighet en gång, oavsett hur många förekomster du skapar.

Obs! Om du aktiverar åtkomligheten ökar filstorleken obetydligt genom att de klasser som krävs tas med vid kompileringen.

De flesta komponenter går att navigera via tangentbordet. Mer information om att aktivera åtkomliga komponenter och navigera med tangentbordet finns i avsnitten om användning i ”[Använda UI-komponenterna](#)” på sidan 44 och hjälpmedelsklasserna i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

Kapitel 4: Använda UI-komponenterna

I det här kapitlet beskrivs hur du använder användargränssnittskomponenter (UI) i ActionScript 3.0, som ingår i Flash.

Använda komponenten Button

Button-komponenten är en rektangulär knapp vars storlek kan ändras och som en användare kan klicka på med musen eller mellanslagstangenten för att initiera en åtgärd i programmet. Du kan lägga till en anpassad ikon på en Button. Du kan också ändra beteendet för en Button från push-knapp till en toggle-knapp. En alternativknapps kontroll förblir nedtryckt när användaren klickar på den, och återgår till uppåt-läget när användaren klickar på den igen.

En Button är en grundläggande del av många formulär och webbprogram. Du kan använda knappar när du vill att en användare ska initiera en händelse. De flesta formulär har till exempel en Skicka-knapp. Du kan också lägga till Föregående- och Nästa-knappar i en presentation.

Använda komponenten Button

Du kan aktivera eller inaktivera en knapp i ett program. I inaktiverat läge tar knappen inte emot inmatningar från musen eller tangentbordet. En aktiverad knapp får fokus om du klickar på den eller går till den med tabbtangenten. När en Button-förekomst har fokus kan du använda följande tangenter för att styra den:

Nyckel	Beskrivning
Skift+Tabb	Flyttar fokus till föregående objekt.
Mellanslagstangent	Trycker på eller släpper knappen och utlöser <code>click</code> -händelsen.
Tabb	Flyttar fokus till nästa objekt.
Enter/Retur	Flyttar fokus till nästa objekt om en knapp har angetts som standard-Button för FocusManager.

Mer information om hur du styr fokus finns i avsnittet om gränssnittet IFocusManager och klassen FocusManager i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#) och ”Arbeta med FocusManager” på sidan 26.

En direktförhandsvisning av varje Button-förekomst reflekterar ändringar som gjorts för parametrar i egenskapsinspektören eller komponentinspektören vid redigeringen.

Obs! Om en ikon är större än knappen visas den utanför knappens gränser.

Om du vill ange en knapp som den push-knapp som är standard i ett program (knappen som tar emot `click`-händelsen när en användare trycker på Enter), anger du `FocusManager.defaultButton`. Följande kod anger till exempel att standardknappen ska vara en Button-förekomst som kallas `submitButton`.

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

När du lägger till Button-komponenten i ett program kan du göra den åtkomlig för en skärmläsare genom att lägga till följande rader med ActionScript-kod:

```
import fl.accessibility.ButtonAccImpl;

ButtonAccImpl.enableAccessibility();
```


Du aktiverar bara en komponents åtkomlighet en gång, oavsett hur många förekomster du skapar.

Button-komponentparametrar

Du kan ange följande utvecklingsparametrar i egenskapsinspektören (Fönster > Egenskaper > Egenskaper) eller i komponentinspektören (Fönster > Komponentinspektören) för varje Button-förekomst: `emphasized`, `label`, `labelPlacement`, `selected` och `toggle`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. När du tilldelar ett värde till parametrarna anger du egenskapens startläge i programmet. Om du anger egenskapen i ActionScript åsidosätter det värdet som du anger i parametern. Information om möjliga värden på de här parametrarna finns i Button-klassen i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med komponenten Button

I proceduren nedan beskrivs hur du lägger till en Button-komponent i ett program när du redigerar. I det här exemplet ändrar Button läget på en ColorPicker-komponent när du klickar på den.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en Button-komponent från panelen Komponenter till scenen och ange följande värden för den i egenskapsinspektören:
 - Ange förekomstnamnet **aButton**.
 - Ange **Visa** som etikettparameter.
- 3 Lägg till en ColorPicker på scenen och ge den förekomstnamnet **aCp**.
- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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;
    }
}
```

Den andra raden med kod registrerar funktionen `clickHandler()` som händelsehanterare för händelsen `MouseEvent.CLICK`. Händelsen inträffar när en användare klickar på `Button`, vilket gör att funktionen `clickHandler()` utför någon av följande åtgärder, beroende på vilket värdet `Button` har:

- Visa gör `ColorPicker` synlig och ändrar `Button`-etiketten till Inaktivera.
- Inaktivera inaktiverar `ColorPicker` och ändrar `Button`-etiketten till Aktivera.
- Aktivera aktiverar `ColorPicker` och ändrar `Button`-etiketten till Dölj.
- Dölj gör `ColorPicker` osynlig och ändrar `Button`-etiketten till Visa.

5 Välj Kontroll > Testa filmen för att köra programmet.

Skapa ett program med komponenten Button

Följande procedur skapar en alternativknapps kontroll med `ActionScript` och visar händelsetypen i utdatapanelen när du klickar på `Button`. I exemplet skapas `Button`-förekomsten genom att klassens konstruktor anropas och läggs till på scenen genom att anropa metoden `addChild()`.

1 Skapa ett nytt Flash-dokument (`ActionScript 3.0`).

2 Dra `Button`-komponenten från panelen Komponenter till det aktuella dokumentets bibliotekspanel.

Komponenten läggs till i biblioteket, men den syns inte i programmet.

3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande kod för att skapa en `Button`-förekomst:

```
import fl.controls.Button;

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

Metoden `move()` placerar knappen vid 50 (x-koordinat), 50 (y-koordinat) på scenen.

4 Nu lägger du till följande `ActionScript` för att skapa en händelseavlyssnare och en händelsehanterarfunktion:

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

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

5 Välj Kontroll > Testa filmen.

När du klickar på knappen visar Flash meddelandet "Händelsetyp: klick" på utdatapanelen.

Använda komponenten CheckBox

En `CheckBox` är en fyrkantig ruta som kan markeras eller avmarkeras. När den är markerad visas en bock i rutan. Du kan lägga till en textetikett i en `CheckBox` och placera den till vänster om, höger om, ovanför eller nedanför `CheckBox`.

Du kan använda flera `CheckBox` om du vill samla in en uppsättning med `true`- och `false`-värden som inte är ömsesidigt uteslutande. Ett program som samlar in information om vilken sorts bil du vill köpa kan använda `CheckBoxes` för att låta dig välja funktioner.

Användarinteraktion med CheckBox

Du kan aktivera eller inaktivera en CheckBox i ett program. Om en CheckBox är aktiverad och en användare klickar på den eller på dess etikett, får CheckBox fokus för inmatning och dess nedtryckta utseende visas. Om en användare flyttar pekaren utanför markeringsramen för en CheckBox eller dess etikett samtidigt som han håller ned musknappen återgår komponentens utseende till det ursprungliga läget, och den behåller indatafokus. Läget för en CheckBox ändras inte förrän musknappen släpps ovanför komponenten. Dessutom har CheckBox två inaktiverade lägen, markerad och avmarkerad, som använder `selectedDisabledSkin` och `disabledSkin`, och som inte tillåter interaktion via mus eller tangentbord.

Om en CheckBox är inaktiverad visas dess inaktiverade utseende, oavsett användarinteraktion. I inaktiverat läge tar en CheckBox inte emot inmatningar från musen eller tangentbordet.

En CheckBox-förekomst får fokus om en användare klickar på den eller går till den med tabbtangenten. När en CheckBox-förekomst har fokus kan du använda följande tangenter för att styra den:

Nyckel	Beskrivning
Skift+Tabb	Flyttar fokus till föregående element.
Mellanslagstangent	Markerar eller avmarkerar komponenten och utlöser <code>change</code> -händelsen.
Tabb	Flyttar fokus till nästa element.

Mer information om hur fokuskontroll finns i ”Arbeta med FocusManager” på sidan 26 och i FocusManager -klassen i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

En direktförhandsvisning av varje CheckBox-förekomst reflekterar ändringar som gjorts för parametrar i egenskapsinspektören eller komponentinspektören vid redigeringen.

När du lägger till CheckBox-komponenten i ett program kan du göra den åtkomlig för en skärmläsare genom att lägga till följande rader med ActionScript-kod:

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

Du aktiverar bara en komponents åtkomlighet en gång, oavsett hur många förekomster du har av komponenten.

CheckBox-komponentparametrar

Du kan ställa in följande utvecklingsparametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av CheckBox-komponenten: `label`, `labelPlacement` och `selected`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i CheckBox-klassen i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

Skapa ett program med Checkbox

Följande procedur beskriver, med hjälp av ett utdrag ur ett låneansökningsformulär, hur du lägger till en CheckBox-komponent i ett program under utvecklingen. I formuläret får den som ansöker frågan om han eller hon är en husägare, och där finns en CheckBox där användaren kan svara ”ja”. Om användaren svarar ”ja” visas två alternativknappar där användaren kan ange husets relativa värde.

Skapa ett program med Checkbox-komponenten

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).

- 2 Dra en CheckBox-komponent från panelen Komponenter till scenen.
- 3 Gör följande i egenskapsinspektören:
 - Ange **homeCh** som förekomstnamn.
 - Ange **140** som värde för bredden (W).
 - Ange ”**Äger ditt hus?**” för etikettparametern.
- 4 Dra två RadioButton-komponenter från panelen Komponenter till scenen och placera dem nedanför och till höger om CheckBox. Ange följande värden för dem i egenskapsinspektören:
 - Ange **underRb** och **overRb** som förekomstnamn.
 - Ange **120** för parametern W (bredd) för båda RadioButtons.
 - Ange **Under 500 000 kr?** för etikettparametern för **underRb**.
 - Ange **Över 500 000 kr?** för etikettparametern för **overRb**.
 - Ange **valueGrp** för parametern groupName för båda RadioButtons.
- 5 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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;
}
```

Den här koden skapar en händelsehanterare för en CLICK-händelse som aktiverar RadioButtons **underRb** och **overRb** om CheckBox **homeCh** markeras, och inaktiverar dem om **homeCh** inte markeras. Mer information finns i avsnittet om klassen `MouseEvent` i [Referenshandbok för ActionScript 3.0](#) i *Adobe Flash Professional*.

- 6 Välj Kontroll > Testa filmen.

Följande exempel kopierar det föregående programmet, men skapar CheckBox och RadioButtons med ActionScript.

Skapa en CheckBox med ActionScript

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra CheckBox-komponenten och RadioButton-komponenten från panelen Komponenter till det aktuella dokumentets bibliotekspanel. Om bibliotekspanelen inte är öppen trycker du på Ctrl+L eller väljer Fönster > Bibliotek för att öppna den.
Då blir komponenterna tillgängliga för ditt program, men de placeras inte på scenen.
- 3 Öppna panelen Åtgärder, markera bildruta 1 i huvudtidslinjen och ange följande kod för att skapa och placera komponentförekomsterna:

```
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?";
```

Koden använder konstruktorerna `CheckBox()` och `RadioButton()` för att skapa komponenterna och metoden `addChild()` för att placera dem på scenen. Den använder metoden `move()` för att placera komponenterna på scenen.

- 4 Nu lägger du till följande ActionScript för att skapa en händelseavlyssnare och en händelsehanterarfunktion:

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

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

Den här koden skapar en händelsehanterare för en `CLICK`-händelse som aktiverar alternativknapparna `underRb` och `overRb` om `CheckBox` `homeCh` markeras, och inaktiverar dem om `homeCh` inte markeras. Mer information finns i avsnittet om klassen `MouseEvent` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

- 5 Välj Kontroll > Testa filmen.

Använda komponenten ColorPicker

Med komponenten `ColorPicker`-komponenten kan en användare välja en färg i en lista med färgrutor. I standardläget för `ColorPicker` visas en färg på en fyrkantig knapp. När användaren klickar på knappen visas listan med färger i en färgrutepanel, tillsammans med ett textfält som visar det hexadecimala värdet på den aktuella färgen.

Du kan ställa in färgerna som visas i `ColorPicker` genom att ange de färgvärden som du vill visa i dess `colors`-egenskap.

Använda komponenten ColorPicker

Med en ColorPicker kan användaren välja en färg och använda den på ett annat objekt i programmet. Om du till exempel vill låta användaren anpassa vissa element i programmet, som bakgrundsfärg eller textens färg, kan du ta med en ColorPicker och använda den färg som användaren väljer.

Användaren väljer en färg genom att klicka på dess färgruta i panelen, eller genom att ange dess hexadecimala värde i textfältet. När användaren har valt en färg kan du använda ColorPickers egenskap `selectedColor` för att använda färgen på text eller andra objekt i programmet.

En ColorPicker-förekomst får fokus om användaren rör pekaren över den eller går till den med tabbtangenten. När en ColorPickers färgrutepanel är öppen kan du använda följande tangenter för att styra den:

Nyckel	Beskrivning
Hem	Markerar den första färgen i färgrutepanelen.
Upppilen	Flyttar markeringen en rad uppåt i färgrutepanelen.
Nedpilen	Flyttar markeringen en rad nedåt i färgrutepanelen.
Högerpil	Flyttar markeringen i färgrutepanelen en färg åt höger.
Vänsterpil	Flyttar markeringen i färgrutepanelen en färg åt vänster.
End	Markerar den sista färgen i färgrutepanelen.

ColorPicker-komponentparametrar

Du kan ställa in följande utvecklingsparametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av ColorPicker-komponenten: `selectedColor` och `showTextField`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i ColorPicker-klassen i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med komponenten ColorPicker

I följande exempel läggs en ColorPicker-komponent till i ett program under utvecklingen. Varje gång du ändrar färgen i ColorPicker i det här exemplet anropar funktionen `changeHandler()` funktionen `drawBox()` för att rita en ny ruta med den färg som du valde i ColorPicker.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en ColorPicker från panelen Komponenter till mitten av scenen och ge den förekomstnamnet **aCp**.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen.
- 5 Klicka på ColorPicker och välj en färg att färglägga rutan med.

Skapa en ColorPicker med ActionScript

I det här exemplet används konstruktorn `ColorPicker()` och `addChild()` för att skapa en `ColorPicker` på scenen. Egenskapen `colors` får färgvärdena för röd (0xFF0000), grön (0x00FF00) och blå (0x0000FF) för att ange färgerna som `ColorPicker` ska visa. En `TextArea` skapas och varje gång du väljer en annan färg i `ColorPicker` ändrar exemplet färgen på texten i `TextArea` så att den matchar.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra `ColorPicker`-komponenten från panelen Komponenter till bibliotekspanelen.
- 3 Dra `TextArea`-komponenten från panelen Komponenter till bibliotekspanelen.
- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen.

Använda komponenten ComboBox

Med ComboBox-komponenten kan användaren göra ett enda val i en listruta. En ComboBox kan vara statisk eller redigerbar. Med en redigerbar ComboBox kan användaren ange text direkt i textfältet högst upp i listan. Om listrutan når ända ned till dokumentets nederkant öppnas den uppåt i stället för nedåt. En ComboBox består av tre underkomponenter: komponenterna BaseButton, TextInput och List.

I en redigerbar ComboBox består träffområdet enbart av – inte av textrutan. För en statisk ComboBox består träffområdet av både knappen och textrutan. Träffområdet svarar genom att öppna eller stänga listrutan.

När användaren markerar något i listan, med musen eller via tangentbordet, kopieras markeringens etikett till textfältet högst upp i ComboBoxen.

Använda ComboBox-komponenten

Du kan använda ComboBox-komponenter i formulär och program där användaren behöver göra ett enda val i en lista. Du kan till exempel skapa en listruta för länder i ett formulär för kundadresser. En redigerbar ComboBox används för mer komplexa scenarier. I ett program som ger köranvisningar kan du till exempel använda en redigerbar ComboBox där användaren kan ange den ursprungliga adressen och destinationsadressen. Listrutan innehåller de adresser som användaren har angett tidigare.

Om ComboBoxen är redigerbar, det vill säga att egenskapen `editable` är `true`, tar följande tangenter bort fokus från textinmatningsrutan och låter det föregående värdet stå kvar. Undantaget är Enter-tangenten, som använder det nya värdet först om användaren anger text.

Nyckel	Beskrivning
Skift + tabbtangent	Flyttar fokus till föregående objekt. Om ett nytt objekt markeras, skickas en ny <code>change</code> -händelse.
Tabb	Flyttar fokus till nästa objekt. Om ett nytt objekt markeras, skickas en ny <code>change</code> -händelse.
Nedpilen	Flyttar markeringen ett objekt nedåt.
End	Flyttar markeringen längst ned i listan.
Escape	Stänger listrutan och returnerar fokus till ComboBox.
Enter	Stänger listrutan och returnerar fokus till ComboBox. När ComboBox är redigerbar och användaren anger text, gör Retur att värdet får den angivna texten.
Hem	Flyttar markeringen högst upp i listan.
Page Up	Flyttar markeringen en sida uppåt.
Page Down	Flyttar markeringen en sida nedåt.

När du lägger till ComboBox-komponenten i ett program kan du göra den åtkomlig för en skärmläsare genom att lägga till följande rader med ActionScript-kod:

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

Du aktiverar bara en komponents åtkomlighet en gång, oavsett hur många förekomster du har av komponenten.

ComboBox-komponentparametrar

Du kan ställa in följande utvecklingsparametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av ComboBox-komponenten: `dataProvider`, `editable`, `prompt` och `rowCount`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i ComboBox-klassen i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#). Information om hur du använder parametern `dataProvider` finns i ”[Använda parametern DataProvider](#)” på sidan 29.

Skapa ett program med komponenten ComboBox

I proceduren nedan beskrivs hur du lägger till en ComboBox-komponent i ett program när du redigerar. ComboBox är redigerbar, och om du skriver **Lägg till** i textfältet lägger exemplet till ett objekt i listrutan.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en ComboBox till scenen och ge den förekomstnamnet **aCb**. På fliken Parametrar ger du parametern `editable` värdet `true`.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande kod:

```
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 Välj Kontroll > Testa filmen.

Skapa en ComboBox med ActionScript

Följande exempel skapar en ComboBox med ActionScript och fyller i den med en lista med universitet i San Francisco, Kalifornien. Egenskapen `width` för ComboBoxen ställs in så att den passar prompttextens bredd och egenskapen `dropdownWidth` ställs in så att den är lite bredare, för att få plats med det längsta universitetsnamnet.

Listan med universitet skapas i en Array-förekomst, där egenskapen `label` används för att lagra universitetsnamnen och egenskapen `data` för att lagra URL:er till universitetens webbplatser. Arrayen tilldelas ComboBox genom att dess `dataProvider`-egenskap ställs in.

När en användare väljer ett universitet från listan aktiveras en `Event.CHANGE`-händelse och ett anrop till funktionen `changeHandler()`, som läser in egenskapen `data` i en URL-begäran för att komma åt universitetets webbplats.

Observera att den sista raden ställer in `ComboBox`-förekomstens `selectedIndex`-egenskap till `-1` för att prompten ska visas igen när listan stängs. Annars skulle prompten bytas ut mot namnet på universitetet som markerades.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra ComboBox-komponenten från panelen Komponenter till bibliotekspanelen.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmerna.

Du kan implementera och köra det här exemplet i Flash-utvecklingsmiljön, men du får varningsmeddelanden om du försöker komma åt universitetens webbplatser genom att klicka på objekt i ComboBoxen. Om du vill komma åt den fungerande ComboBoxen på Internet går du till följande webbplats:

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

Använda komponenten DataGrid

Med komponenten DataGrid kan du visa data i ett rutnät med rader och kolumner och hämta data från en array eller extern XML-fil, som du kan tolka till en array för DataProvider. DataGrid-komponenten innehåller lodrät och vågrät rullningen, händelsestöd (inklusive stöd för redigerbara celler) och sorteringsfunktioner.

Du kan ändra storlek på och anpassa egenskaper som teckensnitt, färg och kantlinjer för kolumnerna i ett rutnät. Du kan använda ett anpassat filmklipp som cellrenderare för alla kolumner i ett rutnät. (En cellrenderare visar innehållet i en cell.) Du kan stänga av rullningslister och använda DataGrid-metoderna för att skapa en stilvisning i sidovy. Mer information om anpassning finns i klassen DataGridColumn i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

Fler hjälpsnitt

[Skapa, fylla och storleksändra DataGrid-komponenten](#)

[Anpassa och sortera DataGrid-komponenten](#)

[Filtrera och formatera data i DataGrid-komponenten](#)

Använda komponenten DataGrid

Använd musen eller tangentbordet när du vill interagera med en DataGrid-komponent.

Om egenskapen `sortableColumns` och kolumnens egenskap `sortable` båda är `true`, sorteras data baserat på kolumnens värden när du klickar på en kolumnrubrik. Du kan inaktivera sorteringen av en enskild kolumn genom att ställa in dess `sortable`-egenskap till `false`.

Om egenskapen `resizableColumns` är `true` kan du ändra storlek på kolumner genom att dra kolumnavgränsarna i rubrikraden.

Om du klickar i en redigerbar cell får cellen fokus. Om du klickar i en cell som inte är redigerbar har det ingen effekt på fokus. En enskild cell är redigerbar när både egenskapen `DataGrid.editable` och egenskapen `DataGridColumn.editable` för cellen är `true`.

Mer information finns i klasserna `DataGrid` och `DataGridColumn` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

När en DataGrid-förekomst får fokus, antingen via klickning eller tabbning, använder du följande tangenter för att styra den:

Nyckel	Beskrivning
Nedpilen	När en cell redigeras flyttas insättningspunkten till slutet av cellens text. Om cellen inte är redigerbar hanterar nedpiltangenten markeringen som List-komponenten gör.
Upppilen	När en cell redigeras flyttas insättningspunkten till början av cellens text. Om cellen inte är redigerbar hanterar uppiltangenten markeringen som List-komponenten gör.
Skift+uppil/nedpil	Om DataGrid inte är redigerbart och <code>allowMultipleSelection</code> är <code>true</code> markeras på varandra följande rader. Om du går i motsatt riktning med motsatt pil avmarkeras de markerade raderna till du passerar den startraden. Sedan markeras raderna i den riktningen.
Skift+klick	Om <code>allowMultipleSelection</code> är <code>true</code> markeras alla rader mellan den markerade raden och den aktuella inmatningspunkten (markad cell).
Ctrl+klick	Om <code>allowMultipleSelection</code> är <code>true</code> markeras ytterligare rader, som inte behöver följa direkt efter varandra.
Högerpil	När en cell redigeras flytts insättningspunkten ett tecken åt höger. Om en cell inte är redigerbar händer ingenting.
Vänsterpil	När en cell redigeras flytts insättningspunkten ett tecken åt vänster. Om en cell inte är redigerbar händer ingenting.
Hem	Markerar den första raden i DataGrid.
End	Markerar den sista raden i DataGrid.
PageUp	Markerar den första raden på en sida i DataGrid. En sida består av antalet rader som DataGrid kan visa utan att du behöver rulla.

Nyckel	Beskrivning
PageDown	Markerar den sista raden på en sida i DataGrid. En sida består av antalet rader som DataGrid kan visa utan att du behöver rulla.
Retur/Enter/Skift+Enter	När en cell är redigerbar utförs ändringen och insättningspunkten flyttas till cellen i samma kolumn på nästa rad (uppåt eller nedåt, beroende på vilket läge Skift har).
Skift+tabbtangent/tabbtangent	Om DataGrid är redigerbart flyttas fokus till föregående/nästa objekt tills du når kolumnens slut, och sedan till föregående/näst rad till du når den första eller sista cellen. Om den första cellen är markerad flyttar Skift+tabbtangent fokus till den föregående kontrollen. Den den sista cellen är markerad flyttar tabbtangenten fokus till nästa kontroll. Om DataGrid inte är redigerbart flyttas fokus till föregående/nästa kontroll.

Du kan använda DataGrid-komponenten som bas för olika typer av datadrivna program. Det är enkelt att visa en formaterad tabellvy med data, men du kan också använda cellrenderaren för att bygga mer sofistikerade och redigerbara delar av användargränssnitt. Nedan anges några praktiska användningsområden för DataGrid-komponenten:

- En webbmail-klient
- Sökresultatsidor
- Kalkylbladsprogram, till exempel låneräknare och skatteformulärsprogram

När du utformar ett program med DataGrid-komponenten kan det vara till hjälp att förstå hur List-komponenten är utformad, eftersom DataGrid-klassen utökar klassen SelectableList. Mer information om klassen SelectableList och komponenten List finns i klasserna SelectableList och List i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

När du lägger till en DataGrid-komponent i ett program kan du göra den åtkomlig för en skärmläsare genom att lägga till följande rader med ActionScript-kod:

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

Du aktiverar bara en komponents åtkomlighet en gång, oavsett hur många förekomster komponenten har. Mer information finns i kapitel 18, "Skapa åtkomligt innehåll" i *Använda Flash*.

DataGrid-komponentparametrar

Du kan ställa in följande utvecklingsparametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av DataGrid-komponenten: `allowMultipleSelection`, `editable`, `headerHeight`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `resizableColumns`, `rowHeight`, `showHeaders`, `verticalLineScrollSize`, `verticalPageScrollSize` och `verticalScrollPolicy`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i avsnittet om klassen DataGrid i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med komponenten DataGrid

Om du vill skapa ett program med DataGrid-komponenten måste du först bestämma var dina data kommer från. Vanligtvis kommer data från en Array som du kan dra in i rutnätet genom att ange egenskapen `dataProvider`. Du kan också använda metoderna i klasserna DataGrid och DataGridColumn för att lägga till data i rutnätet.

Använda en lokal DataProvider med en DataGrid-komponent:

I det här exemplet skapas en DataGrid som visar ett fotbollslags spelarlista (roster). Spelarlistan (roster) definieras i en Array (aRoster) som tilldelas DataGridens dataProvider-egenskap.

- 1 I Flash väljer du först Arkiv > Ny och sedan Flash File (ActionScript 3.0).
- 2 Dra en DataGrid-komponent från panelen Komponenter till scenen.
- 3 Ange förekomstnamnet **aDg** i egenskapsinspektören.
- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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);
};
```

Funktionen `bldRosterGrid()` anger storleken på DataGrid samt ordningen på kolumnerna och deras storlek.

- 5 Välj Kontroll > Testa filmen.

Ange kolumner och lägg till sortering för en DataGrid-komponent i ett program

Observera att du kan klicka på vilken kolumnrubrik som helst för att sortera DataGrid-innehållet i fallande ordning efter kolumnens värden.

I följande exempel används metoden `addColumn()` för att lägga till `DataGridColumn`-förekomster i en `DataGrid`. Kolumnerna representerar spelarnamn och deras resultat. I exemplet anges också egenskapen `sortOptions` för att specificera sorteringsalternativen för varje kolumn: `Array.CASEINSENSITIVE` för namnkolumnen och `Array.NUMERIC` för resultatcolumnen. Storleken på `DataGrid` ändras genom att längden ställs in till antalet rader och bredden till 200.

- 1 I Flash väljer du först Arkiv > Ny och sedan Flash File (ActionScript 3.0).
- 2 Dra en `DataGrid`-komponent från panelen Komponenter till scenen.
- 3 Ange förekomstnamnet **aDg** i egenskapsinspektören.
- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen.

Skapa en förekomst av en `DataGrid`-komponent med ActionScript

I det här exemplet skapas en `DataGrid` med ActionScript som fylls i med en `Array` med spelarnamn och resultat.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra `DataGrid`-komponenten från panelen Komponenter till det aktuella dokumentets bibliotekspanel. Komponenter läggs till i biblioteket, men den syns inte i programmet.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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);
```

Den här koden skapar `DataGrid`-förekomsten och ändrar sedan storlek på och placerar rutnätet.

- 4 Skapa en `array`, lägg till data i `array`en och identifiera `array`en som `DataProvider` för `DataGrid`:

Använda UI-komponenterna

```
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 Välj Kontroll > Testa filmen.

Läsa in en DataGrid med en XML-fil

I följande exempel används klassen DataGridColumn för att skapa DataGrid-kolumner. DataGrid fylls i genom att ett XML-objekt skickas som value-parametern för DataProvider()-konstruktorn.

- 1 Använd en textredigerare för att skapa en XML-fil med följande data och spara den som team.xml i samma mapp som du tänker spara FLA-fil.

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

- 2 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 3 I panelen Komponenter dubbelklickar du på komponenten DataGrid för att lägga till den på scenen.
- 4 Ange förekomstnamnet **aDg** i egenskapsinspektören.
- 5 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:


```
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 Välj Kontroll > Testa filmerna.

Använda komponenten Label

Med komponenten Label visas en rad med text, vanligtvis för att identifiera ett annat element eller en annan aktivitet på en webbsida. Du kan ange att en etikett ska formateras med HTML för att dra nytta av dess textformateringstaggar. Du kan även styra justeringen av och storleken på en etikett. Label-komponenter har inga kantlinjer, kan inte sättas i fokus och sänder inga händelser.

En direktförhandsvisning av varje Label-förekomst reflekterar ändringar som gjorts på parametrar i egenskapsinspektören eller komponentinspektören vid redigeringen. Etiketten har ingen kant, så det enda sättet att se den i direktförhandsvisning är att ange etikettens textparameter.

Använda komponenten Label

Använd Label-komponenten för att skapa en textetikett för en annan komponent i ett formulär, till exempel etiketten "Namn:" till vänster om ett TextInput-fält där ett användarnamn ska anges. Det är praktiskt att använda en Label-komponent i stället för ett vanligt textfält, eftersom du kan använda stilar för att få ett konsekvent grafiskt utseende och känsla.

Om du vill rotera en Label-komponent måste du aktivera teckensnitten, annars visas de inte när du testar filmerna.

Label-komponentparametrar

Du kan ställa in följande utvecklingsparametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av Label-komponenten: `autoSize`, `condenseWhite`, `selectable`, `text` och `wordWrap`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i avsnittet om klassen Label i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med komponenten Label

I proceduren nedan beskrivs hur du lägger till en Label-komponent i ett program när du redigerar. I det här exemplet visar etiketten texten Förfalldatum.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en Label-komponent från panelen Komponenter till scenen och ge den följande värden i egenskapsinspektören:
 - Ange **aLabel** som förekomstnamn.
 - Ange **80** som W-värde.
 - Ange **100** som X-värde.
 - Ange **100** som Y-värde.
 - Ange **Förfalldatum** som `text`-parameter.
- 3 Dra en TextArea-komponent till scenen och ge den följande värden i egenskapsinspektören:
 - Ange **aTa** som förekomstnamn.
 - Ange **22** som H-värde.
 - Ange **200** som X-värde.
 - Ange **100** som Y-värde.
- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen.

Skapa en förekomst av en Label-komponent med ActionScript

Med följande exempel skapas en Label-parameter med ActionScript. En Label används för att identifiera funktionen för en ColorPicker-komponent och egenskapen `htmlText` används för att formatera Label-texten.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra Label-komponenten från panelen Komponenter till det aktuella dokumentets bibliotekspanel.
- 3 Dra ColorPicker-komponenten från panelen Komponenter till det aktuella dokumentets bibliotekspanel.
- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen.

Använda komponenten List

Komponenten List är en rullningsbar listruta där ett eller flera alternativ kan vara markerade. En lista kan också visa grafik, inklusive andra komponenter. Du lägger till objekten som visas i listan genom att använda dialogrutan Värden, som visas när du klickar på etikett- eller parameterfälten. Du kan också använda metoderna `List.addItem()` och `List.addItemAt()` för att lägga till objekt i listan.

List-komponenten använder ett nollbaserat index, där objektet med index 0 är det översta objektet som visas. När du lägger till, tar bort eller byter ut listobjekt med metoderna och egenskaperna för List-klassen kan du behöva ange listobjektets index.

Användarinteraktion med komponenten List

Du kan ställa in en lista så att användarna kan markera ett eller flera alternativ. Till exempel kan en användare som besöker en e-handelsplats behöva markera vilket objekt han vill köpa. Det finns 30 objekt i en lista som användaren kan bläddra igenom, och användaren väljer ett objekt genom att klicka på det.

Du kan också utforma en List som använder anpassade filmklipp som rader, så att du kan visa mer information för användaren. I ett e-postprogram kan till exempel varje brevlåda vara en List-komponent och varje rad kan ha ikoner som anger prioritet och status.

Listan får fokus när du klickar på den eller tabbar till den, och sedan kan du använda följande tangenter för att styra den:

Nyckel	Beskrivning
Alfanumeriska tangenter	Hoppa till nästa objekt som har <code>Key.getAscii()</code> som det första tecknet i etiketten.
Kontroll	Växlingstangent som gör att du kan göra flera markeringar eller avmarkeringar som inte är i följd.
Nedpilen	Markeringen flyttar nedåt ett objekt.
Hem	Markeringen flyttar högst upp i listan.
Page Down	Markeringen flyttar nedåt en sida.
Page Up	Markeringen flyttar uppåt en sida.
Skift	Gör sammanhängande markering möjlig.
Upppilen	Markeringen flyttar uppåt ett objekt.

Obs! Observera att rullningsstorlekar anges i pixlar och inte i rader.

Obs! Sidstorleken som används av tangenterna `Page Up` och `Page Down` är ett mindre än antalet objekt som får plats i visningen. Om du till exempel bläddrar nedåt genom en listruta med tio rader visas listobjekten 0-9, 9-18, 18-27 och så vidare, med ett överlappande objekt per sida.

Mer information om att kontrollera fokus finns i avsnittet om gränssnittet `IFocusManager` och i klassen `FocusManager` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#) och ”Arbeta med `FocusManager`” på sidan 26.

En direktförhandsvisning av varje `List`-förekomst på scenen reflekterar ändringar som gjorts på parametrar i egenskapsinspektören eller komponentinspektören vid redigeringen.

När du lägger till `List`-komponenten i ett program kan du göra den åtkomlig för en skärmläsare genom att lägga till följande rader med `ActionScript`-kod:

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

Du aktiverar bara en komponents åtkomlighet en gång, oavsett hur många förekomster komponenten har. Mer information finns i kapitel 18, ”Skapa åtkomligt innehåll” i *Använda Flash*.

List-komponentparametrar

Du kan ställa in följande parametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av `List`-komponenten: `allowMultipleSelection`, `dataProvider`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `multipleSelection`, `verticalLineScrollSize`, `verticalPageScrollSize` och `verticalScrollPolicy`. Var och en av de här parametrarna har en motsvarande `ActionScript`-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i avsnittet om klassen `List` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#). Information om hur du använder parametern `dataProvider` finns i ”Använda parametern `DataProvider`” på sidan 29.

Skapa ett program med komponenten List

I exemplen nedan beskrivs hur du lägger till en `List`-komponent i ett program när du redigerar.

Lägga till en enkel List-komponent i ett program

Här består List av etiketter som identifierar bilmodeller och datafält som innehåller priser.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en List-komponent från komponentpanelen till scenen.
- 3 Gör följande i egenskapsinspektören:
 - Ange förekomstnamnet **aList**.
 - Tilldela värdet **200** till W (bredd).
- 4 Använd textverktyget och skapa ett textfält nedanför **aList** och ge det förekomstnamnet **aTf**.
- 5 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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;
}
```

Den här koden använder metoden `addItem()` för att fylla i `aList` med tre objekt, och tilldela varje objekt ett `label`-värde, som visas i listan, samt ett `data`-värde. När det markerar ett objekt i listan anropar händelseavlyssnaren funktionen `showData()` som visar `data`-värdet för det markerade objektet.

- 6 Välj Kontroll > Testa filmen för att kompilera och köra programmet.

Fylla i en List-förekomst med en DataProvider

I det här exemplet skapas en List med bilmodeller och deras pris. Men här används en `DataProvider` för att fylla i List, i stället för metoden `addItem()`.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en List-komponent från komponentpanelen till scenen.
- 3 Gör följande i egenskapsinspektören:
 - Ange förekomstnamnet **aList**.
 - Tilldela värdet **200** till W (bredd).
- 4 Använd textverktyget och skapa ett textfält nedanför **aList** och ge det förekomstnamnet **aTf**.
- 5 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen för att visa List med dess objekt.

Använda en List-komponent för att styra en MovieClip-förekomst

I följande exempel skapas en List med namn på färger, och när en färg markeras används den på ett MovieClip.

- 1 Skapa ett Flash-dokument (ActionScript 3.0).
- 2 Dra List-komponenten från panelen Komponenter till scenen och ge den följande värden i egenskapsinspektören:
 - Ange **aList** som förekomstnamn.
 - Ange **60** som H-värde.
 - Ange **100** som X-värde.
 - Ange **150** som Y-värde.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen för att köra programmet.
- 5 Klicka på färger i List för att se dem visas i ett MovieClip.

Skapa en förekomst av en List-komponent med ActionScript

I det här exemplet skapas en enkel lista med ActionScript och den fylls med metoden `addItem()`.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra List-komponenten från panelen Komponenter till bibliotekspanelen.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen för att köra programmet.

Använda komponenten NumericStepper

Med komponenten `NumericStepper` kan användaren stega genom en ordnad uppsättning med siffror. Komponentens består av en siffra i en textruta som visas bredvid små upp- och nedpilar. När användaren trycker på knapparna ökar eller minskar siffran med den enhet som anges i parametern `stepSize` tills användaren släpper knappen, eller tills det högsta eller lägsta värdet nås. Texten i `NumericStepper`-komponentens textruta kan också redigeras.

En direktförhandsvisning av varje `NumericStepper`-förekomst reflekterar inställningen av värdeparametern i egenskapsinspektören eller komponentinspektören. Men det går inte att använda musen eller tangentbordet för att interagera med `NumericSteppers` pilknappar i direktförhandsvisningen.

Använda komponenten NumericStepper

Du kan använda `NumericStepper`-komponenten på alla ställen där du vill att användaren ska välja ett numeriskt värde. Du kan till exempel använda en `NumericStepper`-komponent i ett formulär för att ange månaden, dagen och året när ett kreditkort förfaller. Du kan också använda en `NumericStepper`-komponent för att göra så att användaren kan öka eller minska teckensnittsstorleken.

Komponenten `NumericStepper` hanterar enbart numeriska data. Du måste också ändra storlek på nummerlistan vid utvecklingen för att visa fler än två numeriska platser (till exempel siffrorna 5246 eller 1,34).

Du kan aktivera eller inaktivera en nummerlista i ett program. I inaktiverat läge tar en NumericStepper inte emot inmatningar från musen eller tangentbordet. När den är aktiverad får NumericStepper fokus om du klickar på eller tabbar till den, och dess interna fokus har angetts till textrutan. När en NumericStepper-förekomst har fokus kan du använda följande tangenter för att styra den:

Nyckel	Beskrivning
Nedpilen	Värdet ändras med en enhet.
Vänsterpil	Flyttar insättningspunkten till vänster inuti textrutan.
Högerpil	Flyttar insättningspunkten till höger inuti textrutan.
Skift+Tabb	Flyttar fokus till föregående objekt.
Tabb	Flyttar fokus till nästa objekt.
Upppilen	Värdet ändras med en enhet.

Mer information om att kontrollera fokus finns i avsnittet om gränssnittet FocusManager i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#) och ”Arbeta med FocusManager” på sidan 26.

NumericStepper-komponentparametrar

Du kan ställa in följande parametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av NumericStepper: `maximum`, `minimum`, `stepSize` och `value`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i avsnittet om klassen NumericStepper i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med NumericStepper

I proceduren nedan beskrivs hur du lägger till en NumericStepper-komponent i ett program när du redigerar. I exemplet placeras en NumericStepper-komponent och en Label-komponent på scenen, och en avlyssnare för en `Event.CHANGE`-händelse för NumericStepper-förekomsten skapas. När värdet i nummerlistan ändras visas det nya värdet i Label-förekomstens `text`-egenskap.

- 1 Dra en NumericStepper-komponent från panelen Komponenter till scenen.
- 2 Ange förekomstnamnet **aNs** i egenskapsinspektören.
- 3 Dra en Label-komponent från panelen Komponenter till scenen.
- 4 Ange förekomstnamnet **aLabel** i egenskapsinspektören.
- 5 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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;
};
```

I det här exemplet anges värdet på NumericStepper för etikettens `text`-egenskap. Funktionen `changeHandler()` uppdaterar etikettens `text`-egenskap när värdet i NumericStepper-förekomsten ändras.

- 6 Välj Kontroll > Testa filmen.

Skapa en NumericStepper med ActionScript:

I det här exemplet skapas tre NumericSteppers med ActionScript-kod, en var för att ange månaden, dagen och året i användarens födelsedatum. Labels läggs också till för en prompt och för identifierare för var och en av det tre NumericSteppers.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en Label till bibliotekspanelen.
- 3 Dra en NumericStepper-komponent till bibliotekspanelen.
- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen för att köra programmet.

Använda komponenten ProgressBar

Med komponenten ProgressBar visas förloppet vid inläsning av innehåll, vilket känns tryggt för användaren när innehållet är stort och kan fördröja körningen av programmet. ProgressBar är bra att använda om du vill visa förloppet för inlästa bilder och delar i ett program. Inläsningsprocessen kan vara bestämd eller obestämd. En *bestämd* förloppsindikator är en linjär representation av en uppgifts förlopp över tiden och används när mängden innehåll som ska läsas in är känd. En *obestämd* förloppsindikator används när mängden innehåll som ska läsas in är okänd. Du kan även lägga till en Label-komponent för att visas inläsningsförloppet som en procentsats.

Komponenten ProgressBar använder niosegmentsskalning och har ett indikatorskal, ett spårskal och ett obestämt skal.

Användarinteraktion med komponenten ProgressBar

Det finns tre lägen som du kan använda ProgressBar-komponenten i. De mest använda lägena är händelseläge och avsökningsläge. De här lägena anger en inläsningsprocess som antingen aktiverar `progress`- och `complete`-händelser (händelse- och avsökningsläge), eller visar `bytesLoaded`- och `bytesTotal`-egenskaper (avsökningsläge). Du kan också använda ProgressBar-komponenten i manuellt läge genom att ange egenskaperna `maximum`, `minimum` och `value` tillsammans med anrop till metoden `ProgressBar.setProgress()`. Du kan ställa in den obestämda egenskapen så att den anger om ProgressBar har en randig fyllning och en källa med okänd storlek (`true`) eller en täckande fyllning och en källa med känd storlek (`false`).

Du ställer in läget för ProgressBar genom att ange dess `mode`-egenskap, antingen med parametern `mode` i egenskapsinspektören eller komponentinspektören, eller genom att använda ActionScript.

Om du använder ProgressBar för att visa bearbetningsstatus, till exempel att analysera 100 000 objekt, och den finns i en loop med en bildruta, visas inga uppdateringar i ProgressBar eftersom skärmen inte ritas om.

ProgressBar-komponentparametrar

Du kan ställa in följande parametrar i egenskapsinspektören eller komponentinspektören för varje ProgressBar-förekomst: `direction`, `mode` och `source`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn.

Du kan skriva ActionScript för att styra de här och andra alternativ för ProgressBar-komponenten med hjälp av dess egenskaper, metoder och händelser. Mer information finns i avsnittet om klassen ProgressBar i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med ProgressBar

I proceduren nedan visas hur du lägger till en ProgressBar-komponent i ett program när du redigerar. I det här exemplet använder ProgressBar händelseläget. I händelseläget aktiverar det inlästa innehållet `progress`- och `complete`-händelserna som ProgressBar skickar för att ange förloppet. När `progress`-händelsen inträffar uppdaterar exemplet en etikett som visar hur stor procentandel av innehållet som har lästs in. När `complete`-händelsen inträffar visas ”Inläsning klar” och värdet på egenskapen `bytesTotal`, som är filens storlek.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en ProgressBar-komponent från panelen Komponenter till scenen.
 - Ange förekomstnamnet **aPb** i egenskapsinspektören.
 - Ange **200** för X-värdet i avsnittet Parametrar.
 - Ange **260** som Y-värde.
 - Välj `event` som `mode`-parameter.
- 3 Dra en Button-komponent från panelen Komponenter till scenen.
 - Ange förekomstnamnet **loadButton** i egenskapsinspektören.
 - Ange **220** som X-parameter.
 - Ange **290** som Y-parameter.
 - Ange **Load Sound** som `label`-parameter.
- 4 Dra Label-komponenten till scenen och ge den förekomstnamnet **progLabel**.
 - Ange **150** som W-värde.
 - Ange **200** som X-parameter.
 - Ange **230** som Y-parameter.
 - I avsnittet Parametrar rensar du värdet på `text`-parametern.
- 5 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod som läser in en mp3-ljudfil:

```
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 Välj Kontroll > Testa filmen.

Skapa ett program med ProgressBar-komponenten i avsökningsläge

I följande exempel ställs avsökningsläge in för ProgressBar. I avsökningsläget bestäms förloppet genom avlyssning av progress-händelser för innehållet som läses in och dess bytesLoaded- och bytesTotal-egenskaper används för att beräkna förloppet. Det här exemplet läser in ett ljudobjekt, avlyssnar dess progress-händelser och beräknar hur stor procentandel som har lästs in med hjälp av dess bytesLoaded- och bytesTotal-egenskaper. Den inlästa procentandelen visas både på en etikett och i utdatapanelen.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en ProgressBar-komponent från panelen Komponenter till scenen och ange följande värden i egenskapsinspektören:
 - Ange **aPb** som förekomstnamn.
 - Ange **185** som X-värde.
 - Ange **225** som Y-värde.
- 3 Dra en Label-komponent till scenen och ange följande värden i egenskapsinspektören:
 - Ange **progLabel** som förekomstnamn.
 - Ange **180** som X-värde.

- Ange **180** som Y-värde.
- I avsnittet Parametrar rensar du värdet på textparametern.

- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod som skapar ett Sound-objekt (aSound) och anropar loadSound() som läser in ett ljud i Sound-objektet:

```
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 Välj Kontroll > Testa filmen för att köra programmet.

Skapa ett program med ProgressBar-komponenten i manuellt läge

I följande exempel ställs manuellt läge in för ProgressBar. I manuellt läge måste du ange förloppet manuellt genom att anropa metoden setProgress() och ge den de aktuella och maximala värdena för att bestämma hur långt förloppet har kommit. Du anger inte egenskapen source i manuellt läge. I exemplet används en NumericStepper-komponent med ett maximalt värde på 250 för att öka ProgressBar stegvis. När värdet i NumericStepper ändras och utlöser en CHANGE-händelse anropar händelsehanteraren (nsChangeHandler) metoden setProgress() för att flytta fram ProgressBar. Den procentandel av förloppet som är klar visas också, baserat på det maximala värdet.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra ProgressBar-komponenten från panelen Komponenter till scenen och ge den följande värden i egenskapsinspektören:
 - Ange **aPb** som förekomstnamn.
 - Ange **180** som X-värde.
 - Ange **175** som Y-värde.
- 3 Dra en NumericStepper-komponent till scenen och ange följande värden i egenskapsinspektören:
 - Ange **aNs** som förekomstnamn.
 - Ange **220** som X-värde.
 - Ange **215** som Y-värde.
 - I parameteravsnittet anger du **250** för maximum-parametern, **0** för minimum-värdet, **1** för stepSize-parametern och **0** för value-parametern.
- 4 Dra en Label-komponent till scenen och ange följande värden i egenskapsinspektören:
 - Ange **progLabel** som förekomstnamn.

- Ange **150** som W-värde.
- Ange **180** som X-värde.
- Ange **120** som Y-värde.
- På fliken Parametrar rensar du värdeetiketten för textparametern.

5 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande kod:

```
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 Välj Kontroll > Testa filmen för att köra programmet.
- 7 Klicka på uppil på NumericStepper för att flytta fram ProgressBar.

Skapa en ProgressBar med ActionScript

I det här exemplet skapas en ProgressBar med ActionScript. Dessutom kopieras funktionen i det föregående exemplet, vilket skapar en ProgressBar i manuellt läge.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra ProgressBar-komponenten till bibliotekspanelen.
- 3 Dra NumericStepper-komponenten till bibliotekspanelen.
- 4 Dra Label-komponenten till bibliotekspanelen.
- 5 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande kod:

```
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 Välj Kontroll > Testa filmen för att köra programmet.
- 7 Klicka på uppil på NumericStepper för att flytta fram ProgressBar.

Använda komponenten RadioButton

Med komponenten RadioButton kan du tvinga användaren att göra ett enda val i en uppsättning med alternativ. Komponenterna måste användas i en grupp med minst två RadioButton-förekomster. Endast en medlem i gruppen kan väljas åt gången. Om en alternativknapp väljs i en grupp avmarkeras den knapp som är markerad för tillfället. Du ställer in `groupName`-parametern så att den anger vilken grupp en alternativknapp tillhör.

En alternativknapp är en grundläggande del av många formulärprogram på webben. Du använder alternativknappar när du vill att användaren ska göra ett val i en uppsättning med alternativ. Du använder till exempel alternativknappar i formulär som frågar efter vilket kreditkort en kund vill använda.

Använda komponenten RadioButton

En alternativknapp kan vara aktiverad eller inaktiverad. En inaktiverad alternativknapp tar inte emot mus- eller tangentbordsinmatning. Om användaren klickar på en RadioButton-komponentgrupp får endast den valda alternativknappen fokus. Användaren kan sedan använda följande tangenter för att styra den:

Nyckel	Beskrivning
Upppil/vänsterpil	Markeringen flyttar till den föregående alternativknappen i gruppen med alternativknappar.
Nedpil/högerpil	Markeringen flyttar till nästa alternativknapp i gruppen med alternativknappar.
Tabb	Flyttar fokus från gruppen med alternativknappar till nästa komponent.

Mer information om att kontrollera fokus finns i avsnittet om gränssnittet IFocusManager och i klassen FocusManager i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#) och ”Arbeta med FocusManager” på sidan 26.

En direktförhandsvisning av varje RadioButton-förekomst på scenen reflekterar ändringar som gjorts på parametrar i egenskapsinspektören eller komponentinspektören vid redigeringen. Men den ömsesidiga uteslutandet av markering visas inte i direktförhandsvisningen. Om du ger den markerade parametern värdet `true` för två alternativknappar i samma grupp ser båda ut som om de är markerade, även om endast den förekomst som skapades sist visas som markerad vid körningen. Mer information finns i ”Komponentparametrar för RadioButton” på sidan 76.

När du lägger till RadioButton-komponenten i ett program kan du göra den åtkomlig för en skärmläsare genom att lägga till följande rader med kod:

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

Du aktiverar bara en komponents åtkomlighet en gång, oavsett hur många förekomster du har av komponenten. Mer information finns i Kapitel 18, ”Skapa åtkomligt innehåll” i Använda Flash.

Komponentparametrar för RadioButton

Du kan ställa in följande utvecklingsparametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av RadioButton-komponenten: `groupName`, `label`, `labelPlacement`, `selected` och `value`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i avsnittet om klassen RadioButton i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Du kan skriva ActionScript för att ange ytterligare alternativ för RadioButton-förekomster med hjälp av metoderna, egenskaperna och händelserna i RadioButton-klassen.

Skapa ett program med komponenten RadioButton

I proceduren nedan beskrivs hur du lägger till RadioButton-komponenter i ett program när du redigerar. I det här exemplet används RadioButtons för att visa en fråga där svaret kan vara ja eller nej. Data från RadioButton visas i en TextArea.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra två RadioButton-komponenter från panelen Komponenter till scenen.
- 3 Markera den första alternativknappen. I egenskapsinspektören ger du den förekomstnamnet `yesRb` och gruppnamnet `rbGroup`.

- 4 Markera den andra alternativknappen. I egenskapsinspektören ger du den förekomstnamnet **noRb** och gruppnamnet **rbGroup**.
- 5 Dra en TextArea-komponent från panelen Komponenter till scenen och ge den förekomstnamnet **aTa**.
- 6 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen för att köra programmet.

Skapa en RadioButton med ActionScript

I det här exemplet används ActionScript för att skapa tre RadioButtons för färgerna röd, blå och grön, och en grå ruta ritas. Egenskapen `value` för varje RadioButton anger det hexadecimala värdet på färgen som associeras med knappen. När användaren klickar på en RadioButton anropar `clickHandler()`-funktionen `drawBox()` och överför färgen från RadioButtons `value`-egenskap för att färglägga rutan.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra RadioButton-komponenten till bibliotekspanelen.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen för att köra programmet.

Mer information finns i avsnittet om klassen `RadioButton` i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

Använda komponenten ScrollPane

Du kan använda komponenten `ScrollPane` för att visa innehåll som är för stort för området som det läses in till. Om du till exempel har en stor bild och bara ett litet utrymme för den i ett program, kan du läsa in den i en `ScrollPane`. `ScrollPane` tar emot filmklipp, JPEG-, PNG-, GIF- och SWF-filer.

Komponenter som `ScrollPane` och `UILoader` har `complete`-händelser som gör att du kan avgöra när innehållet har lästs in fullständigt. Om du vill ange egenskaper för innehållet i en `ScrollPane`- eller `UILoader`-komponent avlyssnar du `complete`-händelsen och anger egenskapen i händelsehanteraren. Med följande kod skapas till exempel en avlyssnare för `Event.COMPLETE` och en händelsehanterare som anger `alpha`-egenskapen för `ScrollPane`-innehållet till `.5`:

Använda UI-komponenterna

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

Om du anger en plats när du läser in innehåll till ScrollPane, måste du ange platsen som 0, 0 (X- och Y-koordinater). Följande kod läser till exempel in ScrollPane korrekt eftersom rutan ritas på platsen 0, 0:

```
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
```

Mer information finns i avsnittet om klassen ScrollPane i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Använda komponenten ScrollPane

En ScrollPane kan vara aktiverad eller inaktiverad. En inaktiverad ScrollPane tar inte emot mus- eller tangentbordsinmatning. Användare kan använda följande tangenter för att styra en ScrollPane när den har fokus:

Nyckel	Beskrivning
Nedpilen	Innehållet flyttas upp en lodrät radrullning.
Upppilen	Innehållet flyttas ned en lodrät radrullning.
End	Innehållet flyttas längst ned i ScrollPane.
Vänsterpil	Innehållet flyttas en vågrät radrullning till höger.
Högerpil	Innehållet flyttas en vågrät radrullning till vänster.
Hem	Innehållet flyttas högst upp i ScrollPane.
End	Innehållet flyttas längst ned i ScrollPane.
PageDown	Innehållet flyttas upp en lodrät sidrullning.
PageUp	Innehållet flyttas ned en lodrät sidrullning.

Användare kan använda musen för att interagera med ScrollPane, både på innehållet och på de lodräta och vågräta rullningslisterna. Användaren kan dra innehåll med musen när egenskapen `scrollDrag` har värdet `true`. När en handpekare visas över innehållet betyder det att användaren kan dra innehållet. Till skillnad från med de flesta kontroller inträffar åtgärderna när musknappen trycks ned, och fortsätter tills den släpps. Om innehållet har giltiga tabbstopp måste du ange värdet `false` för `scrollDrag`. Annars orsakar alla musträffar på innehållet rullningsdragning.

ScrollPane-komponentparametrar

Du kan ställa in följande parametrar för varje ScrollPane-instans i egenskapsinspektören eller komponentinspektören: `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `scrollDrag`, `source`, `verticalLineScrollSize`, `verticalPageScrollSize` och `verticalScrollPolicy`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i avsnittet om klassen ScrollPane i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Du kan skriva ActionScript för att styra de här och andra alternativ för en ScrollPane-komponent med hjälp av dess egenskaper, metoder och händelser.

Skapa ett program med komponenten ScrollPane

I proceduren nedan beskrivs hur du lägger till en ScrollPane-komponent i ett program när du redigerar. I det här exemplet läser ScrollPane in en bild från en sökväg som anges av egenskapen `source`.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra ScrollPane-komponenten från panelen Komponenter till scenen och ge den förekomstnamnet **aSp**.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen för att köra programmet.

Skapa en ScrollPane-förekomst med ActionScript

Exemplet skapar en ScrollPane, ställer in dess storlek och läser in en bild till den med hjälp av egenskapen `source`. Två avlyssnare skapas också. Den första avlyssnar en `scroll`-händelse och visar bildens position när användaren rullar lodrätt eller vågrätt. Den andra avlyssnar en `complete`-händelse och visar ett meddelande i utdatapanelen om att bilden har lästs in.

Det här exemplet skapar en ScrollPane med ActionScript och placerar ett MovieClip (en röd ruta) som är 150 pixlar brett och 300 pixlar högt i den.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra ScrollPane-komponenten från panelen Komponenter till bibliotekspanelen.
- 3 Dra en DataGrid-komponent från panelen Komponenter till bibliotekspanelen.
- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen för att köra programmet.

Använda komponenten Slider

Med komponenten Slider kan användaren välja ett värde genom att skjuta en grafisk *tumme* mellan ändpunkterna på ett spår som motsvarar ett intervall av värden. Du kan till exempel använda ett skjutreglage om du vill att användaren ska kunna välja ett värde, till exempel en siffra eller en procentsats. Du kan också använda ActionScript för att göra så att skjutreglagets värde påverkar ett annat objekts beteende. Du kan till exempel associera skjutreglaget med en bild och krympa eller förstora den baserat på skjutreglagets relativa position eller värde.

Det aktuella värdet på Slider bestäms av reglagets relativa plats mellan spårets ändpunkter eller min- och maxvärdena för Slider.

Med Slider kan du ha ett kontinuerligt intervall av värden mellan min- och maxvärdena, med du kan också ställa in parametern `snapInterval` så att den anger fasta intervall mellan min- och maxvärdena. En Slider kan visa skalmarkeringar, som är oberoende skjutreglagets tilldelade värden, vid angivna intervall längs spåret.

Slider har en vågrät orientering som standard, men du kan ge den en lodrät orientering genom att ställa in `direction`-parametern till vertikal. Skjutreglagets spår sträcker sig från ena änden till den andra, och skalmarkeringarna placeras från vänster till höger precis ovanför spåret.

Användarinteraktion med Slider-komponenten

När en Slider-förekomst har fokus kan du använda följande tangenter för att styra den:

Nyckel	Beskrivning
Högerpil	Ökar det associerade värdet för ett vågrätt reglage.
Uppilen	Ökar det associerade värdet för ett lodrätt reglage.
Vänsterpil	Minskar det associerade värdet för ett vågrätt reglage.

Nyckel	Beskrivning
Nedpilen	Minskar det associerade värdet för ett lodrätt reglage.
Skift+Tabb	Flyttar fokus till föregående objekt.
Tabb	Flyttar fokus till nästa objekt.

Mer information om att kontrollera fokus finns i avsnittet om gränssnittet IFocusManager och i klassen FocusManager i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#) och ”Arbeta med FocusManager” på sidan 26.

En direktförhandsvisning av varje Slider-förekomst reflekterar ändringar som gjorts på parametrar i egenskapsinspektören eller komponentinspektören vid redigeringen.

Slider-komponentparametrar

Du kan ställa in följande utvecklingsparametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av Slider-komponenten: `direction`, `liveDragging`, `maximum`, `minimum`, `snapInterval`, `tickInterval` och `value`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i avsnittet om klassen Slider i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med Slider

Följande exempel skapar en Slider-förekomst som gör att användaren kan uttrycka sin nöjdhetnivå med en hypotetisk händelse. Användaren flyttar reglaget till höger eller vänster för att ange en högre eller lägre nöjdhetsnivå.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en Label-komponent från panelen Komponenter till mitten av scenen.
 - Ge den förekomstnamnet **valueLabel**.
 - Tilldela värdet **0procent** till `text`-parametern.
- 3 Dra en Slider-komponent från panelen Komponenter och centrera den nedanför `value_lbl`.
 - Ge den förekomstnamnet **aSlider**.
 - Tilldela den en bredd (W:) på **200**.
 - Tilldela den en höjd (H:) på **10**.
 - Tilldela ett värde på **100** till `maximum`-parametern.
 - Tilldela ett värde på **10** till både `snapInterval`- och `tickInterval`-parametern.
- 4 Dra en annan Label-förekomst från bibliotekspanelen och centrera den nedanför `aSlider`.
 - Ge den förekomstnamnet **promptLabel**.
 - Tilldela den en bredd (W:) på 250.
 - Tilldela den en höjd (H:) på 22.
 - Skriv **Ange din nöjdhetsnivå** som `text`-parameter.
- 5 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen.

När du flyttar skjutreglaget från ett intervall till ett annat i det här exemplet uppdaterar en avlyssnare för `SliderEvent.CHANGE`-händelsen egenskapen `text` för `valueLabel` så att den visar den procentandel som motsvarar reglagets position.

Skapa ett program med Slider-komponenten med ActionScript

Följande exempel skapar en Slider med ActionScript. Exemplet läser in en bild av en blomma och använder Slider så att användaren kan tona ut bilden eller göra den ljusare genom att ändra dess `alpha`-egenskap så att den motsvarar värdet på Slider.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra Label-komponenten och Slicer-komponenten från panelen Komponenter till det aktuella dokumentets bibliotekspanel.
Komponenterna läggs till i biblioteket, men de syns inte i programmet.
- 3 Öppna panelen Åtgärder, markera bildruta 1 i huvudtidslinjen och ange följande kod för att skapa och placera komponentförekomsterna:

```
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 Välj Kontroll > Testa filmen för att köra programmet.

5 Flytta skjutreglaget till vänster om du vill tona ut bilden och till höger om du vill göra den ljusare.

Använda komponenten TextArea

Komponenten TextArea är en wrapper för det ursprungliga ActionScript TextField-objektet. Du kan använda TextArea-komponenten för att visa text och även för att redigera textinmatning om egenskapen `editable` har värdet `true`. Komponentens visningsområde kan utöka sig över flera rader med text och radbryter långa textrader om egenskapen `wordWrap` har värdet `true`. Med egenskapen `restrict` kan du begränsa de tecken som användaren kan ange och med `maxChars` kan du ange det maximala antalet tecken som användaren kan ange. Om texten överstiger textområdets vågräta och lodräta gränser visas vågräta och lodräta rullningslistor automatiskt, såvida inte deras associerade egenskaper, `horizontalScrollPolicy` och `verticalScrollPolicy`, har värdet `off`.

Du använder en `TextArea`-komponent när du behöver ett textfält med flera rader. Du kan till exempel använda en `TextArea`-komponent som ett kommentarsfält i ett formulär. Sedan kan du ställa in en avlyssnare som kontrollerar om fältet tomt när användaren tabbar ut ur fältet. Avlyssnaren kan visa ett felmeddelande om att en kommentar måste anges i fältet.

Om du behöver ett textfält med en enda rad använder du `TextInput`-komponenten.

Du kan ställa in `textFormat`-stilen med metoden `setStyle()` för att ändra stilen på texten som visas i en `TextArea`-förekomst. Du kan också formatera en `TextArea`-komponent med HTML genom att använda egenskapen `htmlText` i `ActionScript`, och du kan ställa in `displayAsPassword`-egenskapen på `true` för att maskera text med asterisker. Om du ställer in egenskapen `condenseWhite` på `true`, tar Flash bort extra vitt utrymme i ny text som beror på mellanslag, radbrytningar och så vidare. Det påverkar inte text som redan finns i kontrollen.

Använda komponenten `TextArea`

En `TextArea`-komponent kan aktiveras eller inaktiveras i ett program. I inaktiverad status kan den inte ta emot inmatningar från mus eller tangentbord. I aktiverad status följer den samma fokus, val och navigeringsregler som `ActionScript` `TextField`-objekt. När en `TextArea`-förekomst har fokus kan du använda följande tangenter för att styra den:

Nyckel	Beskrivning
Piltangenter	Flytta insättningspunkten uppåt, nedåt, till vänster eller till höger, om texten är redigerbar.
Page Down	Flyttar insättningspunkten till slutet av texten, om texten är redigerbar.
Page Up	Flyttar insättningspunkten till början av texten, om texten är redigerbar.
Skift+Tabb	Flyttar fokus till föregående objekt i tabbslingan.
Tabb	Flyttar fokus till nästa objekt i tabbslingan.

Mer information om att kontrollera fokus finns i avsnittet om gränssnittet `FocusManager` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#) och ”Arbeta med `FocusManager`” på sidan 26.

TextArea-komponentparametrar

Du kan ställa in följande utvecklingsparametrar för varje `TextArea`-förekomst i egenskapsinspektören eller komponentinspektören: `condenseWhite`, `editable`, `horizontalScrollPolicy`, `maxChars`, `restrict`, `text`, `verticalScrollPolicy` och `wordwrap`. Var och en av de här parametrarna har en motsvarande `ActionScript`-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i avsnittet om klassen `TextArea` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

En direktförhandsvisning av varje `TextArea`-förekomst reflekterar ändringar som gjorts på parametrar i egenskapsinspektören eller komponentinspektören vid redigeringen. Om en rullningslist behövs visas den i direktförhandsvisningen, men den fungerar inte där. Det går inte att markera text i direktförhandsvisningen, och du kan inte ange text i komponentförekomsten på scenen.

Du kan skriva `ActionScript` för att styra de här och andra alternativ för `TextArea`-komponenten med hjälp av dess egenskaper, metoder och händelser. Mer information finns i avsnittet om klassen `TextArea` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med komponenten TextArea

I proceduren nedan beskrivs hur du lägger till en TextArea-komponent i ett program när du redigerar. Exemplet anger en `focusOut`-händelsehanterare i TextArea-förekomsten, som bekräftar att användaren har skrivit något i textområdet innan fokus flyttas till en annan del av gränssnittet.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en TextArea-komponent från panelen Komponenter till scenen och ge den förekomstnamnet **aTa**. Låt parametrarna ha kvar standardinställningarna.
- 3 Dra ytterligare en TextArea-komponent från panelen Komponenter till scenen, placera den nedanför den första och ge den förekomstnamnet **bTa**. Låt parametrarna ha kvar standardinställningarna.
- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
import flash.events.FocusEvent;

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

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

Det här exemplet begränsar tecknen som du kan ange i aTa-textområdet till små bokstäver, kommatecken, apostrof och mellanslag. Det anger också händelsehanterare för händelserna `change`, `KEY_FOCUS_CHANGE` och `MOUSE_FOCUS_CHANGE` för aTa-textområdet. Funktionen `changeHandler()` gör så att texten som du anger i textområdet aTa text automatiskt visas i textområdet bTa, genom att tilldela `aTa.text` till `bTa.text` vid varje `change`-händelse. Funktionen `k_m_fHandler()` för händelserna `KEY_FOCUS_CHANGE` och `MOUSE_FOCUS_CHANGE` förhindrar att du trycker på tabbtangenten för att flytta till nästa fält utan att ange text. Det gör den genom att förhindra standardbeteendet.

- 5 Välj Kontroll > Testa filmen.

Om du trycker på tabbtangenten för att flytta fokus till det andra textområdet utan att ange text ska ett felmeddelande visas, och fokus ska återgå till det första textområdet. När du anger text i det första textområdet visas den duplicerad i det andra textområdet.

Skapa en TextArea-förekomst med ActionScript

Följande exempel skapar en TextArea-komponent med ActionScript. Egenskapen `condenseWhite` får värdet `true` för att dra ihop tomrum och tilldela text till egenskapen `htmlText` och dra nytta av formateringsattributen för HTML-text.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra TextArea-komponenten till bibliotekspanelen.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
import fl.controls.TextArea;

var aTa:TextArea = new TextArea();

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

I det här exemplet används egenskapen `htmlText` för att använda HTML-attributen fetstil och understrykning på ett textblock och visa det i textområdet `a_ta`. I exemplet ställs också egenskapen `condenseWhite` in på `true` för att dra ihop tomrum inuti textblocket. Metoden `setSize()` ställer in textområdets höjd och bredd, och metoden `move()` ställer in dess position. Metoden `addChild()` lägger till `TextArea`-förekomsten på scenen.

4 Välj Kontroll > Testa filmen.

Använda komponenten TextInput

Komponenten `TextInput` är en enkelradstextkomponent som är en wrapper för det ursprungliga `ActionScript TextField`-objektet. Om du behöver ett textfält med flera rader använder du `TextArea`-komponenten. Du kan till exempel använda en `TextInput`-komponent som ett lösenordsfält i ett formulär. Du kan också ställa in en avlyssnare som kontrollerar om fältet har tillräckligt många tecken när användaren tabbar ut ur fältet. Avlyssnaren kan visa ett felmeddelande som anger att korrekt antal tecken måste skrivas in.

Du kan ställa in `textFormat`-egenskapen med metoden `setStyle()` för att ändra stilen på texten som visas i en `TextInput`-förekomst. En `TextInput`-komponent kan även formateras med HTML, eller som ett lösenordsfält som döljer texten.

Användarinteraktion med komponenten TextInput

En `TextInput`-komponent kan aktiveras eller inaktiveras i ett program. I inaktiverad status tas inmatning från mus eller tangentbord inte emot. I aktiverad status följer den samma fokus, val och navigeringsregler som `ActionScript TextField`-objekt. När en `TextInput`-förekomst har fokus kan du också använda följande tangenter för att styra den:

Nyckel	Beskrivning
Piltangenter	Flyttar insättningspunkten ett tecken åt vänster eller höger.
Skift+Tabb	Flyttar fokus till föregående objekt.
Tabb	Flyttar fokus till nästa objekt.

Mer information om att kontrollera fokus finns i avsnittet om gränssnittet `FocusManager` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#) och ”Arbeta med `FocusManager`” på sidan 26.

En direktförhandsvisning av varje `TextInput`-förekomst reflekterar ändringar som gjorts för parametrar i egenskapsinspektören eller komponentinspektören vid redigeringen. Det går inte att markera text i direktförhandsvisningen, och du kan inte ange text i komponentförekomsten på scenen.

När du lägger till TextInput-komponenten i ett program kan du använda hjälpmedelspanelen för att göra den åtkomlig för skärmläsare.

TextInput-komponentparametrar

Du kan ställa in följande utvecklingsparametrar för varje förekomst av TextInput-komponenten i egenskapsinspektören eller komponentinspektören: `editable`, `displayAsPassword`, `maxChars`, `restrict` och `text`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om möjliga värden på de här parametrarna finns i avsnittet om klassen TextInput i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Du kan skriva ActionScript för att styra de här och andra alternativ för TextInput-komponenten med hjälp av dess egenskaper, metoder och händelser. Mer information finns i avsnittet om klassen TextInput i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med komponenten TextInput

I proceduren nedan beskrivs hur du lägger till en TextInput-komponent i ett program. Exemplet använder två TextInput-fält för att ta emot och bekräfta ett lösenord. En händelseavlyssnare används för att se om minst åtta tecken har angetts, och att texten för de två fälten överensstämmer.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en Label-komponent från panelen Komponenter till scenen och ge den följande värden i egenskapsinspektören:
 - Ange förekomstnamnet **pwdLabel**.
 - Ange värdet **100** för W.
 - Ange värdet **50** för X.
 - Ange värdet **150** för Y.
 - I avsnittet Parametrar anger du värdet **Lösenord:** för textparametern.
- 3 Dra ytterligare en Label-komponent från panelen Komponenter till scenen och ge den följande värden:
 - Ange förekomstnamnet **confirmLabel**.
 - Ange värdet **100** för W.
 - Ange värdet **50** för X.
 - Ange värdet **200** för Y.
 - I avsnittet Parametrar anger du värdet **Bekräftalösenordet:** för textparametern.
- 4 Dra ytterligare en TextInput-komponent från panelen Komponenter till scenen och ge den följande värden:
 - Ange förekomstnamnet **pwdTi**.
 - Ange värdet **150** för W.
 - Ange värdet **190** för X.
 - Ange värdet **150** för Y.
 - I avsnittet Parametrar dubbelklickar du på värdet för parametern `displayAsPassword` och väljer **true**. Då maskeras värdet som anges i textfältet med asterisker.
- 5 Dra ytterligare en TextInput-komponent från panelen Komponenter till scenen och ge den följande värden:
 - Ange förekomstnamnet **confirmTi**.

- Ange värdet **150** för W.
- Ange värdet **190** för X.
- Ange värdet **200** för Y.
- I avsnittet Parametrar dubbelklickar du på värdet för parametern `displayAsPassword` och väljer **true**. Då maskeras värdet som anges i textfältet med asterisker.

6 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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);
```

Den här koden ställer in en `enter`-händelsehanterare för `TextInput`-förekomsten med namnet `confirmTi`. Om de två lösenorden inte överensstämmer med varandra, eller om användaren skriver färre än åtta tecken visas följande felmeddelande: "Fel lösenord. Ange lösenordet igen." Om lösenorden är åtta eller fler tecken långa och de överensstämmer, visas värdet som angavs i utdatapanelen.

7 Välj Kontroll > Testa filmen.

Skapa en `TextInput`-förekomst med ActionScript

Med följande exempel skapas en `TextInput`-komponent med ActionScript. Exemplet skapar också en `Label` som används för att uppmana användaren att ange sitt namn. Komponentens `restrict`-egenskap ställs in så att enbart versala och gemena bokstäver, punkt och mellanslag tillåts. Dessutom skapas ett `TextFormat`-objekt som används för att formatera texten i `Label`- och `TextInput`-komponenterna.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra `TextInput`-komponenten från panelen Komponenter till bibliotekspanelen.
- 3 Dra en `Label`-komponent från panelen Komponenter till bibliotekspanelen.
- 4 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen för att köra programmet.

Använda komponenten TileList

Komponenten `TileList` består av en lista rader och kolumner med data från en `DataProvider`. Ett *objekt* refererar till en enhet med data som lagras i en cell i `TileList`. Ett objekt, som kommer från en `DataProvider`, har vanligtvis en `label`-egenskap och en `source`-egenskap. Egenskapen `label` identifierar innehållet som ska visas i en cell och `source` tillhandahåller ett värde för den.

Du kan skapa en `Array`-förekomst eller hämta en från en server. `TileList`-komponenten har metoder som företräder dess `DataProvider`, till exempel metoderna `addItem()` och `removeItem()`. Om ingen extern `DataProvider` finns med i listan skapas en `DataProvider`-förekomst automatiskt med hjälp av dessa metoder, vilken visas genom `List.dataProvider`.

Använda komponenten TileList

`TileList` renderar varje cell som använder en `Sprite` som implementerar `ICellRenderer`-gränssnittet. Du kan ange den här renderaren med `TileList`-egenskapen `cellRenderer`. `TileList`-komponentens standard-`CellRenderer` är `ImageCell`, som visar en bild (klass, bitmapp, förekomst eller URL) och en valfri etikett. Etiketten består av en enda rad som justeras mot nederkanten av cellen. Du kan bara rulla en `TileList` i en riktning.

När en `TileList`-förekomst har fokus kan du också använda följande tangenter för att komma åt objekt i den:

Nyckel	Beskrivning
Upppil och nedpil	Gör att du kan flytta uppåt och nedåt i en kolumn. Om egenskapen <code>allowMultipleSelection</code> är <code>true</code> kan du använda de här tangenterna tillsammans med Skift-tangenten för att markera flera celler.
Högerpil och vänsterpil	Gör så att du kan flytta till vänster eller höger i en rad. Om egenskapen <code>allowMultipleSelection</code> är <code>true</code> kan du använda de här tangenterna tillsammans med Skift-tangenten för att markera flera celler.
Hem	Markerar den första cellen i en <code>TileList</code> . Om egenskapen <code>allowMultipleSelection</code> är <code>true</code> kan du hålla ned Skift-tangenten och trycka på Home-tangenten för att markera alla celler från den aktuella markeringen till den första cellen.
End	Markerar den sista cellen i en <code>TileList</code> . Om egenskapen <code>allowMultipleSelection</code> är <code>true</code> kan du hålla ned Skift-tangenten och trycka på End-tangenten för att markera alla celler från den aktuella markeringen till den sista cellen.
Ctrl	Om egenskapen <code>allowMultipleSelection</code> har värdet <code>true</code> kan du markera flera celler, utan någon särskild ordning.

När du lägger till `TileList` -komponenten i ett program kan du göra den åtkomlig för en skärmläsare genom att lägga till följande rader med ActionScript-kod:

```
import fl.accessibility.TileListAccImpl;

TileListAccImpl.enableAccessibility();
```

Du aktiverar bara en komponents åtkomlighet en gång, oavsett hur många förekomster komponenten har. Mer information finns i kapitel 18, "Skapa åtkomligt innehåll" i *Använda Flash*.

TileList-komponentparametrar

Du kan ställa in följande utvecklingsparametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av `TileList`-komponenten: `allowMultipleSelection`, `columnCount`, `columnWidth`, `dataProvider`, `direction`, `horizontalScrollLineSize`, `horizontalScrollPageSize`, `labels`, `rowCount`, `rowHeight`, `scrollPolicy`, `verticalScrollLineSize` och `verticalScrollPageSize`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. Information om hur du använder parametern `dataProvider` finns i "Använda parametern `DataProvider`" på sidan 29.

Du kan skriva ActionScript för att ange ytterligare alternativ för `TileList`-förekomster genom att använda dess metoder, egenskaper och händelser. Mer information finns i avsnittet om klassen `TileList` i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

Skapa ett program med komponenten TileList

I det här exemplet används `MovieClips` för att fylla i en `TileList` med en array med färger.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en `TileList`-komponent till scenen och ge den förekomstnamnet `aTL`.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen för att testa programmet.

Skapa en TileList-komponent med ActionScript

I det här exemplet skapas en TileList-instans och instanser av komponenterna ColorPicker, ComboBox, NumericStepper och CheckBox läggs till i den. Exemplet skapar en Array som innehåller etiketter och namnen på de komponenter som ska visas, och tilldelar Arrayen (dp) till TileList-egenskapen dataProvider. Egenskaperna columnWidth och rowHeight samt metoden setSize() används för att göra en layout för TileList, metoden move() för att placera den på scenen, stilen contentPadding för att lägga in utrymme mellan TileList-förekomstens kantlinjer och dess innehåll, samt metoden sortItemsOn() för att sortera innehållet efter etiketterna.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra följande komponenter från panelen Komponenter till bibliotekspanelen. ColorPicker, ComboBox, NumericStepper, CheckBox och TileList.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:


```
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 Välj Kontroll > Testa filmen för att testa programmet.

Använda komponenten UILoader

Komponenten UILoader är en behållare som kan visa filer av typen SWF, JPEG, progressiv JPEG, PNG och GIF. Du använder en UILoader när du behöver hämta innehåll från en fjärranslutning och föra in det i ett Flash-program. Du kan till exempel använda en UILoader för att lägga till en företagslogotyp (JPEG-fil) i ett formulär. Du kan också använda komponenten UILoader i ett program som visar foton. Använd metoden `load()` för att läsa in innehåll, egenskapen `percentLoaded` för att avgöra hur mycket innehåll som har lästs in och händelsen `complete` för att avgöra när inläsningen är klar.

Du kan skala innehållet i UILoader eller ändra storlek på själva UILoader för att få plats med hela innehållet. Som standard skalas innehållet så att det passar i UILoader. Du kan också läsa in innehåll vid körning och övervaka inläsningsprocessen (när innehållet har lästs in en gång lagras det i cacheminnet, så förloppet hoppar snabbt till 100 %). Om du anger en plats när du läser in innehåll i UILoader måste du ange platsen som 0, 0 (X- och Y-koordinater).

Använda komponenten UILoader

En UILoader-komponent kan inte få fokus. Men innehåll som har lästs in i UILoader-komponenten kan få fokus och ha egna fokusinteraktioner. Mer information om att kontrollera fokus finns i avsnittet om gränssnittet FocusManager i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#) och ”Arbeta med FocusManager” på sidan 26.

UILoader-komponentparametrar

Du kan ställa in följande utvecklingsparametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av UILoader-komponenten: `autoLoad`, `maintainAspectRatio`, `source` och `scaleContent`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn.

En direktförhandsvisning av varje UILoader-förekomst reflekterar ändringar som gjorts för parametrar i egenskapsinspektören eller komponentinspektören vid redigeringen.

Du kan skriva ActionScript för att ange ytterligare alternativ för UILoader-förekomster genom att använda dess metoder, egenskaper och händelser. Mer information finns i avsnittet om klassen UILoader i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med komponenten UILoader

I proceduren nedan beskrivs hur du lägger till en UILoader-komponent i ett program när du redigerar. I det här exemplet läser UILoader in en GIF-bild av en logotyp.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en UILoader-komponent från panelen Komponenter till scenen.
- 3 Ange förekomstnamnet `aUI` i egenskapsinspektören.
- 4 Markera UILoader på scenen och i komponentinspektören, och ange `http://www.helpexamples.com/images/logo.gif` för `source`-parametern.

Skapa en förekomst av en UILoader-komponent med ActionScript

Med det här exemplet skapas en UILoader-komponent med ActionScript och en JPEG-bild av en blomma läses in. När `complete`-händelsen inträffar visas antalet inlästa byte i utdatapanelen.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra UILoader-komponenten från panelen Komponenter till bibliotekspanelen.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

```
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 Välj Kontroll > Testa filmen.

Använda komponenten UIScrollBar

Med komponenten UIScrollBar-komponent som gör att du kan lägga till en rullningslist i ett textfält. Du kan lägga till en rullningslist i ett textfält under utvecklingen, eller under körning med ActionScript. Om du vill använda UIScrollBar-komponenten skapar du ett textfält på scenen och drar UIScrollBar-komponenten från panelen Komponenter till någon av kvadranterna i textfältets begränsningsram.

Om rullningslistens är kortare än den kombinerade storleken på rullningspilarna visas den inte korrekt. En av piltangenterna döljs bakom den andra. Flash har ingen felsökning för det här. I det här fallet är det en god idé att dölja rullningslistan med ActionScript. Om rullningslistan har en sådan storlek att det inte finns plats för rullningrutan (reglaget) gör Flash rullningsrutan osynlig.

Komponenten UIScrollBar fungerar på samma sätt som alla rullningslistor. Den har pilknappar i varje ände, och ett rullningsspår och en rullningsruta (reglage) däremellan. Den kan fästas vid någon av kanterna på ett textfält och användas både lodrätt och vågrätt.

Mer information om TextField finns i avsnittet om klassen TextField i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Använda komponenten UIScrollBar

Till skillnad från många andra komponenter kan UIScrollBar ta emot kontinuerliga indata från musen, till exempel när användaren håller ned musknappen, i stället för att kräva upprepade klickningar.

Det finns ingen tangentbordsinteraktion för UIScrollBar-komponenten.

UIScrollBar-komponentparametrar

Du kan ställa in följande utvecklingsparametrar i egenskapsinspektören eller komponentinspektören för varje förekomst av UIScrollBar-komponenten: `direction` och `scrollTargetName`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn.

Du kan skriva ActionScript för att ange ytterligare alternativ för UIScrollBar-förekomster genom att använda metoder, egenskaper och händelser för klassen. Mer information finns i avsnittet om klassen UIScrollBar i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Skapa ett program med komponenten UIScrollBar

I proceduren nedan beskrivs hur du lägger till en UIScrollBar-komponent i ett program när du redigerar.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Skapa ett dynamiskt textfält som är tillräckligt högt för att en eller två textrader ska få plats, och ge det förekomstnamnet `myText` i egenskapsinspektören.
- 3 I egenskapsinspektören anger du Multiline eller Multiline No Wrap som värde för textindatafältets Line Type om du tänker använda rullningslistan vågrätt.
- 4 Öppna åtgärdspanelen, välj bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod för att fylla `text`-egenskapen så att användaren måste rulla nedåt för att se allt:

```
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."
```

Obs! Se till att textfältet på scenen är så litet att du behöver rulla i det för att se all text. Om det inte är tillräckligt litet visas inte rullningslistan, eller också visas den som två rader utan reglage (den del som du drar för att rulla innehållet).

- 5 Kontrollera att objektfästning är aktiverad (Visa > Fästning > Fäst mot objekt).
- 6 Dra en UIScrollBar-förekomst från panelen Komponenter till textindatafältet, i närheten av den sida som du vill fästa den vid. Komponenten måste överlappa textfältet när du släpper musknappen för att den ska fästa ordentligt vid fältet. Ge den förekomstnamnet **mySb**.

Komponentens `scrollTargetName`-egenskap fylls automatiskt i med textfältets förekomstnamn i egenskapsinspektören och komponentinspektören. Om den inte visas på fliken Parametrar kan det hända att du inte har överlappat UIScrollBar-förekomsten tillräckligt.

- 7 Välj Kontroll > Testa filmen.

Skapa en förekomst av en UIScrollBar-komponent med ActionScript

Du kan skapa en UIScrollBar-förekomst med ActionScript och associera den med ett textfält vid körningen. Följande exempel skapar en vågrät UIScrollBar-förekomst och fäster den vid nederkanten på en textfältförekomst som kallas **myTxt**, som fylls i med text från en URL. I exemplet ställs också rullningslistans storlek in så att den överensstämmer med textfältets storlek:

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra ScrollBar-komponenten till bibliotekspanelen.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande ActionScript-kod:

Använda UI-komponenterna

```
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 Välj Kontroll > Testa filmen.

Kapitel 5: Anpassa UI-komponenterna

Om anpassning av UI-komponenter

Du anpassar utseendet på komponentern i programmen genom att ändra något eller båda av följande element:

Stilar Alla komponenter har en uppsättning stilar som du kan ställa in för att ange vilka värden Flash använder för att återge komponentens utseende. Stilar anger vanligtvis skal och ikoner som används för en komponent i olika lägen, och även vilken textformatering och vilka utfyllningsvärden som ska användas.

Skins Ett skal består av den samling symboler som utgör komponentens grafiska utseende i ett viss läge. Medan en stil anger vilket skal som ska användas, är ett skal ett grafiskt element som Flash använder för att rita komponenten. *Skalning är processen att ändra en komponents utseende genom att ändra eller byta ut dess grafik.*

Obs! Standardutseendet på komponenter i ActionScript 3.0 kan anses vara att tema (Aeon Halo), men de här skalerna är inbyggda i komponenterna. Komponenterna i ActionScript 3.0 fungerar inte med de externa temafilerna som komponenterna i ActionScript 2.0 fungerade med.

Ställa in stilar

En komponents stil anger vanligtvis värden för dess skal, ikoner, textformatering och utfyllning när Flash ritat komponenten i olika lägen. Flash ritat till exempel en Button med ett annat skal för att visa dess ned-läge, som inträffar när du klickar med musknappen på den, än skalet som används för att visa upp-läget eller det normala läget. Ett annat skal används också för att visa att knappen är i inaktiverat läge, vilket inträffar när egenskapen `enabled` har värdet `false`.

Du kan ställa in stilar för komponenter på dokument-, klass- och förekomstnivå. Dessutom kan vissa stilegenskaper ärvas från en överordnad komponent. List-komponenten ärver till exempel ScrollBar-stilar från BaseScrollPane.

Du kan ställa in stilar för att anpassa en komponent på följande sätt:

- Ställa in stilar för en komponentförekomst. Du kan ändra färg- och textegenskaperna för en enda komponentförekomst. Det är effektivt i vissa situationer, men det kan vara tidsödande om du måste ställa in enskilda egenskaper för alla komponenter i ett dokument.
- Ställa in stilar för alla komponenter av en viss typ i ett dokument. Om du vill ha ett konsekvent utseende på alla komponenter av en viss typ, till exempel alla CheckBoxes eller Buttons i ett dokument, ställer du in stilar på komponentnivå.

Värdena på stilegenskaperna som ställs in för behållare ärvs av komponenterna i behållaren.

Flash visar inte ändringar som görs i stilegenskaper när du visar komponenterna på scenen med funktionen direktförhandsvisning.

Förstå stilinställningar

Nedan beskrivs några huvudpunkter om att använda stilar:

Arv En underordnad komponent är som standard inställd så att den ärver en stil från den överordnade komponenten. Du kan inte ställa in arv för stilar i ActionScript.

Prioritet Om en komponentstil har ställts in på flera sätt använder Flash den första stilen som den träffar på enligt dess rangordning. Flash letar efter stilar i följande ordning tills ett värde hittas:

- 1 Flash letar efter en stilegenskap i komponentförekomsten.
- 2 Om stilen är en av de ärvande stilarna letar Flash igenom den överordnade hierarkin för ett ärvt värde.
- 3 Flash letar efter stilen i komponenten.
- 4 Flash letar efter en global inställning i StyleManager.
- 5 Om egenskapen fortfarande inte är definierad har egenskapen värdet `undefined`.

Komma åt en komponents standardstilar

Du kan komma åt en komponents standardstilar genom att använda den statiska `getStyleDefinition()`-metoden för komponentklassen. Följande kod hämtar till exempel standardstilarna för ComboBox-komponenten och visar standardvärdena för egenskaperna `buttonWidth` och `downArrowDownSkin`:

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

Ställa in och få format på en komponentförekomst

Alla förekomster av UI-komponenter kan anropa metoderna `setStyle()` och `getStyle()` direkt för att ställa in eller hämta en stil. Följande syntax ställer in en stil och ett värde för en komponentförekomst:

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

Den här syntaxen hämtar en stil för en komponentförekomst:

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

Observera att metoden `getStyle()` returnerar typen `Object` eftersom den kan returnera flera stilar med olika datatyper. Följande kod ställer till exempel in teckensnittsstilen för en TextArea-förekomst (`aTa`) och hämtar den sedan med metoden `getStyle()`. Exemplet byter ut det returnerade värdet mot ett `TextFormat`-objekt för att tilldela det till en `TextFormat`-variabel. Utan bytet skickar compilatorn ett fel för ett försök att tvinga en `Object`-variabel till en `TextFormat`-variabel.

```
import flash.text.TextFormat;

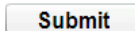
var tf:TextFormat = new TextFormat();
tf.font = "Georgia";
aTa.setStyle("textFormat", tf);
aTa.text = "Hello World!";
var aStyle:TextFormat = aTa.getStyle("textFormat") as TextFormat;
trace(aStyle.font);
```

Ställa in textegenskaper med TextFormat

Använd `TextFormat`-objektet för att formatera text för en komponentförekomst. `TextFormat`-objektet har egenskaper som gör att du kan ange textegenskaper som `bold`, `bullet`, `color`, `font`, `italic`, `size`, bland andra. Du ställer in de här egenskaperna i `TextFormat`-objektet och anropar sedan metoden `setStyle()` för att använda dem på en komponentförekomst. Följande kod ställer till exempel in egenskaperna `font`, `size` och `bold` för ett `TextFormat`-objekt och använder dem på en `Button`-förekomst:

```
/* 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);
```

Bilden nedan visar vilken effekt de här inställningarna har på en knapp med en Submit-etikett:



Stilegenskaper som ställs in för en komponentförekomst med `setStyle()` har den högsta prioriteten och åsidosätter alla andra stilinställningar. Ju fler egenskaper du ställer in med `setStyle()` för en komponentförekomst, desto långsammare återges komponenten vid körning.

Ställa in ett format för alla förekomster av en komponent

Du kan ställa in en stil för alla förekomster av en komponentklass med den statiska metoden `setComponentStyle()` i `StyleManager`-klassen. Du kan till exempel ställa in att textfärgen ska vara röd för alla Buttons genom att först dra en Button till scenen och sedan lägga till följande ActionScript-kod i åtgärdspanelen på bildruta 1 i tidslinjen:

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

Alla Buttons som du sedan lägger till på scenen får röda etiketter.

Ställa in en stil för alla komponenter

Du kan ställa in en stil för alla komponenter med den statiska metoden `setStyle()` i `StyleManager`-klassen.

- 1 Dra en List-komponent till scenen och ge den förekomstnamnet **aList**.
- 2 Dra en Button-komponent till scenen och ge den förekomstnamnet **aButton**.
- 3 Tryck på **F9** eller välj Åtgärder på menyn Fönster för att öppna åtgärdspanelen, om den inte är öppen, och ange följande kod i bildruta 1 i tidslinjen för att ställa in textfärgen till röd för alla komponenter:

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

- 4 Lägg till följande kod i åtgärdspanelen för att fylla i listan med text.

```
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 Välj Kontroll > Testa filmen eller tryck på Ctrl+Enter för att kompilera koden och testa innehållet. Texten på knappetiketten och i listan ska vara röd.

Om skal

En komponents utseende består av grafiska element, till exempel en kontur, en fyllningsfärg, ikoner och till och med andra komponenter. En ComboBox innehåller till exempel en List-komponent och en List-komponent innehåller en ScrollBar. Tillsammans utgör de grafiska elementen utseendet på ComboBox. Men en komponents utseende ändras beroende på dess läge. En CheckBox utan etikett ser till exempel ut ungefär så här när den visas i ett program:



En CheckBox i normalt upp-läge

Om du klickar håller ned musknappen ovanför CheckBox ändras utseendet och den ser ut så här:



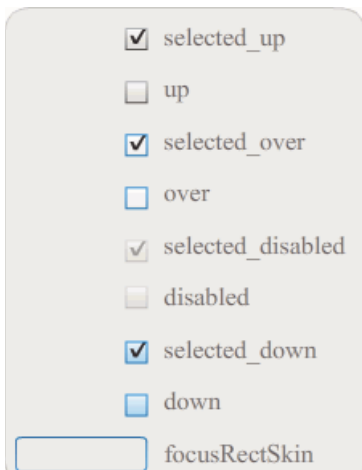
En CheckBox i ned-läge

Och när du släpper musknappen återgår CheckBox till det ursprungliga utseendet, men har en bock för att visa att den har markerats.



En CheckBox i markerat läge

De ikoner som representerar komponenten i de olika lägena kallas tillsammans för komponentens *skal*. Du kan ändra en komponents utseende i ett läge eller alla lägen genom att redigera skalerna i Flash, precis som du gör med andra Flash-symboler. Du kommer åt en komponents skal på två sätt. Det enklaste är att dra komponenten till scenen och dubbelklicka på den. Då öppnas en palett med komponentens skal, och den ser ut ungefär så här för en CheckBox.



En CheckBoxs skal

Du kan också komma åt en komponents skal individuellt från panelen Bibliotek. När du drar en komponent till scenen kopierar du den också till biblioteket, tillsammans med en mapp med komponentens resurser och alla andra komponenter som den innehåller. Om du till exempel drar en ComboBox till scenen innehåller även panelen Bibliotek komponenterna List, ScrollBar och TextInput, som är inbyggda i ComboBox, tillsammans med en mapp med skal för var och en av komponenterna, samt en mapp med delade resurser som innehåller element som de här komponenterna delar. Du kan redigera skala för alla dessa komponenter genom att öppna deras skalmappar (ComboBoxSkins, ListSkins, ScrollBarSkins och TextInputSkins) och dubbelklicka på ikonerna för skalet som du vill redigera. Om du till exempel dubbelklickar på ComboBox_downSkin öppnas skalet i symbolredigeringsläge. Se följande bild:



Skapa ett nytt skal

Om du vill skapa ett nytt utseende på en komponent i ett dokument redigerar du komponentens skal så att de ändrar utseende. Om du vill komma åt en komponents skal dubbelklickar du på komponenten på scenen, så öppnas en palett med skalerna. Sedan dubbelklickar du på det skal som du vill redigera, så öppnas det i symbolredigeringsläge. Om du till exempel dubbelklickar på TextArea-komponenten på scenen så öppnas dess resurser i symbolredigeringsläge. Ställ in zoomkontrollen på 400 %, eller mer om du vill, och redigera symbolen för att ändra dess utseende. När du är klar påverkar ändringen alla förekomster av komponenten i dokumentet. Du kan också dubbelklicka på ett visst skal i bibliotekspanelen för att öppna det på scenen i symbolredigeringsläge.

Du kan ändra komponentskal på följande sätt:

- Skapa ett nytt skal för alla förekomster
- Skapa nya skal för vissa förekomster

Skapa ett skal för alla förekomster

När du redigerar en komponents skal ändrar du som standard komponentens utseende för alla dess förekomster i dokumentet. Om du vill skapa olika utseenden för samma komponent måste du duplicera skalerna som du vill ändra och ge dem olika namn, redigera dem och sedan ställa in stilar för att använda dem. Mer information finns i ”[Skapa skal för vissa förekomster](#)” på sidan 102.

I det här kapitlet beskrivs hur du ändrar ett eller flera skal för var och en av användargränssnittskomponenterna (UI). Om följer en av de här procedurerna för att ändra ett eller flera skal för en UI-komponent ändrar du dem för alla förekomster i dokumentet.

Skapa skal för vissa förekomster

Du kan skapa ett skal för vissa förekomster av en komponent med följande allmänna procedur:

- Markera skalet i komponentens resursmapp i bibliotekspanelen.
- Duplicera skalet och tilldela det ett unikt klassnamn.
- Redigera skalet så att det får utseende som du vill ha.
- Anropa metoden `setStyle()` för komponentförekomsten för att tilldela det nya skalet skalstilen.

Följande procedur skapar ett nytt `selectedDownSkin` för en av två `Button`-förekomster.

- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra två `Buttons` från panelen Komponenter till scenen och ge dem förekomstnamnen **aButton** och **bButton**.
- 3 Öppna bibliotekspanelen och sedan mapparna `Component Assets` och `ButtonSkins` som finns i bibliotekspanelen.
- 4 Klicka på `selectedDownSkin` för att markera det.
- 5 Högerklicka så att snabbmenyn visas. Välj `Duplicera`.
- 6 I dialogrutan `Duplicera symbol` ger du skalet ett unikt namn, till exempel **Button_mySelectedDownSkin**. Sedan klickar du på `OK`.
- 7 I mappen `Library > Component Assets > ButtonSkins` i biblioteket markerar du `Button_mySelectedDownSkin` och högerklickar för att öppna snabbmenyn. Välj `Länkning` så öppnas dialogrutan `Länkningsegenskaper`.
- 8 Klicka i rutan `Exportera för ActionScript`. Låt kryssrutan `Exportera` i första bildrutan vara markerad och se till att klassnamnet är unikt. Klicka på `OK`, och klicka sedan på `OK` igen när ett varningsmeddelande visas om att ingen klassdefinition kan hittas och att en klassdefinition kommer att skapas.
- 9 Dubbelklicka på skalet `Button_mySelectedDownSkin` i panelen `Bibliotek` för att öppna det i symbolredigeringsläge.
- 10 Klicka på den blå fyllningen i mitten av skalet tills färgen visas i fyllningsfärgväljaren i egenskapsinspektören. Klicka på färgväljaren och välj färgen `#00CC00` för skalets fyllning.
- 11 Klicka på knappen `Tillbaka till vänster` om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 12 I egenskapsinspektören klickar du på fliken `Parametrar` för varje knapp och ställer in värdet `true` för växlingsparametern.
- 13 Lägg till följande kod i åtgärdspanelen i den första bildrutan på tidslinjen:

```
bButton.setStyle("selectedDownSkin", Button_mySelectedDownSkin);  
bButton.setStyle("downSkin", Button_mySelectedDownSkin);
```
- 14 Välj `Kontroll > Testa filmen`.
- 15 Klicka på varje knapp. Observera att ned-skalet (markerat och omarkerat) för objektet `bButton` använder den nya skalsymbolen.

Anpassa komponenten Button

Du kan omforma en `Button`-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget `Omforma` fritt eller något av kommandona på menyn `Ändra > Omforma`. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper i `Button`-klassen, t.ex. `height`, `width`, `scaleX` och `scaleY`.

Om du ändrar storleken på knappen förblir ikonens eller etikettens storlek densamma. Begränsningsramen för en `Button` motsvarar dess kantlinje och anger också förekomstens träffområde. Om du ökar förekomstens storlek ökar du också träffområdets storlek. Om begränsningsramen är för liten för etiketten beskärs etiketten så att den passar.

Om en `Button` har en ikon och den är större än knappen, sträcker sig ikonen utanför kantlinjerna för `Button`.

Använda format med komponenten Button

En Buttons stil anger vanligtvis värden för dess skal, ikoner, textformatering och utfyllning när komponenten ritas i sina olika lägen.

Följande procedur placerar två Buttons på scenen och ställer in värdet `true` för egenskapen `emphasized` för båda Buttons när användaren klickar på en av dem. Stilen `emphasizedSkin` för den andra Button anges till stilen `selectedOverSkin` när användaren klickar på den, så de två Buttons visar olika skal för samma läge.

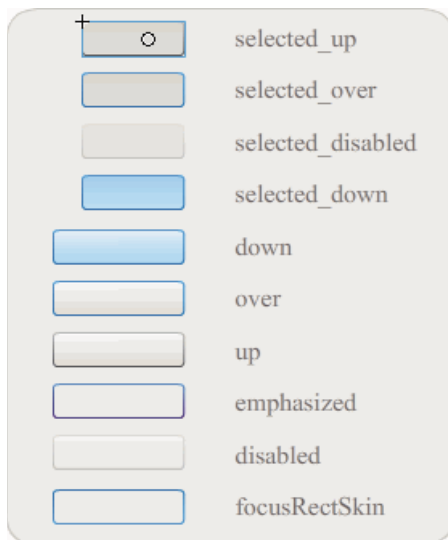
- 1 Skapa en Flash-fil (ActionScript 3.0).
- 2 Dra två Buttons till scenen, en åt gången, och ge dem förekomstnamnen **aBtn** och **bBtn**. På fliken Parametrar i egenskapsinspektören ger du dem etiketterna Button A och Button B.
- 3 Lägg till följande kod i åtgärdspanelen i den första bildrutan på tidslinjen:

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

- 4 Välj Kontroll > Testa filmen.
- 5 Klicka på en av knapparna för att se vilken effekt stilen `emphasizedSkin` har på varje knapp.

Använda skal med komponenten Button

Komponenten Button använder följande skin som motsvarar dess olika lägen. Om du vill redigera ett eller flera skal för att ändra utseendet på Button dubbelklickar du på Button-förekomsten på scenen för att öppna en palett med skal. Se bilden nedan:

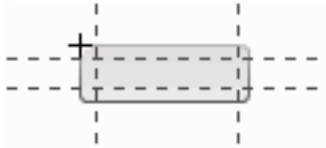


Button-skal

Om en knapp är aktiverad visas dess över-läge när du rör pekaren över den. Knappen får inmatningsfokus och visar sitt ned-läge när du klickar på den. Knappen återgår till sitt över-läge när du släpper musknappen. Om pekaren flyttas bort från knappen medan du håller ned musknappen, återgår knappen till det ursprungliga läget. Om växlingsparametern har värdet `true` visas det nedtryckta, eller klickade, läget med `selectedDownSkin`, upp-läget med `selectedUpSkin` och över-läget med `selectedOverSkin`.

Om en Button är inaktiverad visas dess inaktiverade läge, oavsett användarinteraktion.

Om du vill redigera något av skalens dubbelklickar du på det för att öppna det i symbolredigeringsläge, som på följande bild:



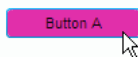
Button i symbolredigeringsläge

Nu kan du använda utvecklingsverktygen i Flash för att redigera skalet.

Med följande procedur ändras färgen på skalet `selected_over` för en Button.

- 1 Skapa en ny Flash-fil (ActionScript 3.0).
- 2 Dra en Button från panelen Komponenter till scenen. På fliken Parametrar anger du värdet `true` för växlingsparametern.
- 3 Dubbelklicka på Button för att öppna paletten med dess skal.
- 4 Dubbelklicka på skalet `selected_over` för att öppna det i symbolredigeringsläge.
- 5 Ställ in zoomkontrollen på 400% för att förstora ikonerna för redigering.
- 6 Dubbelklicka på bakgrunden tills dess färg visas i fyllningsfärgväljaren i egenskapsinspektören.
- 7 Välj färgen `#CC0099` i fyllningsfärgväljaren för att använda den på bakgrunden till skalet `selected_over`.
- 8 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 9 Välj Kontroll > Testa filmen.
- 10 Klicka på knappen för att ställa den i markerat läge.

När du rör muspekaren över knappen ska skalet `selected_over` visas på samma sätt som på följande bild.



Button som visar skalet `selected_over` med ändrad färg

Anpassa komponenten CheckBox

Du kan omforma en CheckBox-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper i klassen `CheckBox`. Du kan till exempel ändra storleken på en CheckBox genom att ställa in egenskaperna `height`, `width`, `scaleX` och `scaleY`. Om du ändrar storlek på en CheckBox förblir storleken på etiketten och kryssruteikonen densamma. Endast storleken på begränsningsramen ändras.

Begränsningsramen för en CheckBox-förekomst är osynlig och anger också förekomstens träffområde. Om du ökar förekomstens storlek ökar du också träffområdets storlek. Om begränsningsramen är för liten för etiketten beskär etiketten så att den passar.

Använda stilar med CheckBox

Du kan ställa in stilegenskaper för att ändra utseendet på en CheckBox-förekomst. Följande procedur ändrar till exempel storleken och färgen på en CheckBox-etikett.

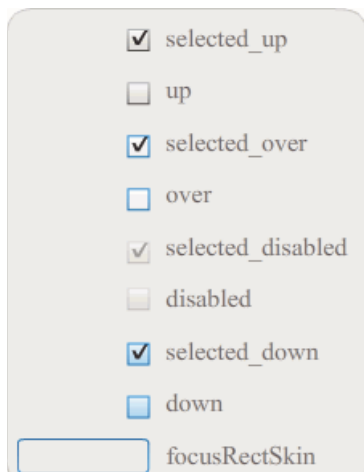
- 1 Dra en CheckBox-komponent från panelen Komponenter till scenen och ge den förekomstnamnet **myCb**.
- 2 Klicka på fliken Parametrar i egenskapsinspektören och ange följande värde för etikettparametern: **Mindre än 500 kr?**
- 3 Ange följande kod i åtgärdspanelen på bildruta 1 i huvudtidslinjen:

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

Mer information finns i ”[Ställa in stilar](#)” på sidan 98. Information om att ställa in stilegenskaper för att ändra komponentens ikoner och skal finns i ”[Skapa ett nytt skal](#)” på sidan 102 och ”[Använda skal med CheckBox](#)” på sidan 106.

Använda skal med CheckBox

Komponenten CheckBox har följande skal som du kan redigera för att ändra komponentens utseende:



CheckBox-skal

Det här exemplet ändrar komponentens kontur- och bakgrundsfärg i lägena `up` och `selectedUp`. Du utför liknande steg för att ändra skalerna för andra lägen.

- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra CheckBox-komponenten till scenen. Då placeras den också i biblioteket tillsammans med en mapp med dess resurser.
- 3 Dubbelklicka på CheckBox-komponenten på scenen för att öppna dess panel med skalikoner.
- 4 Dubbelklicka på ikonen `selected_up` för att öppna den i symbolredigeringsläge.

- 5 Ställ in zoomkontrollen på 800% för att förstora ikonerna för redigering.
- 6 Klicka på kantlinjen till CheckBox för att markera den. Använd fyllningsfärgväljaren i egenskapsinspektören för att välja färgen #0033FF och använd den på kantlinjen.
- 7 Dubbelklicka på CheckBox-bakgrunden för att markera den och använd fyllningsfärgväljaren igen för att ställa in färgen på bakgrunden till #00CCFF.
- 8 Upprepa steg 4 till 8 för upp-skalet för CheckBox.
- 9 Välj Kontroll > Testa filmen.

Anpassa komponenten ColorPicker

Den enda storleksändring du kan göra för en ColorPicker gör du via dess stilar: `swatchWidth`, `swatchHeight`, `backgroundPadding`, `textFieldWidth` och `textFieldHeight`. Om du försöker ändra storleken på ColorPicker med verktyget Omforma fritt eller via ActionScript med metoden `setSize()` eller egenskaperna `width`, `height`, `scaleX` eller `scaleY`, ignoreras dessa värden när du skapar SWF-filen och ColorPicker visas med standardstorleken. Palettens bakgrund ändrar storlek så att den matchar antalet kolumner som ställts in med `setStyle()` för stilen `columnCount`. Standardantalet kolumner är 18. Du kan ställa in anpassade färger till 1024, och paletten ändrar automatiskt storlek så att den matchar antalet färgrutor.

Använda stilar med komponenten ColorPicker

Du kan ställa in flera stilar för att ändra utseendet på komponenten ColorPicker. Följande procedur ändrar till exempel antalet kolumner (`columnCount`) i ColorPicker till tolv, ändrar höjden (`swatchHeight`) och bredden (`swatchWidth`) på färgrutorna och ändrar utfyllningen för både textfältet (`textPadding`) och bakgrunden (`backgroundPadding`).

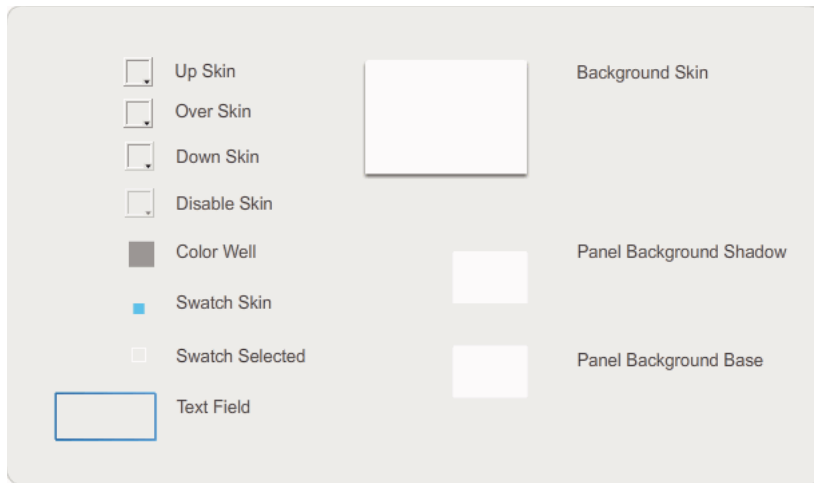
- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra ColorPicker-komponenten till scenen och ge den förekomstnamnet **aCp**.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande kod:

```
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 Välj Kontroll > Testa filmen.
- 5 Klicka på ColorPicker för att öppna den och se hur de här inställningarna har ändrat dess utseende.

Använda skal med komponenten ColorPicker

Komponenten ColorPicker använder följande skal för att representera de olika visuella lägena.

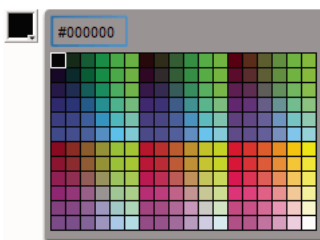


ColorPicker-skal

Du kan ändra färgen på Background-skalet om du vill ändra färgen på palettens bakgrund.

- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra ColorPicker-komponenten till scenen.
- 3 Dubbelklicka på den för att öppna paletten med skal.
- 4 Dubbelklicka på Background-skalet tills det har markerats och fyllningsfärgväljaren visas i egenskapsinspektören.
- 5 Välj färgen #999999 med fyllningsfärgväljaren för att använda den på Background-skalet.
- 6 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 7 Välj Kontroll > Testa filmen.

När du klickar på ColorPicker ska palettens bakgrund vara grå, som på följande bild.



ColorPicker med mörkgrått Background-skal

Anpassa komponenten ComboBox

Du kan omforma en ComboBox-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper i ComboBox-klassen, till exempel `height` och `width`, samt `scaleX` och `scaleY`.

ComboBox ändrar storlek så att den angivna passar bredden och höjden. Listan ändrar storlek så att den passar komponentens bredd, såvida inte egenskapen `dropdownWidth` har ställts in.

Om texten är för lång för att passa i ComboBox beskärs texten så att den passar. Du måste ändra storlek på ComboBox och ställa in egenskapen `dropdownWidth` så att texten passar in.

Använda stilar med komponenten ComboBox

Du kan ställa in stilegenskaper för att ändra utseendet på en ComboBox-komponent. Stilarna anger värden för komponentens skal, cellrenderare, utfyllning och knappbredd. Följande exempel ställer in stilarna `buttonWidth` och `textPadding`. Stilen `buttonWidth` ställer in bredden på knappens träffområde och är aktiv när ComboBox är redigerbar och du bara kan trycka på knappen för att öppna listrutan. Stilen `textPadding` anger storleken på utrymmet mellan textfältets yttre kantlinje och texten. Det är praktiskt när du vill centrera texten lodrätt i textfältet om du gör ComboBox högre. Annars kan det se ut som om texten finns högst upp i textfältet.

- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra ComboBox-komponenten till scenen och ge den förekomstnamnet **aCb**.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande kod:

```
import fl.data.DataProvider;

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

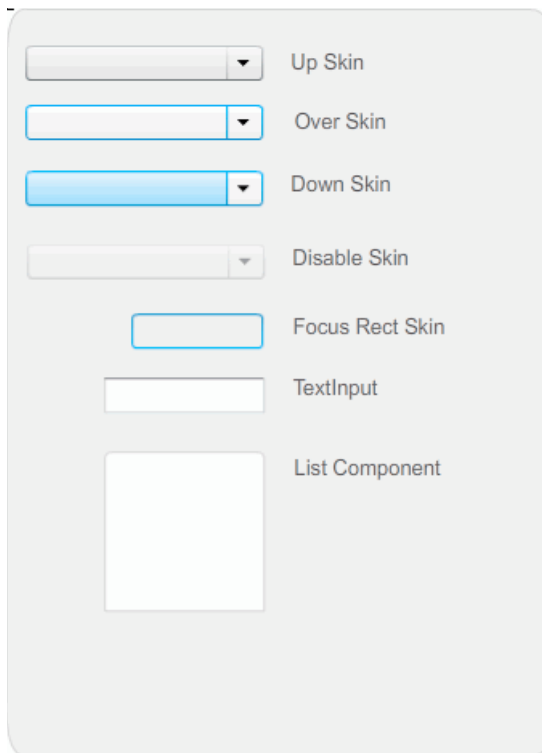
var items:Array = [
    {label:"San Francisco", data:"601 Townsend St."},
    {label:"San Jose", data:"345 Park Ave."},
    {label:"San Diego", data:"10590 West Ocean Air Drive, Suite 100"},
    {label:"Santa Rosa", data:"2235 Mercury Way, Suite 105"},
    {label:"San Luis Obispo", data:"3220 South Higuera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

- 4 Välj Kontroll > Testa filmen.

Observera att det område på knappen där du kan klicka för att öppna listan bara är ett smalt område på högra sidan. Notera också att texten är centrerad lodrätt i textfältet. Du kan försöka köra exemplet utan de två `setStyle()`-satserna för att se vilken effekt de har.

Använda skal med ComboBox

Komponenten ComboBox använder följande skal för att representera de olika visuella lägena:



ComboBox-skal

Du kan ändra färgen på ned-skalet om du vill ändra färgen på komponenten i inaktivt läge på scenen.

- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra ComboBox-komponenten till scenen.
- 3 Dubbelklicka på den för att öppna paletten med skal.
- 4 Dubbelklicka på Up-skalet tills det markeras och öppnas för redigering.
- 5 Ställ in zoomkontrollen på 400 %.
- 6 Klicka i mitten av skalet tills dess färg visas i fyllningsfärgväljaren i egenskapsinspektören.
- 7 Välj färgen #33FF99 med fyllningsfärgväljaren för att använda den på Up-skalet.
- 8 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 9 Välj Kontroll > Testa filmen.

ComboBox ska visas på scenen på samma sätt som på följande bild.



ComboBox med anpassad färg på Background-skalet

Anpassa komponenten DataGrid

Du kan omforma en DataGrid-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper, t.ex. `width`, `height`, `scaleX` och `scaleY`. Det det inte finns någon vågrät rullningslist justeras kolumnbredderna proportionerligt. Om kolumnjusteringar (och därmed celljusteringar) inträffar kan texten i cellerna beskäras.

Använda stilar med komponenten DataGrid

Du kan ställa in stilegenskaper för att ändra utseendet på en DataGrid-komponent. DataGrid-komponenten ärver stilar från List-komponenten. (Se ”[Använda format med komponenten List](#)” på sidan 116.)

Ställa in format för en enskild kolumn

Ett DataGrid-objekt kan ha flera kolumner och du kan ange olika cellåtergivare för varje kolumn. Varje kolumn i en DataGrid representeras av ett DataGridColumn-objekt och klassen DataGridColumn innehåller en `cellRenderer`-egenskap som du kan definiera kolumnens CellRenderer för.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra DataGrid-komponenten till bibliotekspanelen.
- 3 Lägg till följande kod i åtgärdspanelen i den första bildrutan på tidslinjen. Den här koden skapar en DataGrid med en lång textsträng i den tredje kolumnen. I slutet ger den kolumnens `cellRenderer`-egenskap namnet på en cellåtergivare som återger en cell med flera rader.

```
/* 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 Spara FLA-filen som MultiLineGrid.fla.
- 5 Skapa en ny ActionScript-fil.
- 6 Kopiera följande ActionScript-kod till i skriptfönstret:

```
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 Spara ActionScript-filen som MultiLineCell.as i samma mapp som du sparade MultiLineGrid.fla i.
- 8 Återgå till MultiLineGrid.fla-programmet och välj Kontroll > Testa film.
DataGrid bör se ut så här:

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 för MultiLineGrid.fla-programmet

Ställa in rubrikformat

Du ställer in textstilen för en rubrikrad genom att använda `headerTextFormat`-stilen. Följande exempel använder `TextFormat`-objektet för att ställa in `headerTextFormat`-stilen så att den använder teckensnittet Arial, röd färg, teckensnittsstorleken 14 och kursiv.

- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra DataGrid-komponenten till scenen och ge den förekomstnamnet `aDg`.
- 3 Öppna åtgärdspanelen, markera bildruta 1 i huvudtidslinjen och ange följande kod:

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

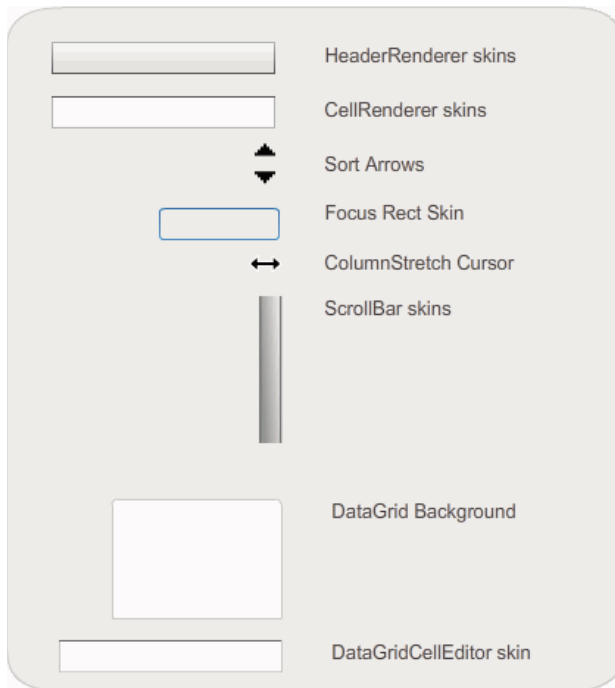
var myDP:Array = new Array();
myDP = [{firstName:"Winston", lastName:"Elstad"},
        {firstName:"Ric", lastName:"Dietrich"},
        {firstName:"Ewing", lastName:"Canepa"},
        {firstName:"Kevin", lastName:"Wade"},
        {firstName:"Kimberly", lastName:"Dietrich"},
        {firstName:"AJ", lastName:"Bilow"},
        {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 Välj Kontroll > Testa filmen för att köra programmet.

Använda skal med komponenten DataGrid

Komponenten DataGrid använder följande skal för att representera de olika visuella lägena:



DataGrid-skal

CellRenderer-skalet används för brödcellerna i DataGrid, och HeaderRenderer-skalet används för rubrikraden. Följande procedur ändrar bakgrundsfärgen på rubrikraden, men du kan använda samma procedur för att ändra bakgrundsfärgen på brödcellerna i DataGrid genom att redigera CellRenderer-skalet.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra DataGrid-komponenten till scenen och ge den förekomstnamnet **aDg**.
- 3 Dubbelklicka på komponenten för att öppna paletten med skal.
- 4 Ställ in zoomkontrollen på 400% för att förstora ikonerna för redigering.
- 5 Dubbelklicka på HeaderRenderer-skalet för att öppna paletten med HeaderRenderer-skal.
- 6 Dubbelklicka på Up_Skin för att öppna det i symbolredigeringsläge, och klicka på dess bakgrund tills den har markerats och fyllningsfärgväljaren visas i egenskapsinspektören.
- 7 Välj färgen #00CC00 med fyllningsfärgväljaren för att använda den på bakgrunden till det Up_Skin HeaderRenderer-skalet.
- 8 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 9 Lägg till följande kod i åtgärdspanelen på bildruta 1 i tidslinjen för att lägga till data i DataGrid:

Anpassa UI-komponenterna

```

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 Välj Kontroll > Testa filmen för att testa programmet.

DataGrid ska se ut som det gör i följande bild, med en grön bakgrund i rubrikraden.

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

DataGrid med anpassad bakgrund till rubrikrad

Anpassa komponenten Label

Du kan omforma en Label-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Du kan också ställa in utvecklingsparametern `autoSize`. Om du ställer in den här parametern ändras inte begränsningsramen i direktförhandsvisningen, men storleken på Label ändras. Storleken på Label ändras beroende på `wordwrap`-parametern. Om parametern är `true` ändras storleken på Label så att den passar texten. Om parametern är `false` ändras storleken på Label vågrätt. Vid körning använder du metoden `setSize()`. Mer information finns i avsnitten om metoden `Label.setSize()` och egenskapen `Label.autoSize` i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*. Se även ”[Skapa ett program med komponenten Label](#)” på sidan 62.

Använda format med komponenten Label

Du kan ställa in stilegenskaper för att ändra utseendet på en Label-förekomst. All text i en förekomst av en Label-komponent måste ha samma stil. Label-komponenten har en `textFormat`-stil som har samma attribut som `TextFormat`-objektet, och som gör att du kan ställa in samma egenskaper för innehållet i `Label.text` som för ett vanligt `TextField` i Flash. Följande exempel ställer in röd textfärg för en etikett.

- 1 Dra en Label-komponent från panelen Komponenter till scenen och ge den förekomstnamnet `a_label1`.
- 2 Klicka på fliken Parametrar och byt ut textegenskapens värden mot texten:

Färga mig röd

- 3 Markera bildruta 1 i huvudtidslinjen, öppna åtgärdspanelen och ange följande kod:

```
/* 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 Välj Kontroll > Testa filmen.

Mer information om Label-format finns i klassen Label i [Referenshandbok för ActionScript® 3.0 i Adobe Flash Professional](#).

Skal och Label

Label-komponenten har inga visuella element som skalerna kan användas på.

Anpassa komponenten List

Du kan omforma en List-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` och tillämpliga egenskaper i List-klassen, till exempel `height`, `width`, `scaleX` och `scaleY`.

När storleken på en lista ändras, krymper raderna vågrätt och eventuell text på raderna beskärs. Lodrätt så läggs rader till eller tas bort efter behov. Rullningslistor placeras automatiskt där de behövs.

Använda format med komponenten List

Du kan ställa in stilegenskaper för att ändra utseendet på en List-komponent. Stilarna anger värden för komponentens skal och utfyllning när komponenten ritas.

De olika skalstilarna gör att du kan ange olika klasser att använda för skalet. Mer information om att använda skalstilar finns i "Om skal" på sidan 101.

Följande procedur ställer in värdet på stilen `contentPadding` för List-komponenten. Observera att värdet på den här inställningen subtraheras från storleken på List för att få utfyllning runt innehållet, så du kan behöva öka storleken på List för att förhindra att texten i den beskärs.

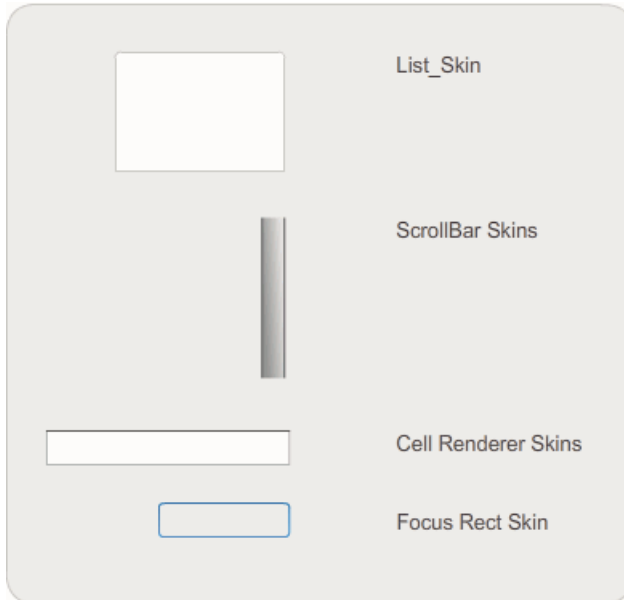
- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra en List-komponent från panelen Komponenter till scenen och ge den förekomstnamnet **aList**.
- 3 Markera bildruta 1 i huvudtidslinjen, öppna panelen Åtgärder och ange följande kod, som ställer in stilen `contentPadding` och lägger till data i List:

```
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 Välj Kontroll > Testa filmen.

Använda skal med komponenten List

Komponenten List använder följande skal för att representera de olika visuella lägena:

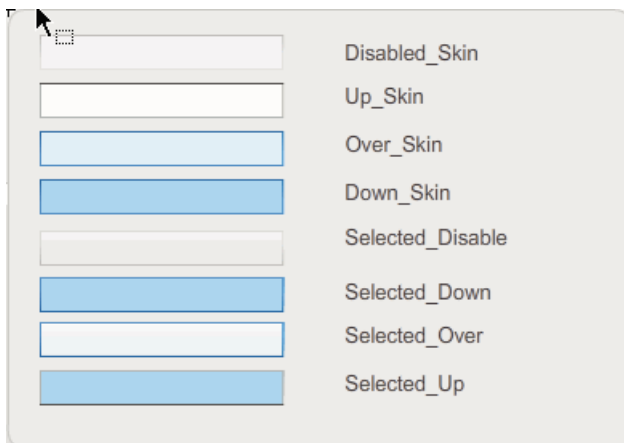


List-skal

Mer information om skalning av ScrollBar finns i ”[Anpassa komponenten UIScrollBar](#)” på sidan 131. Information om skalning av Focus Rect-skal finns i ”[Anpassa komponenten TextArea](#)” på sidan 126

Obs! Om du ändrar ScrollBar-skalet i en komponent ändras skalet för alla andra komponenter som använder ScrollBar.

Dubbelklicka på Cell Renderer-skalet för att öppna ytterligare en palett med skal för en List-cells olika lägen.



Cell Renderer-skal för List

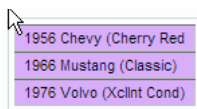
Du kan ändra utseendet på List-cellerna genom att redigera de här skalen. Följande procedur ändrar färgen på Up-skalet för att ändra utseende på List i dess normala inaktiva läge.

- 1 Skapa ett nytt Flash-fildokument (ActionScript 3.0).
- 2 Dra en List-komponent från panelen Komponenter till scenen och ge den förekomstnamnet **aList**.
- 3 Dubbelklicka på List för att öppna paletten med skal.
- 4 Dubbelklicka på Cell Renderer-skalet för att öppna paletten med cellrenderarskal.
- 5 Dubbelklicka på Up_Skin så att det öppnas för redigering.
- 6 Klicka på skalets fyllnadsområde för att markera det. En fyllningsfärgväljare ska visas i egenskapsinspektören med skalets aktuella fyllningsfärg.
- 7 Välj färgen #CC66FF med fyllningsfärgväljaren för att använda den på fyllningen för Up_Skin.
- 8 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 9 Lägg till följande kod i åtgärdspanelen på bildruta 1 i tidslinjen för att lägga till data i List:

```
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 Välj Kontroll > Testa filmen.

Listan ska visas på samma sätt som i bilden nedan:



List-celler med anpassad färg på Up_Skin

Inramningen kommer från inställningen av `contentPadding`-stilen.

Anpassa komponenten NumericStepper

Du kan omforma en NumericStepper-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper och metoder i NumericStepper-klassen, t.ex. `width`, `height`, `scaleX` och `scaleY`.

Bredden på knapparna med ned- och uppil ändras inte om du ändrar storlek på NumericStepper-komponenten. Om nummerlistan ändras så att den blir större än standardhöjden är standardbeteendet att pilknapparna fästs mot komponentens övre och nedre kant. Annars bestämmer niosegmentsskalning hur knapparna ritas. Pilknapparna visas alltid till höger om textrutan.

Stilar och komponenten NumericStepper

Du kan ställa in stilegenskaper för NumericStepper-komponenten för att ändra dess utseende. Stilarna anger värden för komponentens skal, utfyllning och textformat när komponenten ritas. Med stilen `textFormat` kan du ändra storleken och utseendet på värdet för NumericStepper. De olika skalstilarna gör att du kan ange olika klasser att använda för skalet. Mer information om att använda skalstilar finns i ”Om skal” på sidan 101.

Den här proceduren använder stilen `textFormat` för att ändra utseendet på värdet som NumericStepper visar.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra NumericStepper-komponenten från panelen Komponenter till scenen och ge den förekomstnamnet `myNs`.
- 3 Lägg till följande kod i bildruta 1 på huvudtidslinjen i panelen Åtgärder:

```
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 Välj Kontroll > Testa filmen.

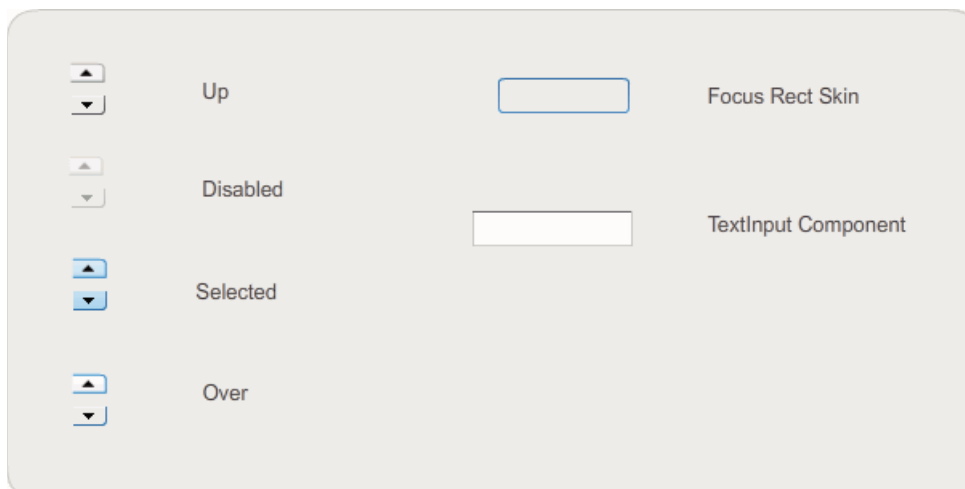
Skal och komponenten NumericStepper

Komponenten NumericStepper har skal som representerar lägena upp, ned, inaktiverad och markerad för knapparna.

Om en nummerlista är aktiverad visas ned- och upp-knapparnas över-lägen när pekaren flyttas över dem. Knapparnas ned-lägen visas när du trycker på dem. Knapparna återgår till sitt över-läge när du släpper musknappen. Om pekaren flyttas bort från knapparna medan du håller ned musknappen, återgår knapparna till det ursprungliga läget.

Om en nummerlista är inaktiverad visas dess inaktiverade läge, oavsett användarinteraktion.

En NumericStepper-komponent har följande skal:



NumericStepper-skal

- 1 Skapa en ny FLA-fil.
- 2 Dra NumericStepper-komponenten till scenen.

- 3 Ställ in zoomkontrollen på 400 % för att förstora bilden för redigering.
- 4 Dubbelklicka på bakgrunden till TextInput-skalet på panelen med skal tills du kommer ned till Group-nivå, och bakgrundsfärgen visas i fyllningsfärgväljaren i egenskapsinspektören.
- 5 Använd fyllningsfärgväljaren i egenskapsinspektören och markera färgen #9999FF för att använda den på TextInput-skalets bakgrund.
- 6 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 7 Dubbelklicka på NumericStepper för att öppna skalpanelen igen.
- 8 Dubbelklicka på bakgrunden till uppilknappen i Up-gruppen tills bakgrunden markeras och dess färg visas i fyllningsfärgväljaren i egenskapsinspektören.
- 9 Välj färgen #9966FF för att använda den på uppilknappens bakgrund.
- 10 Upprepa steg 8 och 9 för nedpilknappen i Up-gruppen.
- 11 Välj Kontroll > Testa filmen.

NumericStepper-förekomsten ska se ut som på följande bild:



Anpassa komponenten ProgressBar

Du kan omforma en ProgressBar-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper i ProgressBar-klassen, till exempel `height`, `width`, `scaleX` och `scaleY`.

ProgressBar har tre skal: ett spårskal, ett indikatorskal och ett obestämt skal. Den använder niosegmentsskalning för att skala resurserna.

Stilar och komponenten ProgressBar

Du kan ställa in stilegenskaper för att ändra utseendet på en ProgressBar-förekomst. ProgressBar-stilarna anger värden för dess skal och utfyllning när komponenten ritas. Följande exempel förstorar en ProgressBar-förekomst och ställer in dess `barPadding`-stil.

- 1 Skapa en ny FLA-fil.
- 2 Dra en ProgressBar-komponent från panelen Komponenter till scenen och ge den förekomstnamnet **myPb**.
- 3 Ange följande kod i åtgärdspanelen på bildruta 1 i huvudtidslinjen:

```
myPb.width = 300;  
myPb.height = 30;  
  
myPb.setStyle("barPadding", 3);
```

- 4 Välj Kontroll > Testa filmen.

Mer information om att ställa in skalstilar finns i ["Om skal"](#) på sidan 101.

Skal och komponenten ProgressBar

Komponenten ProgressBar använder skal för att representera förloppsindikatorns spår, den fullständiga indikatorn och en obestämd indikator, så som visas på följande bild.



ProgressBar-skal

Indikatorn placeras över spårskalet, och barPadding används för att bestämma placeringen. Resurserna skalas med niosegmentsskalning.

Den obestämda indikatorn används när ProgressBar-förekomstens `indeterminate`-egenskap har värdet `true`. Skalets storlek ändras lodrätt och vågrätt för att passa storleken på ProgressBar.

Du kan redigera de här skalerna för att ändra utseendet på ProgressBar. I följande exempel ändras till exempel färgerna på den obestämda indikatorn.

- 1 Skapa en ny FLA-fil.
- 2 Dra en ProgressBar-komponent till scenen och dubbelklicka på den för att öppna dess panel med skalikoner.
- 3 Dubbelklicka på skalet för den obestämda indikatorn.
- 4 Ställ in zoomkontrollen på 400% för att förstora ikonerna för redigering.
- 5 Dubbelklicka på ett av de diagonala fälten, och håll sedan ned Skift-tangenten och klicka på ett av de andra. Den aktuella färgen visas i fyllningsfärgväljaren i egenskapsinspektören.
- 6 Klicka på fyllningsfärgväljaren i egenskapsinspektören för att öppna den, och välj färgen #00CC00 för att använda den på de markerade diagonala fälten.
- 7 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 8 Välj Kontroll > Testa filmen.

ProgressBar ska se ut som på följande bild.



Anpassa komponenten RadioButton

Du kan omforma en RadioButton-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()`.

Begränsningsramen för en RadioButton-komponent är osynlig och anger också komponentens träffområde. Om du ökar komponentens storlek ökar du också träffområdets storlek.

Om begränsningsramen är för liten för etiketten beskärs etiketten så att den passar.

Använda format med komponenten RadioButton

Du kan ställa in stilegenskaper för att ändra utseendet på en RadioButton. Stilegenskaperna för RadioButton anger värden för dess skal, ikoner, textformatering och utfyllnad när komponenten ritas. RadioButton-stilarna anger värden för dess skal och utfyllnad för dess layout när komponenten ritas.

Följande exempel hämtar stilen `textFormat` från en CheckBox-komponent och använder den på en RadioButton för att göra stilarna på deras etiketter identiska.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en CheckBox-komponent till scenen och ge den förekomstnamnet **myCh** i egenskapsinspektören.
- 3 Dra en RadioButton-komponent till scenen och ge den förekomstnamnet **myRb** i egenskapsinspektören.
- 4 Lägg till följande kod i åtgärdspanelen i den första bildrutan på tidslinjen.

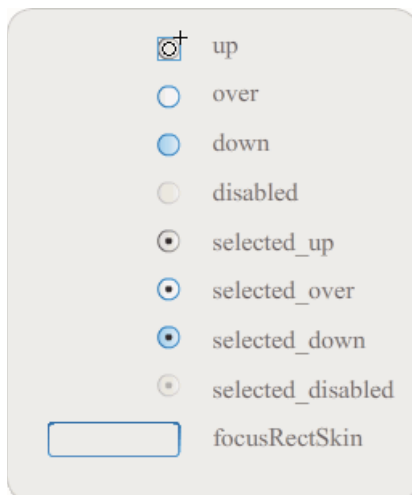
```
var tf:TextFormat = new TextFormat();  
tf.color = 0x00FF00;  
tf.font = "Georgia";  
tf.size = 18;  
myCh.setStyle("textFormat", tf);  
myRb.setStyle("textFormat", myCh.getStyle("textFormat"));
```

Den här koden ställer in `textFormat`-stilen för CheckBox, och använder den sedan på RadioButton genom att anropa metoden `getStyle()` för CheckBox.

- 5 Välj Kontroll > Testa filmen.

Skal och komponenten RadioButton

RadioButton har följande skal som du kan redigera för att ändra dess utseende:



RadioButton-skal

Om en RadioButton är aktiverad och inte markerad, visas dess över-skal när användaren rör pekaren över den. När användaren klickar på en RadioButton får den indatafokus och visar `selected_down`-skalet. När användaren släpper musknappen visar RadioButton `selected_up`-skalet. Om användaren flyttar ut pekaren från RadioButtons träffområde medan musknappen hålls ned visas RadioButtons upp-skal igen.

Om en RadioButton är inaktiverad visas dess inaktiverade läge, oavsett användarinteraktion.

Följande exempel byter ut selected_up-skalet som anger det markerade läget.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra RadioButton-komponenten till scenen och dubbelklicka på den för att öppna dess palett med skal.
- 3 Ställ in zoomkontrollen på 800% för att förstora ikonerna för redigering.
- 4 Dubbelklicka på selected_up-skalet för att markera det och tryck på Delete-tangenten för att ta bort det.
- 5 Välj rektangeln på verktygspanelen.
- 6 I egenskapsinspektören anger du röd (#FF0000) som linjefärg och svart (#000000) som fyllningsfärg.
- 7 Börja vid hårkorsen som markerar symbolens registreringspunkt (även *origo* eller *nollpunkt*), och klicka och dra pekaren för att rita en rektangel.
- 8 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 9 Välj Kontroll > Testa filmen.
- 10 Klicka på RadioButton för att markera den.

RadioButton i markerat läge ska se ut ungefär som på följande bild.



Anpassa komponenten ScrollPane

Du kan omforma en ScrollPane-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` och tillämpliga egenskaper och metoder i ScrollPane-klassen, t.ex. `height`, `width`, `scaleX` och `scaleY`.

Komponenten ScrollPane har följande grafiska egenskaper:

- Registreringspunkten (även *origo* eller *nollpunkt*) för innehållet finns i panelens övre vänstra hörn.
- När den vågräta rullningslistan stängs av visas den lodräta rullningslistan uppifrån och ned längs rullningsfönstrets högra sida. När den lodräta rullningslistan stängs av visas den vågräta rullningslistan från vänster till höger längs rullningsfönstrets nederkant. Du kan också stänga av båda rullningslistorna.
- Om rullningsfönstret är för litet kan det hända att innehållet inte visas korrekt.
- När du ändrar storlek på rullningsfönstret utökas eller minskas rullningsspåret och rullningsrutan (thumb), och deras träffytor ändrar storlek. Knapparnas storlek ändras inte.

Använda format med komponenten ScrollPane

Stilegenskaperna för ScrollPane-komponenterna anger värden för skalerna och utfyllningen för dess layout när komponenten ritas. De olika skalstilarna gör att du kan ange olika klasser att använda för komponentens skal. Mer information om att använda skalstilar finns i ”Om skal” på sidan 101.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).

- 2 Dra en ScrollPane-komponent till scenen och ge den förekomstnamnet **mySp**.
- 3 Klicka på fliken Parametrar i egenskapsinspektören och ange följande värde för parametern `source`:
<http://www.helpexamples.com/flash/images/image1.jpg>
- 4 Ange följande kod i åtgärdspanelen på bildrut 1 i huvudtidslinjen:

```
mySp.setStyle("contentPadding", 5);
```

Observera att utfyllningen används mellan komponentens kantlinje och innehållet, på utsidan av rullningslisterna.
- 5 Välj Kontroll > Testa filmen.

Skal och ScrollPane

Komponenten ScrollPane använder en kantlinje och rullningslister för rullningsresurser. Mer information om skalning av rullningslister finns i ”[Använda skal med komponenten UIScrollBar](#)” på sidan 132.

Anpassa komponenten Slider

Du kan omforma en Slider-komponent vågrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vi körning använder du metoden `setSize()` eller tillämpliga egenskaper i Slider-klassen, till exempel egenskaperna `width` och `scaleX`.

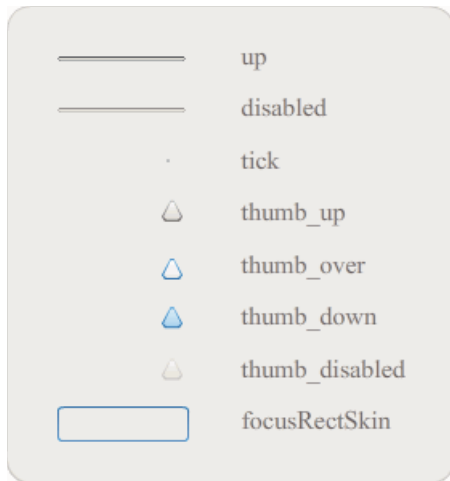
Du kan bara göra ett reglage längre. Du kan inte göra det högre. Flash ignorerar egenskapen `height` och höjdparametern för metoden `setSize()`. Men du kan skapa ett lodrätt reglage och göra det längre lodrätt.

Stilar och komponenten Slider

Slider-komponentens stilar anger bara skalens klasser och ett värde för `FocusRectPadding`, som anger antalet pixlar som ska användas för utfyllningen mellan komponentens begränsningsram och dess yttre gräns. Mer information om att använda skalstilar finns i ”[Om skal](#)” på sidan 101.

Skal och komponenten Slider

Komponenten Slider har följande skal som du kan redigera för att ändra komponentens utseende.



Slider-skal

Följande exempel redigerar upp-spåret så att färgen ändras till blå.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en Slider-komponent från panelen Komponenter till scenen.
- 3 Dubbelklicka på Slider-komponenten för att öppna dess skalpanel.
- 4 Dubbelklicka på upp-spårets registreringsmärke för att öppna det i symbolredigeringsläge.
- 5 Ställ in zoomkontrollen på 800% för att förstora ikonerna för redigering. Observera att Sliderns spår består av tre fält.
- 6 Klicka på det översta fältet för att markera det. När det är markerat visas dess färg i fyllningsfärgväljaren i egenskapsinspektören.
- 7 Använd fyllningsfärgväljaren i egenskapsinspektören och markera färgen #000066 för att använda den på Slider-spårets översta fält.
- 8 Klicka på det mittersta fältet i Slider-spåret för att markera det. När det är markerat visas dess färg i fyllningsfärgväljaren i egenskapsinspektören.
- 9 Använd fyllningsfärgväljaren i egenskapsinspektören och markera färgen #0066FF för att använda den på Slider-spårets mittersta fält.
- 10 Klicka på det nedersta fältet i Slider-spåret för att markera det. När det är markerat visas dess färg i fyllningsfärgväljaren i egenskapsinspektören.
- 11 Använd fyllningsfärgväljaren i egenskapsinspektören och markera färgen #00CCFF för att använda den på Slider-spårets nedersta fält.
- 12 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 13 Välj Kontroll > Testa filmen.

Slider ska se ut på samma sätt som i bilden nedan.



Anpassa komponenten TextArea

Du kan omforma en TextArea-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper i TextArea-klassen, till exempel `height`, `width`, `scaleX` och `scaleY`.

När du ändrar storlek på en TextArea-komponent, storleksändras kantlinjen till den nya begränsningsramen. Rullningslistor placeras längs nederkanten och högerkanten om de behövs. Sedan ändras textområdets storlek inuti det återstående området. Det finns inga element med en fast storlek i en TextArea-komponent. Om TextArea-komponentens bredd är för smal för att visa textens storlek beskärs texten.

Stilar och komponenten TextArea

TextArea-komponentens stilar anger värden för skalerna, utfyllningen och textformatet när komponenten ritas. Stilarna `textFormat` och `disabledTextFormat` styr stilen på texten som TextArea visar. Mer information om egenskaper för skalstilar finns i ”[Använda skal med komponenten TextArea](#)” på sidan 126.

Följande exempel ställer in stilen `disabledTextFormat` så att den ändrar utseendet på texten när TextArea inaktiveras, men samma process används för att ställa in stilen `textFormat` för en aktiverad TextArea.

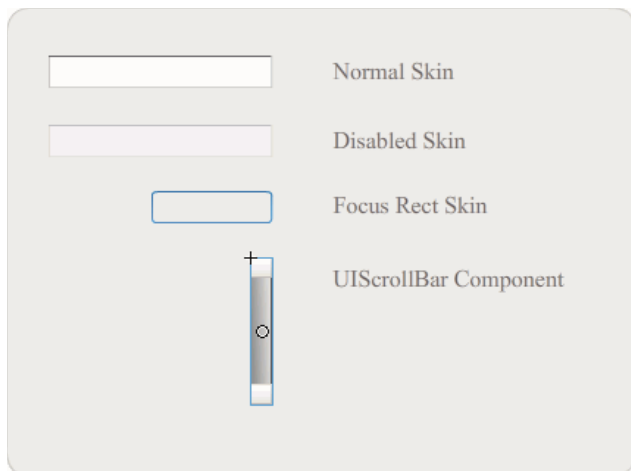
- 1 Skapa en ny Flash-fil.
- 2 Dra en TextArea-komponent till scenen och ge den förekomstnamnet **myTa**.
- 3 Lägg till följande kod i åtgärdspanelen i bildruta 1 på huvudtidslinjen.

```
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 Välj Kontroll > Testa filmen.

Använda skal med komponenten TextArea

Komponenten TextArea har följande skal som du kan redigera för att ändra komponentens utseende.



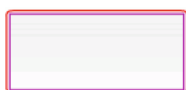
TextArea-skal

Obs! Om du ändrar ScrollBar-skalet i en komponent ändras skalet för alla andra komponenter som använder ScrollBar.

Följande procedur ändrar kantlinjefärgerna för Focus Rect-skalet, som visas när TextArea har fokus, samt Normal-skalet.

- 1 Skapa en ny Flash-fil.
- 2 Dra en TextArea-komponent till scenen och dubbelklicka på den för att öppna dess panel med skalikoner.
- 3 Dubbelklicka på Focus Rect-skalet.
- 4 Klicka på Focus Rect-skalets kantlinje för att markera det. När det är markerat visas dess aktuella färg i fyllningsfärgväljaren i egenskapsinspektören.
- 5 Klicka på fyllningsfärgväljaren i egenskapsinspektören för att öppna den, och välj färgen #CC0000 för att använda den på kantlinjen.
- 6 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 7 Dubbelklicka på TextArea-komponenten för att öppna dess panel med skalikoner.
- 8 Dubbelklicka på Normal-skalet.
- 9 Markera alla sidor i Normal-skalets kantlinje, en i taget, och ställ in färgen #990099 för dem.
- 10 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 11 Välj Kontroll > Testa filmen.

När du markerar TextArea för att börja ange text ska kantlinjen visas på samma sätt som på följande bild:



Den yttre kantlinjen är Focus Rect-skalet och den inre kantlinjen är Normal-skalets kantlinje.

Information om hur du redigerar UIScrollBar-skalet finns i ”[Anpassa komponenten UIScrollBar](#)” på sidan 131.

Anpassa komponenten TextInput

Du kan ändra storleken på en TextInput-förekomst medan du utvecklar och under körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper för klassen TextInput, till exempel `height`, `width`, `scaleX` och `scaleY`.

När du ändrar storlek på en TextInput-komponent, storleksändras kantlinjen till den nya begränsningsramen. Komponenter TextInput använder inte rullningslistor, men insättningspunkten rullar automatiskt när användaren interagerar med texten. Sedan ändras textfältets storlek inuti det återstående området. Det finns inga element med en fast storlek i en TextInput-komponent. Om TextInput-komponenten är för liten för att visa texten beskärs texten.

Stilar och komponenten TextInput

TextInput-komponentens stilar anger värden för skalerna, utfyllningen och textformateringen när komponenten ritas. Stilarna `textFormat` och `disabledTextFormat` styr stilen på texten som visas i komponenten. Mer information om egenskaper för skalstilar finns i "[Skal och komponenten TextInput](#)" på sidan 128.

Följande exempel ställer in stilen `textFormat` för att ställa in teckensnittet, storleken och färgen på texten som visas i TextInput-komponenten. Du använder samma process för att ställa in stilen `disabledTextFormat` som används när komponenten är inaktiverad.

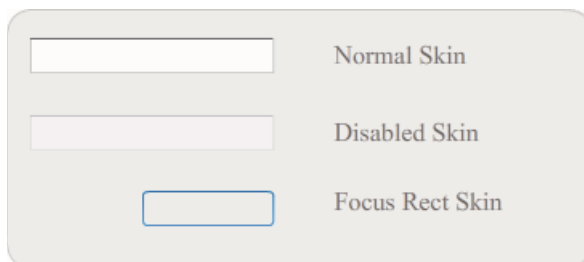
- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra en TextInput-komponent till scenen och ge den förekomstnamnet **myTi**.
- 3 Lägg till följande kod i åtgärdspanelen i bildruta 1 på huvudtidslinjen.

```
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 Välj Kontroll > Testa filmen.

Skal och komponenten TextInput

Komponenten TextInput har följande skal som du kan redigera för att ändra komponentens utseende:



TextInput-bildtext

Följande procedur ändrar kantlinjen och bakgrundsfärgerna på en TextInput-komponent:

- 1 Skapa en ny Flash-fil.
- 2 Dra en TextInput-komponent till scenen och dubbelklicka på den för att öppna dess panel med skal.
- 3 Dubbelklicka på Normal-skalet.
- 4 Ställ in zoomkontrollen på 800% för att förstora ikonerna för redigering.
- 5 Markera alla sidor i Normal-skalets kantlinje, en i taget, och ställ in färgen #993399 för att använda den.
- 6 Dubbelklicka på bakgrunden tills dess färg visas i fyllningsfärgväljaren i egenskapsinspektören. Välj färgen #99CCCC för att använda den på bakgrunden.
- 7 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen för att återgå till dokumentredigeringsläge.
- 8 Välj Kontroll > Testa filmen.

TextInput-komponenten ska se ut som på följande bild:



Anpassa komponenten TileList

Du kan omforma en TileList-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper, t.ex. `width`, `height`, `columnCount`, `rowCount`, `scaleX` och `scaleY`. Den ScrollBar, som TileList innehåller, storleksändras med listrutan.

Stilar och komponenten TileList

TileList-komponentens stilar anger värden för skalerna, utfyllningen och textformateringen när komponenten ritas. Stilarna `textFormat` och `disabledTextFormat` styr stilen på texten som visas i komponenten. Mer information om egenskaper för skalstilar finns i ”[Använda skal med komponenten TileList](#)” på sidan 130

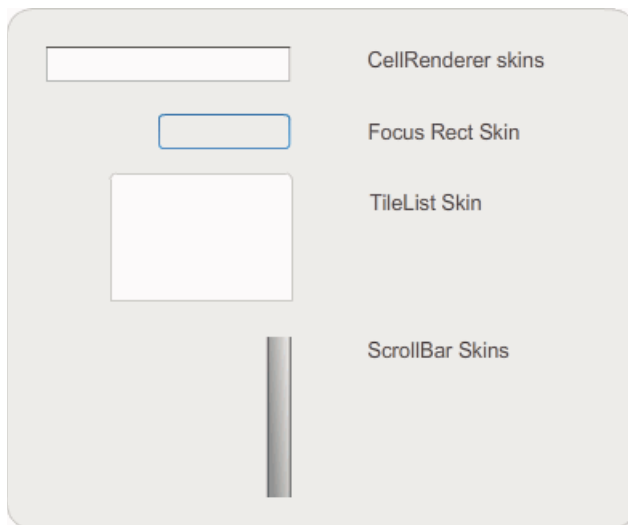
Följande exempel anropar metoden `setRendererStyle()` med hjälp av stilen `textFormat` för att ställa in attributen för teckensnitt, storlek, färg och text för etiketterna som visas i en TileList-förekomst. Samma process används för att ställa in stilen `disabledTextFormat` som används när egenskapen `enabled` har värdet `false`.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra TileList-komponenten till scenen och ge den förekomstnamnet **myTL**.
- 3 Lägg till följande kod i åtgärdspanelen i den första bildrutan på tidslinjen.

```
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.setStyle("textFormat", tf);
```

Använda skal med komponenten TileList

Komponenten TileList har ett TileList-skal, ett CellRenderer-skal och ett ScrollBar-skal. Du kan redigera de här skalerna för att ändra utseendet på TileList:



TileList-skal

Obs! Om du ändrar ScrollBar-skalet i en komponent ändras skalet i alla andra komponenter som använder ScrollBar.

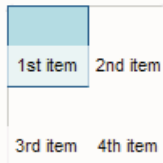
Med följande procedur ändras färgen på TileLists skal CellRenderer Selected_Up.

- 1 Skapa ett Flash-dokument (ActionScript 3.0).
- 2 Dra TileList-komponenten till scenen och dubbelklicka på den för att öppna dess panel med skal.
- 3 Dubbelklicka på CellRenderer-skalet, och dubbelklicka sedan på Selected_Up-skalet. Till slut klickar du på den rektangulära bakgrunden.
- 4 Välj färgen #99FFFF med fyllningsfärgväljaren i egenskapsinspektören för att använda den på Selected_Up-skalet.
- 5 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen tills du återgår till dokumentredigeringsläge.

Anpassa UI-komponenterna

- 6 På fliken Parametrar i egenskapsinspektören dubbelklickar du på den andra kolumnen i dataProvider-raden för att öppna dialogrutan Värden. Lägg till objekt med följande etiketter: Första objektet, Andra objektet, Tredje objektet, Fjärde objektet.
- 7 Välj Kontroll > Testa filmen.
- 8 Klicka på en av cellerna i TileList för att markera den, och flytta sedan bort musen från den markerade cellen.

Den markerade cellen ska se ut på samma sätt som i bilden nedan:



TileList-komponent med ändrad färg för Selected_Up-skalet

Anpassa komponenten UILoader

Du kan omforma en UILoader-komponent vågrätt och lodrätt under utvecklingen och vid körning. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper som `width`, `height`, `scaleX` och `scaleY`.

UILoader-komponentens beteende vid storleksändring styrs av egenskapen `scaleContent`. När värdet på `scaleContent` är `true` skalas innehållet så att det passar inom gränserna för UILoader (och skalas om igen när `setSize()` anropas). När värdet på `scaleContent` är `false` är komponentens storlek fast och lika stor som innehållets och `setSize()`, och egenskaperna för storleksändring har ingen effekt.

Komponenten UILoader har inget användargränssnittselement som du kan använda stilar och skal på.

Anpassa komponenten UIScrollBar

Du kan omforma en UIScrollBar-komponent vågrätt och lodrätt under utvecklingen och vid körning. Men för en lodrät UIScrollBar kan du inte ändra bredden, och för en vågrät UIScrollBar kan du inte ändra höjden. Under utvecklingen markerar du komponenten på scenen och använder verktyget Omforma fritt eller något av kommandona på menyn Ändra > Omforma. Vid körning använder du metoden `setSize()` eller tillämpliga egenskaper för UIScrollBar-klassen, till exempel `width`, `height`, `scaleX` och `scaleY`.

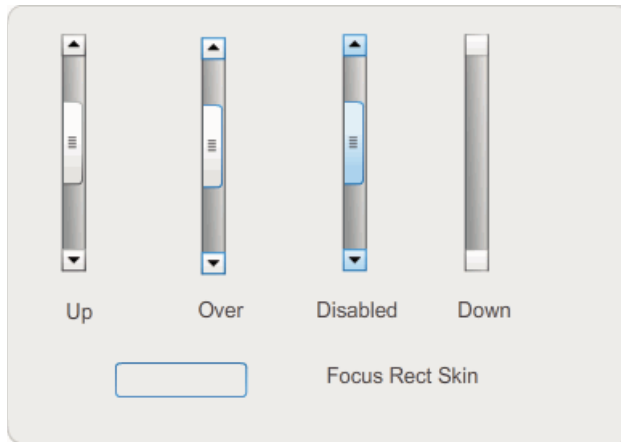
Obs! Om du använder metoden `setSize()` kan du bara ändra bredden på en vågrät rullningslist eller höjden på en lodrät rullningslist. Under utvecklingen kan du ställa in höjden på en vågrät rullningslist och bredden på en lodrät rullningslist, men värdena återställs när filmen publiceras. Endast den dimension för en rullningslist som motsvarar dess längd kan ändras.

Använda format med komponenten UIScrollBar

UIScrollBar-komponentens stilar anger bara skalens klasser och ett värde för `FocusRectPadding`, som anger antalet pixlar som ska användas för utfyllningen mellan komponentens begränsningsram och dess yttre gräns. Mer information om att använda skalstilar finns i "Om skal" på sidan 101.

Använda skal med komponenten UIScrollBar

Komponenten UIScrollBar använder följande skal.



UIScrollBar-skal

Både vågräta och lodräta rullningslistor använder samma skal. När en vågrät rullningslist visas roterar UIScrollBar-komponenten skalet på det sätt som behövs.

Obs! Om du ändrar ScrollBar-skalet i en komponent ändras skalet i alla andra komponenter som använder ScrollBar.

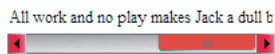
I följande exempel visas hur du ändrar färgen på UIScrollBars reglage och pilknappar.

- 1 Skapa ett nytt Flash-dokument (ActionScript 3.0).
- 2 Dra UIScrollBar-komponenten till scenen och ge den förekomstnamnet **mySb**. På fliken Parametrar ställer du in riktningen till vågrät.
- 3 Dubbelklicka på rullningslistan för att öppna panelen med skal.
- 4 Klicka på Up-skalet för att markera det.
- 5 Ställ in zoomkontrollen på 400% för att förstora ikonerna för redigering.
- 6 Dubbelklicka på bakgrunden till den högra pilen (eller uppilen för en lodrät rullningslist) tills bakgrunden har markerats och färgen visas i fyllningsfärgväljaren i egenskapsinspektören.
- 7 Välj färgen #CC0033 för att använda den på knappbakgrunden.
- 8 Klicka på knappen Tillbaka till vänster om redigeringsfältet ovanför scenen tills du återgår till dokumentredigeringsläge.
- 9 Upprepa steg 6, 7 och 8 för reglaget och den vänstra pilen (eller nedpilen för en lodrät rullningslist).
- 10 Lägg till följande kod på åtgärdspanelen i bildruta 1 i tidslinjen för att koppla rullningslistan till ett TextField.


```
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 Välj Kontroll > Testa filmen.

Komponenten UIScrollBar ska se ut på samma sätt som i bilden nedan.



Vågrät ScrollBar med röd färg på reglage och vänster- och högerpilen

Kapitel 6: Använda komponenten FLVPlayback

Med komponenten FLVPlayback kan du enkelt lägga till en videospelare i Adobe Flash CS5 Professional-programmet och spela videofiler som laddats ned progressivt över HTTP eller spela upp direktuppspelade videofiler från Adobe Macromedia Flash Media Server eller Flash Video Streaming Service (FVSS).

I Adobe Flash Player 9 uppdatering 3 (version 9.0.115.0 eller senare) har avsevärda förbättringar gjorts för uppspelning av videofiler i Flash Player. Den här uppdateringen inkluderar ändringar av FLVPlayback-komponenten som drar nytta av slutanvändarens videomaskinvara för att kunna ge högre uppselningsprestanda. Ändringarna i FLVPlayback-komponenten kan även förbättra återgivningen av de videofiler som visas i helskrämsläge.

Med Flash Player 9 uppdatering 3 förbättras funktionerna i FLVPlayback-komponenten genom att stöd har inkluderats för de MPEG-4-videoformat i HD som använder branschstandardkodningen H.264. Det gäller formaten MP4, M4A, MOV, MP4V, 3GP och 3G2.

Obs! Skyddade MP4-filer, t.ex. de som hämtas från Apple® iTunes® eller krypteras digitalt av FairPlay®, stöds inte.

Den lättanvända FLVPlayback-komponenten har följande egenskaper och fördelar:

- Kan dras till scenen och implementeras snabbt och utan problem
- Stöder helskrämsformat
- Har en uppsättning med redan utformade *skal* som gör att du kan anpassa utseende på uppselningskontrollerna
- Gör att du kan välja färg och alfavärden för redan utformade skal
- Gör att avancerade användare kan skapa egna skal
- Har direktförhandsvisning under utveckling
- Har layoutegenskaper som håller videofilen centrerad när storleken ändras
- Gör att uppspelningen kan startas när tillräckligt stor del av en progressivt nedladdad videofil har laddats ned
- Har referenspunkter som gör att du kan synkronisera video med text, grafik och animering
- Upprätthåller en SWF-fil med hanterlig storlek

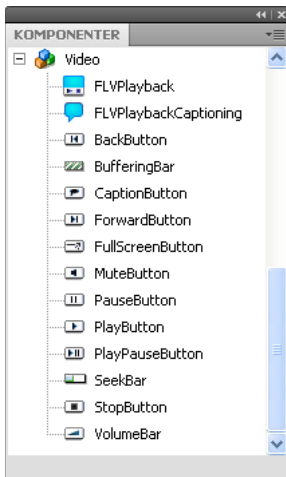
Använda FLVPlayback-komponenten

Det du gör med komponenten FLVPlayback är att placera den på scenen och ange en videofil som den ska spela upp. Dessutom kan du ställa in olika parametrar som styr dess beteende och beskriver videofilen.

Komponenten FLVPlayback innehåller också ett ActionScript-API (Application Programming Interface). Programmeringsgränssnittet innehåller följande klasser, som beskrivs i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#): CuePointType, FLVPlayback, FLVPlaybackCaptioning, NCManager, NCManagerNative, VideoAlign, VideoError, VideoPlayer, VideoState samt flera händelseklasser – AutoLayoutEvent, LayoutEvent, MetadataEvent, SkinErrorEvent, SoundEvent, VideoEvent och VideoProgressEvent.

Använda komponenten FLVPlayback

Komponenten FLVPlayback innehåller anpassade användargränssnittskomponenter för FLV-uppspelning. Komponenter FLVPlayback är en kombination av visningsområdet, eller videospelaren, där du kan visa videofilen och de kontroller som du använder för att styra den. De anpassade användargränssnittskomponenterna för FLV-uppspelning har kontrollknappar och mekanismer som du använder för att spela upp, stoppa, pausa och på andra sätt styra videofilen. Kontrollerna är bland andra BackButton, BufferingBar, CaptionButton (för FLVPlaybackCaptioning), ForwardButton, FullScreenButton, MuteButton, PauseButton, PlayButton, PlayPauseButton, SeekBar, StopButton och VolumeBar. Komponenter FLVPlayback och de anpassade användargränssnittskontrollerna för FLV-uppspelning visas i panelen Komponenter. Se bilden nedan:



FLVPlayback-komponenter i panelen Komponenter

Processen att lägga till uppspelningskontroller i FLVPlayback-komponenten kallas *skalning*. Komponenter FLVPlayback har ett initialt standardskal, SkinOverAll.swf, som innehåller kontrollerna play, stop, back, forward, seekbar, mute, volume, full screen och captioning. Om du vill ändra skalet kan du göra följande:

- Välja i en uppsättning med redan utformade skal
- Skapa ett anpassat skal och lägga till det i uppsättningen med redan utformade skal
- Välja enskilda kontroller bland de anpassade användargränssnittskomponenterna för FLV-uppspelning och anpassa dem

När du väljer ett redan utformat skal kan du välja skalfärgen och alfavärdena separat, antingen vid utvecklingen eller vid körning. Mer information finns i ”[Välja ett tidigare utformat skal](#)” på sidan 152.

När du väljer ett annat skal blir det valda skalet det nya standardskalet.

Mer information om hur du väljer eller skapar ett skal för komponenten FLVPlayback finns i ”[Anpassa komponenten FLVPlayback](#)” på sidan 152.

Skapa ett program med komponenten FLVPlayback

Du tar du med komponenten FLVPlayback i programmet på något av följande sätt:

- Dra komponenten FLVPlayback från panelen Komponenter till scenen, och ange ett värde för parametern `source`.
- Använd guiden Importera video för att skapa komponenten på scenen, och anpassa den genom att välja ett skal.
- Använd konstruktorn `FLVPlayback()` för att dynamiskt skapa en FLVPlayback-instans på scenen, under förutsättning att komponenten finns i biblioteket.

Använda komponenten FLVPlayback

***Obs!** Om du skapar en FLVPlayback-instans med ActionScript måste du också tilldela den ett skal genom att ställa in egenskapen `skin` med ActionScript. När du använder ett skal på det här sättet publiceras det inte automatiskt med SWF-filen. Du måste kopiera både programmets SWF-fil och skalets SWF-fil till din programserver, annars är inte skalets SWF-fil tillgänglig när du kör programmet.*

Dra komponenten FLVPlayback från panelen Komponenter

- 1 I panelen Komponenter klickar du på plusknappen (+) för att öppna videoposten.
 - 2 Dra FLVPlayback-komponenten till scenen.
 - 3 Markera FLVPlayback-komponenten på scenen, leta rätt på cellen Värde för parametern `source` på fliken Parametrar i komponentinspektören, och ange en sträng som specificerar följande:
 - En lokal sökväg till en videofil
 - En URL till en videofil
 - En URL till en SMIL-fil (Synchronized Multimedia Integration Language) som beskriver hur du spelar upp en videofil

Information om hur du skapar en SMIL-fil som beskriver en eller flera FLV-filer finns i ”[Använda en SMIL-fil](#)” på sidan 162.
 - 4 På fliken Parametrar i komponentinspektören, med FLVPlayback-komponenten markerad på scenen, klickar du i cellen Värde för `skin`-parametern.
 - 5 Klicka på förstoringsglaset för att öppna dialogrutan Välj skal.
 - 6 Välj något av följande alternativ:
 - I listrutan Skal väljer du ett av de tidigare utformade skalen för att koppla en uppsättning med uppspelningskontroller till komponenten.
 - Om du har skapat ett anpassat skal väljer du Anpassad skal-URL på popup-menyn. Sedan anger du i URL-textrutorna en URL för SWF-filen som innehåller skalet.
 - Välj Inget och dra enskilda anpassade användargränssnittskomponenter för FLV-uppspelning till scenen för att lägga till uppspelningskontroller.

***Obs!** I de första två fallen visas en förhandsvisning av skalen i visningspanelen ovanför popup-menyn. Du kan använda färgväljaren om du vill ändra färgen på skalet.*

Om du vill ändra färgen på en anpassad användargränssnittskontroll måste du anpassa den. Mer information om att använda anpassade användargränssnittskontroller finns i ”[Skala anpassade användargränssnittskomponenter för FLV-uppspelning individuellt](#)” på sidan 153.
 - 7 Klicka på OK för att stänga dialogrutan Välj skal.
 - 8 Välj Kontroll > Testa film för att köra SWF-filen och starta videon.
- Följande procedur använder guiden Importera video för att lägga till en FLVPlayback-komponent:

Använda guiden Importera video:

- 1 Välj Arkiv > Importera > Importera video.
- 2 Ange videofilens plats genom att välja något av följande alternativ:
 - På datorn
 - Redan distribuerad till en webbserver, Flash Video Streaming Service eller Flash Media Server
- 3 Beroende på vad du valt anger du sökvägen eller URL:en till videofilen och klickar sedan på Nästa.

4 Om du väljer en filsökväg visas dialogrutan Distribution där du kan välja ett alternativ för att ange hur du vill distribuera videon:

- Progressiv nedladdning från en webbserver
- Direktuppspela från Flash Video Streaming Service
- Direktuppspela från Flash Media Server
- Bädda in video i SWF och spela upp i tidslinje

Viktigt! Välj inte alternativet Bädda in video. Komponenten FLVPlayback spelar bara upp extern direktuppspelad video. Med det här alternativet placeras ingen FLVPlayback-komponent på scenen.

5 Klicka på Nästa.

6 Välj något av följande alternativ:

- I listrutan Skal väljer du ett av de tidigare utformade skalerna för att koppla en uppsättning med uppselningskontroller till komponenten.
- Om du har skapat ett anpassat skal för komponenten väljer du Anpassad skal-URL på popup-menyn. Sedan anger du i URL-textrutorna en URL för SWF-filen som innehåller skalet.
- Välj Inget och dra enskilda anpassade användargränssnittskomponenter för FLV-uppspelning till scenen för att lägga till uppselningskontroller.

Obs! I de första två fallen visas en förhandsvisning av skalen i visningspanelen ovanför popup-menyn.

7 Klicka på OK för att stänga dialogrutan Välj skal.

8 Läs vad som står i dialogrutan Slutför videoimport för att se vad som händer sedan, och klicka sedan på Slutför.

9 Om du inte sparade FLA-filen visas dialogrutan Spara som.

10 Välj Kontroll > Testa filmen om du vill köra SWF-filen och starta videon.

Följande procedur lägger till komponenten FLVPlayback med ActionScript.

Skapa en instans dynamiskt med ActionScript:

1 Dra FLVPlayback-komponenten från panelen Komponenter till bibliotekspanelen (Fönster > Bibliotek).

2 Lägg till följande kod i åtgärdspanelen i den första bildrutan på tidslinjen. Ändra *install_drive* till den enhet som du installerade Flash på, och ändra sedan sökvägen till sökvägen för mappen Skal i din installation:

På en Windows-dator:

```
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";
```

På en Macintosh-dator:

```
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";
```

Obs! Om du inte ser egenskaperna `source` och `skin` verkar det genererade filmklippet vara tomt.

3 Välj Kontroll > Testa film för att köra SWF-filen och starta videofilen.

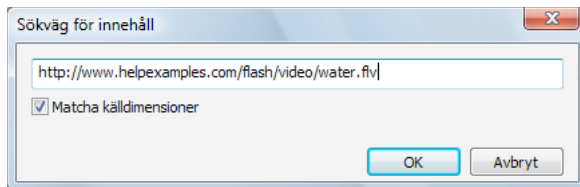
Parametrar för komponenten FLVPlayback

För varje instans av komponenten FLVPlayback kan du ställa in följande parametrar i komponentinspektören eller egenskapsinspektören: `align`, `autoplay`, `cuePoints`, `preview`, `scaleMode`, `skin`, `skinAutoHide`, `skinBackgroundAlpha`, `skinBackgroundColor`, `source` och `volume`. Var och en av de här parametrarna har en motsvarande ActionScript-egenskap med samma namn. När du tilldelar ett värde till parametrarna anger du egenskapens startläge i programmet. Om du anger egenskapen i ActionScript åsidosätter det värdet som du anger i parametern. Information om möjliga värden på de här parametrarna finns i avsnittet om klassen FLVPlayback i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Ange FLVPlayback-parametern source

Med parametern `source` anger du videofilens namn och plats. Båda behövs för att ange för Flash hur filen ska spelas upp.

Öppna dialogrutan Sökväg för innehåll genom att dubbelklicka på cellen Värde för `source`-parametern i komponentinspektören.



Dialogrutan Sökväg för innehåll för FLVPlayback

Dialogrutan Innehållssökväg innehåller en kryssruta, Matcha källans FLV-dimensioner, som anger om FLVPlayback-instansen på scenen ska matcha videokällfilens mått. Videokällfilen innehåller de önskade höjd- och breddmått för uppspelning. Om du markerar det här alternativet ändras FLVPlayback-instansens mått så att de matchar de önskade måtten.

Källan

Ange en URL eller lokal sökväg för videofilen eller en XML-fil som beskriver hur videofilen ska spelas upp. Om du inte känner till videofilens exakta plats klickar du på mappikonen för att öppna en dialogruta där du kan bläddra till rätt plats. När du bläddrar fram en videofil, som finns på samma plats eller under den plats där mål-SWF-filen finns, anger Flash automatiskt sökvägen i relation till den platsen så att du kan använda den från en webserver. Annars är sökvägen en absolut Windows- eller Macintosh-sökväg. Om du vill ange namnet på en lokal XML-fil skriver du sökvägen och namnet.

Om du anger en HTTP-URL spelas videofilen upp som en progressiv nedladdning. Om du anger en URL som är en RTMP-URL direktuppspelas videofilen från Flash Media Server eller FVSS. En URL till en XML-fil kan också vara en direktuppspelad videofil från Flash Media Server eller FVSS.

Viktigt!

Du kan också ange platsen för en SMIL-fil som beskriver hur du direktuppspelar flera videofiler för olika bandbredder. Filen använder SMIL (Synchronized Multimedia Integration Language) för att beskriva FLV-filerna. En beskrivning av SMIL-filen finns i ”[Använda en SMIL-fil](#)” på sidan 162.

Du kan även ange namnet och platsen för videofilen med hjälp av ActionScript `FLVPlayback.source`-egenskapen och metoderna `FLVPlayback.play()` och `FLVPlayback.load()`. Det här tre alternativen åsidosätter parametern `source` i komponentinspektören. Mer information finns i avsnitten om `FLVPlayback.source`, `FLVPlayback.play()` och `FLVPlayback.load()` för klassen `FLVPlayback` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Helskärmstödet

ActionScript 3.0-versionen av `FLVPlayback`-komponenten stöder helskärmsläge, vilket kräver Flash Player 9.0.28.0 eller senare och att även HTML har ställts in korrekt för helskärmvisning. Vissa redan utformade skal har en alternativknapp som används för att växla mellan helskärm på och helskärm av. Ikonen `FullScreenButton` visas på höger sida om kontrollfältet på följande bild.



Helskärmssikon i kontrollfältet

Stöd för helskärm fungerar bara om egenskapen `fullScreenTakeOver` är inställd på `true` (vilket den är som standard).

Helskärmstödet kan förekomma med eller utan stöd för maskinvaruacceleration. Information om stöd för maskinvaruacceleration finns i ”[Maskinvaruacceleration](#)” på sidan 142.

Så här implementerar du helskärmstödet för `FLVPlayback`:

- 1 Lägg till `FLVPlayback`-komponenten i ditt program och tilldela den en videofil.
- 2 Välj ett skal för den `FLVPlayback`-komponent som innehåller helskärmknappen (exempelvis `SkinUnderPlaySeekFullscreen.swf`) eller lägg till användargränssnittskomponenten `FullScreenButton` till `FLVPlayback`-komponenten från avsnittet Video i panelen Komponenter.
- 3 Välj Arkiv > Publiceringsinställningar.
- 4 I dialogrutan Publiceringsinställningar klickar du på fliken HTML.
- 5 På fliken HTML väljer du Flash med helskärmstödet från popup-menyn Mall.
- 6 På fliken HTML markerar du även kryssrutan Identifiera Flash-version och anger version 9.0.28 eller senare, beroende på vilken version av Flash Player som du använder.
- 7 Välj fliken Format och se till att både alternativet Flash (.swf) och alternativet HTML (.html) är markerade. Du kan ersätta standardfilnamnen.
- 8 Klicka på Publicera och sedan på OK.

Som ett alternativ till steg 7 kan du klicka på OK och sedan välja Arkiv > Publicera förhandsgranskning > Standard- (HTML) för att automatiskt öppna den exporterade HTML-filen i din standardwebbläsare. Annars öppnar du den exporterade HTML-filen i din webbläsare för att testa helskärmalternativet.

Om du vill lägga till `FLVPlayback`-komponenten med helskärmstödet på din webbsida öppnar du den exporterade HTML-filen och kopierar den kod som bäddar in SWF-filen i HTML-filen på din webbsida. Koden bör nu se ut ungefär så här:

```
//from the <head> section

<script language="javascript"> AC_FL_RunContent = 0; </script>
<script language="javascript"> DetectFlashVer = 0; </script>
<script src="AC_RunActiveContent.js" language="javascript"></script>
<script language="JavaScript" type="text/javascript">
<!--
// -----
// Globals
// Major version of Flash required
var requiredMajorVersion = 9;
// Minor version of Flash required
var requiredMinorVersion = 0;
// Revision of Flash required
var requiredRevision = 28;
// -----
// -->
</script>

//and from the <body> section

<script language="JavaScript" type="text/javascript">
<!--
if (AC_FL_RunContent == 0 || DetectFlashVer == 0) {
    alert("This page requires AC_RunActiveContent.js.");
} else {
    var hasRightVersion = DetectFlashVer(requiredMajorVersion,
        requiredMinorVersion, requiredRevision);
    if(hasRightVersion) { // if we've detected an acceptable version
        // embed the Flash movie
        AC_FL_RunContent(
            &apos;codebase&apos;, &apos;http://download.macromedia.com/pub/
            shockwave/cabs/flash/swflash.cab#version=9,0,28,0&apos;,
            &apos;width&apos;, &apos;550&apos;,
            &apos;height&apos;, &apos;400&apos;,
            &apos;src&apos;, &apos;fullscreen&apos;,
            &apos;quality&apos;, &apos;high&apos;,
            &apos;pluginspage&apos;, &apos;http://www.macromedia.com/go/
            getflashplayer&apos;,
            &apos;align&apos;, &apos;middle&apos;,
            &apos;play&apos;, &apos>true&apos;,
            &apos;loop&apos;, &apos>true&apos;,
            &apos;scale&apos;, &apos;showall&apos;,
            &apos;wmode&apos;, &apos;window&apos;,
            &apos;devicefont&apos;, &apos>false&apos;,
            &apos;id&apos;, &apos;fullscreen&apos;,
            &apos;bgcolor&apos;, &apos;#ffffff&apos;,
            &apos;name&apos;, &apos;fullscreen&apos;,
            &apos;menu&apos;, &apos>true&apos;,
            &apos;allowScriptAccess&apos;, &apos;sameDomain&apos;,
```



```

        &apos;allowFullScreen&apos;; &apos>true&apos;;
        &apos;movie&apos;; &apos;fullscreen&apos;;
        &apos;salign&apos;; &apos;&apos; ); //end AC code
    } else { // Flash is too old or we can&apos;t detect the plug-in.
        var alternateContent = &apos;Alternative HTML content should be placed
            here.&apos;;
        + &apos;This content requires Adobe Flash Player.&apos;;
        + &apos;<a href=http://www.macromedia.com/go/getflash/>Get Flash</a>
            &apos;;
        document.write(alternateContent); // Insert non-Flash content.
    }
}
// -->
</script>
<noscript>
    // Provide alternative content for browsers that do not support scripting
    // or for those that have scripting disabled.
    Alternative HTML content should be placed here. This content requires Adobe Flash Player.
    <a href="http://www.macromedia.com/go/getflash/">Get Flash</a>
</noscript>

```

Du kan också använda den exporterade HTML-filen som mall för din webbsida och lägga till resterande innehåll på den. Om du gör det bör du dock ändra namnet på HTML-filen så att du inte av misstag skriver över den om du senare skulle exportera HTML-filen från Flash.

Du måste i alla fall överföra filen AC_RunActiveContent.js, som exporteras till samma mapp som HTML-filen, till din webserver.

ActionScript-stöd för helskärmsläge inkluderar egenskaperna `fullScreenBackgroundColor`, `fullScreenSkinDelay` och `fullScreenTakeOver` och metoden `enterFullScreenDisplayState()`. Information om de här ActionScript-elementen finns i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Använda `enterFullScreenDisplayState()`

Du kan även aktivera helskärmsläge genom att anropa metoden `enterFullScreenDisplayState()`, enligt exemplet nedan.

```

function handleClick(e:MouseEvent):void {
    myFLVPlybk.enterFullScreenDisplayState();
}
myButton.addEventListener(MouseEvent.CLICK, handleClick);

```

I det här exemplet aktiveras *inte* helskärmsläget när du klickar på alternativknappen för helskärmsläge i ett FLVPlayback-skal, utan läget aktiveras när du klickar på en knapp (MyButton) som den som har skapat webbsidan har inkluderat för att aktivera helskärmsläget. När du klickar på knappen aktiveras händelsehanteraren `handleClick` som anropar metoden `enterFullScreen DisplayState()`.

Metoden `enterFullScreenDisplayState()` ställer in egenskapen `Stage.displayState` property till `StageDisplayState.FULL_SCREEN` och får därmed samma begränsningar som egenskapen `displayState`. Mer information om metoden `enterFullScreenDisplayState()` och egenskapen `Stage.Egenskapen displayState`, se [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Maskinvaruacceleration

Flash Player 9.0.115.0 och senare versioner innehåller kod som drar nytta av tillgänglig videomaskinvara för att förbättra prestanda och återgivningen av de FLV-filer som FLVPlayback spelar upp i helskrmsläge. Om förutsättningarna är uppfyllda och egenskapen `fullScreenTakeOver` är inställd till `true` använder Flash Player maskinvaruacceleration för att ändra skala på videofilen, i stället för att göra det via programvaran. Om FLVPlayback-komponenten körs i en tidigare version av Flash Player, eller om förutsättningarna för maskinvaruacceleration inte finns, ändrar Flash Player skalan på videofilen, som förut.

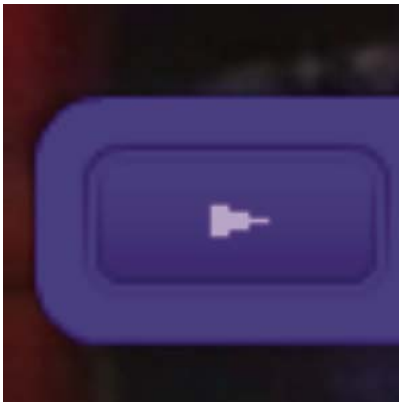
För att utnyttja maskinvaruacceleration för helskrmsstöd måste datorn ha ett DirectX 7-kompatibelt videokort med 4 MB eller mer i VRAM-minne (videominne). Det här maskinvarustödet finns i Windows 2000 och Mac OS X 10.2 samt senare versioner av dessa operativsystem. Direct X[®] innehåller API:er som utgör ett gränssnitt mellan programvara och videomaskinvara för att till exempel accelerera tre- och tvådimensionella bilder.

För att kunna använda accelerationsläget för maskinvara måste du aktivera helskrmsläget på något av följande sätt:

- Använda alternativknappen för helskrmsläge på ett FLVPlayback-skal
- Använda videokontrollen `FullScreenButton`
- Använda ActionScript-metoden `enterFullScreenDisplayState()`. Mer information finns i ”[Använda enterFullScreenDisplayState\(\)](#)” på sidan 141.

Om du aktiverar helskrmsläget genom att ställa in egenskapen `Stage.displayState` till `StageDisplayState.FULLSCREEN` använder inte FLVPlayback maskinvaruacceleration, även om videomaskinvaran och minnet är tillgängligt.

En följd av att använda maskinvaruacceleration för helskrmsstöd är att FLVPlayback-skalen anpassas efter videospelaren och videofilen. Bilden nedan visar hur helskrmsläget ser ut med maskinvaruacceleration på FLVPlayback-skalet (en detalj visas i fullständig upplösning).



Helskrmsläge på en 1600x1200-skärm med en 320x240-pixelvideo

Den här bilden visar resultatet när helskrmsläge används på en bildskärm med upplösningen 1600 x 1200 tillsammans med en videofil som har bredden 320 och höjden 240, vilka är FLVPlayback-komponentens standarddimensioner. Förvrängningseffekten på skalet är tydligare på FLV-filer med mindre dimensioner och på större bildskärmar. Förvrängningen syns därför mindre på större FLV-filer och på mindre bildskärmar. Om du till exempel ändrar från 640 x 480 till 1600 x 1200 ökar storleken på skalet, men det ser mindre förvrängt ut.

Du kan ställa in egenskapen `skinScaleMaximum` för att begränsa anpassningen av FLVPlayback-skalet. Standardvärdet är 4,0 eller 400 %. För att begränsa anpassningen av skalet krävs dock en viss kombination av maskinvara och programvara för att ändra skalan på FLV-filen och det kan påverka prestanda negativt för FLV-filer med stora dimensioner som är kodade med en hög bithastighet. Om videofilen är stor (till exempel minst 640 pixlar bred och 480 pixlar hög) bör inte `skinScaleMaximum` ställas in på ett lågt värde eftersom det kan orsaka märkbara prestandaproblem på stora bildskärmar. Med egenskapen `skinScaleMaximum` kan du balansera prestanda och kvalitet samt utseendet på ett stort skal.

Avsluta helskärmsläge

Om du vill avsluta helskärmsläget klickar du på knappen för helskärmsläge eller på Esc-tangenten.

Om du ställer in följande egenskaper och anropar följande metoder kan det leda till layoutförändringar som kan göra att FLVPlayback-komponenten avslutar helskärmsläget: `height`, `registrationHeight`, `registrationWidth`, `registrationX`, `registrationY`, `scaleX`, `scaleY`, `width`, `x`, `y`, `setScale()` eller `setSize()`.

Om du ställer in egenskaperna `align` eller `scaleMode` ställer FLVPlayback in dem till `center` och `maintainAspectRatio` tills helskärmsläget är avslutat.

Om du ändrar värdet för `fullScreenTakeOver` från `true` till `false` när du använder helskärmsläget orsakar maskinvaruaccelerationsläget att Flash avslutar helskärmsläget.

Layoutjustering för uppspelning av flera videofiler

ActionScript 3.0 FLVPlayback har en `align`-egenskap som anger om en videofil ska centreras när dess storlek ändras, eller om den ska placeras högst upp, längst ned, till vänster eller till höger om komponenten. Utöver komponentens `x`-, `y`-, `width`- och `height`-egenskaper har ActionScript 3.0-komponenten också `registrationX`-, `registrationY`-, `registrationWidth`- och `registrationHeight`-egenskaper. Till att börja med matchar de här egenskaperna egenskaperna `x`, `y`, `width` och `height`. När du laddar ytterligare videofiler ändras inte egenskaperna av automatisk omlayout, så nya videofiler kan centreras på samma plats. Om `scaleMode = VideoScaleMode.MAINTAIN_ASPECT_RATIO` kan efterföljande FLV-filer passas in i komponentens ursprungliga mått i stället för att göra så att komponentens bredd och höjd ändras.

Automatisk uppspelning av progressivt nedladdade videofiler

När du läser in en progressivt nedladdad videofil börjar FLVPlayback spela upp videofilen först när tillräckligt stor del har lästs in så att den kan spelas upp från början till slut.

Om du vill spela upp videofilen innan tillräckligt stor del har lästs in anropar du metoden `play()` utan några parametrar.

Om du vill återgå till läget där du väntar på att tillräckligt stor del av videofilen ska läsas in anropar du först metoden `pause()` och sedan metoden `playWhenEnoughDownloaded()`.

Använda referenspunkter

En referenspunkt är en punkt vid vilken videospelaren skickar en `cuePoint`-händelse när en videofil spelas upp. Du kan lägga till referenspunkter i en FLV-fil vid det tidpunkter när du vill att en händelse ska inträffa för ett annat element på webbsidan. Du kanske vill visa text eller grafik, synkronisera med en Flash-animering eller påverka uppspelningen av FLV-filen genom att pausa den, söka efter en annan tidpunkt i videon eller växla till en annan FLV-fil. Med referenspunkter kan du styra ActionScript-koden så att du synkroniserar punkter i FLV-filen med andra åtgärder på webbsidan.

Det finns tre sorters referenspunkter: navigering, händelse och ActionScript Navigerings- och händelsereferenspunkterna kallas också *inbäddade* eftersom de är inbäddade i FLV-uppspelningen och i FLV-filens metadatapaket.

Med en *navigeringsreferenspunkt* kan du söka upp en viss bildruta i FLV-filen eftersom den skapar en nyckelbildruta i FLV-filen så nära som möjligt till den tidpunkt som du anger. En *nyckelbildruta* är ett datasegment som uppstår mellan bildrutor i FLV-uppspelningen. När du söker efter en navigeringsreferenspunkt söker komponenten efter nyckelbildrutan och startar `cuePoint`-händelsen.

Med en *händelsereferenspunkt* kan du synkronisera en tidpunkt i FLV-filen med en extern händelse på webbsidan. Händelsen `cuePoint` inträffar exakt vid den angivna tidpunkten. Du kan bädda in navigerings- och händelsereferenspunkter i en FLV-fil med guiden Importera video eller Flash Video Encoder. Mer information om guiden Importera video och Flash Video Encoder finns i kapitel 16, ”Arbeta med video” i *Använda Flash*.

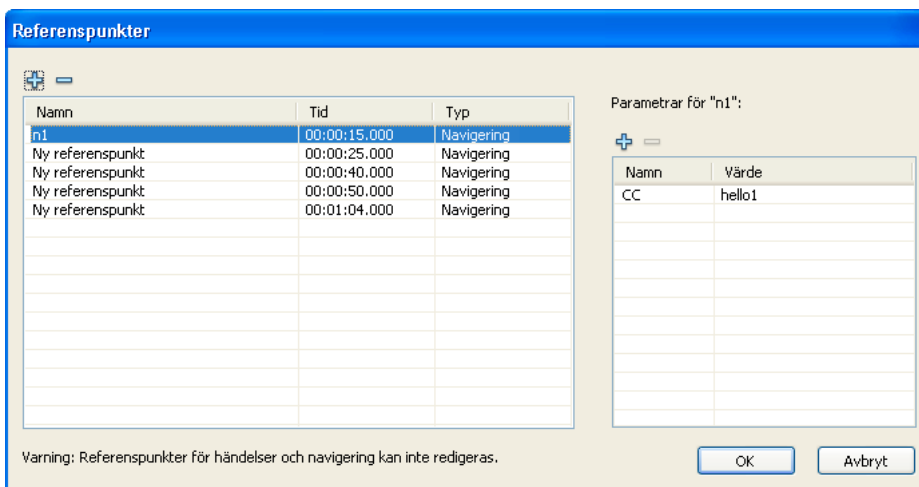
En *ActionScript-referenspunkt* är en extern referenspunkt som du kan lägga till via komponentens dialogruta Referenspunkter för Flash-video eller med metoden `FLVPlayback.addASCuePoint()`. Komponentens lagrar och spårar ActionScript-referenspunkter utanför FLV-filen, och därför är de mindre noggranna än de inbäddade referenspunkterna. ActionScript-referenspunkter kan avvika med en tiondels sekund. Du kan öka noggrannheten för ActionScript-referenspunkter genom att sänka värdet på egenskapen `playheadUpdateInterval`, eftersom komponenten genererar `cuePoint`-händelsen för ActionScript-referenspunkter när spelhuvudet uppdateras. Du hittar mer information i avsnittet om egenskapen `FLVPlayback.playheadUpdateInterval` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

I ActionScript och i FLV-filens metadata representeras en referenspunkt av ett objekt med följande egenskaper: `name`, `time`, `type` och `parameters`. Egenskapen `name` är en sträng som innehåller referenspunktens tilldelade namn. Egenskapen `time` är en siffra som anger tiden i timmar, minuter, sekunder och millisekunder (HH:MM:SS.mmm) när referenspunkten inträffar. Egenskapen `type` är en sträng vars värde är "navigation", "event" eller "actionscript", beroende på vilken typ av referenspunkt som du har skapat. Egenskapen `parameters` är en array med angivna par med namn och värden.

När en `cuePoint`-händelse inträffar är referenspunktobjektet tillgängligt i händelseobjektet via egenskapen `info`.

Använda dialogrutan Referenspunkter för Flash-video

Öppna dialogrutan Referenspunkter för Flash-video genom att dubbelklicka på cellen Värde för `cuePoints`-parametern i komponentinspektören. Dialogrutan ser ut som i följande bild:



Dialogruta för referenspunkter

I dialogrutan visas inbäddade referenspunkter och ActionScript-referenspunkter. Du kan använda den här dialogrutan för att lägga till och ta bort ActionScript-referenspunkter och parametrar för referenspunkter. Du kan också aktivera och inaktivera inbäddade referenspunkter. Men du kan inte lägga till, ändra eller ta bort inbäddade referenspunkter.

Lägga till en ActionScript-referenspunkt:

- 1 Dubbelklicka på värdecellen för parametern `cuePoints` i komponentinspektören för att öppna dialogrutan Referenspunkter för Flash-video.
- 2 Klicka på plustecknet (+) i det övre vänstra hörne ovanför listan med referenspunkter för att lägga till en ActionScript-referenspunkt som standard.
- 3 Klicka på texten Ny referenspunkt i kolumnen Namn och redigera texten för att ge referenspunkten ett namn.
- 4 Klicka på tidsvärdet 00:00:00:000 för att redigera det och ange en tid då referenspunkten ska inträffa. Du kan ange tiden i timmar, minuter, sekunder och millisekunder (HH:MM:SS.mmm).
Om det finns flera referenspunkter flyttar dialogrutan nästa referenspunkt till dess kronologiska position i listan.
- 5 Om du vill lägga till en parameter för den valda referenspunkten klickar du på plustecknet (+) ovanför avsnittet Parametrar och anger värden i kolumnerna Namn och Värde. Upprepa det här steget för alla parametrar.
- 6 Om du vill lägga till fler ActionScript-referenspunkter upprepar du steg 2 till 5 för varje referenspunkt.
- 7 Klicka på OK för att spara ändringarna.

Ta bort en ActionScript-referenspunkt:

- 1 Dubbelklicka på värdecellen för parametern `cuePoints` i komponentinspektören för att öppna dialogrutan Referenspunkter för Flash-video.
- 2 Markera den referenspunkt som du vill ta bort.
- 3 Klicka på minustecknet (-) i det övre vänstra hörnet, ovanför listan med referenspunkter, för att ta bort den.
- 4 Upprepa steg 2 och 3 för varje referenspunkt som du vill ta bort.
- 5 Klicka på OK för att spara ändringarna.

Så här aktiverar eller inaktiverar du en inbäddad referenspunkt i en FLV-fil:

- 1 Dubbelklicka på värdecellen för parametern `cuePoints` i komponentinspektören för att öppna dialogrutan Referenspunkter för Flash-video.
- 2 Markera referenspunkten som du vill aktivera eller inaktivera.
- 3 Klicka på värdet i kolumnen Typ för att visa popup-menyn, eller klicka på nedpilen.
- 4 Klicka på namnet på referenspunktstypen (till exempel Event eller Navigation) för att aktivera den. Klicka på Inaktiverad för att inaktivera den.
- 5 Klicka på OK för att spara ändringarna.

Använda referenspunkter med ActionScript

Du kan använda ActionScript för att lägga till ActionScript-referenspunkter, avlyssna `cuePoint`-händelser, hitta alla referenspunkter eller referenspunkter av en viss typ, söka efter en navigeringsreferenspunkt, aktivera eller inaktivera en referenspunkt, kontrollera om en referenspunkt är aktiverad och ta bort en referenspunkt.

Exemplet i det här avsnittet använder en FLV-fil som heter `cuepoints.flv`, och som innehåller följande tre punkter:

Namn	Tidpunkt	Typ
point1	00:00:00.418	Navigation
point2	00:00:07.748	Navigation
point3	00:00:16.020	Navigation

Lägga till ActionScript-referenspunkter

Du kan lägga till ActionScript-referenspunkter i en FLV-fil med metoden `addASCuePoint()`. Följande exempel lägger till två ActionScript-referenspunkter i FLV-filen när den är klar för uppspelning. Den första referenspunkten läggs till med ett referenspunktsobjekt som anger tidpunkten, namnet och typen för referenspunkten i sina egenskaper. Det andra anropet anger tidpunkten och namnet med metodens `time-` och `name-`parametrar.

```
// 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");
```

Mer information finns i avsnittet om metoden `FLVPlayback.addASCuePoint()` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Avlyssna cuePoint-händelser

Med `cuePoint`-händelsen kan du få kontroll i din ActionScript-kod när en `cuePoint`-händelse inträffar. När en referenspunkt inträffar i följande exempel anropar `cuePoint`-avlyssnaren en händelsehanterarfunktion som visar värdet på egenskapen `playheadTime` samt namnet på och typen av referenspunkt. Använd det här exemplet i kombination med exemplet i föregående avsnitt, *Lägga till ActionScript-referenspunkter*, för att visa resultatet.

```
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);
}
```

Mer information om händelsen `cuePoint` finns i avsnittet om händelsen `FLVPlayback.cuePoint` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Söka efter referenspunkter

Med ActionScript kan du hitta referenspunkter av alla typer, hitta den referenspunkt som ligger närmast en viss tidpunkt eller hitta nästa referenspunkt med ett visst namn.

Händelsehanteraren `ready_listener()` i följande exempel anropar metoden `findCuePoint()` för att hitta referenspunkten `ASpt1`, och anropar sedan metoden `findNearestCuePoint()` för att hitta navigeringsreferenspunkten som ligger närmast referenspunkten `ASpt1` i tiden:

```
import fl.video.FLVPlayback;
import fl.video.CuePointType;
import fl.video.VideoEvent;
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var rtn_obj:Object; //create cue point object
my_FLVPlayback.addASCuePoint(2.02, "ASpt1");//add AS cue point
function ready_listener(eventObject:VideoEvent):void {
    rtn_obj = my_FLVPlayback.findCuePoint("ASpt1", CuePointType.ACTIONSCRIPT);
    traceit(rtn_obj);
    rtn_obj = my_FLVPlayback.findNearestCuePoint(rtn_obj.time, CuePointType.NAVIGATION);
    traceit(rtn_obj);
}
my_FLVPlayback.addEventListener(VideoEvent.READY, ready_listener);
function traceit(cuePoint:Object):void {
    trace("Cue point name is: " + cuePoint.name);
    trace("Cue point time is: " + cuePoint.time);
    trace("Cue point type is: " + cuePoint.type);
}
```

I följande exempel hittar händelsehanteraren `ready_listener()` referenspunkten `ASpt` och anropar metoden `findNextCuePointWithName()` för att hitta nästa referenspunkt med samma namn:

```
import fl.video.*;
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var rtn_obj:Object; //create cue point object
my_FLVPlayback.addASCuePoint(2.02, "ASpt");//add AS cue point
my_FLVPlayback.addASCuePoint(3.4, "ASpt");//add 2nd ASpt
my_FLVPlayback.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:VideoEvent):void {
    rtn_obj = my_FLVPlayback.findCuePoint("ASpt", CuePointType.ACTIONSCRIPT);
    traceit(rtn_obj);
    rtn_obj = my_FLVPlayback.findNextCuePointWithName(rtn_obj);
    traceit(rtn_obj);
}
function traceit(cuePoint:Object):void {
    trace("Cue point name is: " + cuePoint.name);
    trace("Cue point time is: " + cuePoint.time);
    trace("Cue point type is: " + cuePoint.type);
}
```

Mer information om hur du söker efter referenspunkter finns i avsnitten om metoderna `FLVPlayback.findCuePoint()`, `FLVPlayback.findNearestCuePoint()` och `FLVPlayback.findNextCuePointWithName()` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Söka efter navigeringsreferenspunkter

Du kan söka till en navigeringsreferenspunkt, söka till nästa navigeringsreferenspunkt från en viss tid och söka till föregående navigeringsreferenspunkt från en viss tid. Följande exempel spelar upp FLV-filen `cuepoints.flv` och söker till referenspunkten vid 7,748 när `ready`-händelsen inträffar. När `cuePoint`-händelsen inträffar anropar exemplet metoden `seekToPrevNavCuePoint()` för att söka efter den första referenspunkten. När den här `cuePoint`-händelsen inträffar anropar exemplet metoden `seekToNextNavCuePoint()` för att söka efter den sista referenspunkten genom att lägga till 10 sekunder till `eventObject.info.time`, som är tidpunkten för den aktuella referenspunkten.

```
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";
```

Mer information finns i avsnitten om metoderna `FLVPlayback.seekToNavCuePoint()`, `FLVPlayback.seekToNextNavCuePoint()` och `FLVPlayback.seekToPrevNavCuePoint()` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Aktivera och inaktivera en inbäddad referenspunkt i en FLV-fil

Du kan aktivera och inaktivera inbäddade referenspunkter i FLV-filer genom att använda metoden `setFLVCuePointEnabled()`. Inaktiverade referenspunkter utlöser inte `cuePoint`-händelser och fungerar inte med metoderna `seekToCuePoint()`, `seekToNextNavCuePoint()` och `seekToPrevNavCuePoint()`. Däremot kan du hitta inaktiverade referenspunkter med metoderna `findCuePoint()`, `findNearestCuePoint()` och `findNextCuePointWithName()`.

Du kan testa om en inbäddad referenspunkt i en FLV-fil är aktiverad genom att använda metoden `isFLVCuePointEnabled()`. Följande exempel inaktiverar de inbäddade referenspunkterna `point2` och `point3` när videon är klar att spelas upp. Men när den första `cuePoint`-händelsen inträffar testar händelsehanteraren om referenspunkten `point3` är inaktiverad, och aktiverar den om den är inaktiverad.

```
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");
    }
}
```

Mer information finns i avsnitten om metoderna `FLVPlayback.isFLVCuePointEnabled()` och `FLVPlayback.setFLVCuePointEnabled()` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Ta bort en ActionScript-referenspunkt

Du kan ta bort en ActionScript-referenspunkt med metoden `removeASCuePoint()`. Följande exempel tar bort referenspunkten `ASpt2` när referenspunkten `ASpt1` inträffar:

Använda komponenten FLVPlayback

```
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");
    }
}
```

Mer information finns i avsnittet om `FLVPlayback.removeASCuePoint()` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Spela upp flera videofiler

Du kan spela upp videofiler sekventiellt i en `FLVPlayback`-instans genom att helt enkelt läsa in en ny URL i egenskapen `source` när den föregående videofilen slutar spelas upp. Följande ActionScript-kod avlyssnar till exempel `complete`-händelsen som inträffar när en videofil slutar spelas upp. När den här händelsen inträffar anger koden namnet och platsen för en ny videofil i egenskapen `source` och anropar metoden `play()` för att spela upp den nya videon.

```
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");
    }
};
```

Använda flera videospelare

Du kan också öppna flera videospelare i en enda instans av `FLVPlayback`-komponenten för att spela upp flera videor och växla mellan dem när de spelas upp.

Du skapar den första videospelaren när du drar `FLVPlayback`-komponenten till scenen. Komponentens automatiskt den första videospelaren siffran 0 och gör den till standardvideospelaren. Om du vill skapa ytterligare en videospelare ställer du helt enkelt in ett nytt nummer för egenskapen `activeVideoPlayerIndex`. Genom att ställa in egenskapen `activeVideoPlayerIndex` gör du den angivna videospelaren till den *aktiva* videospelaren, som är den spelare som kommer att påverkas av egenskaperna och metoderna i klassen `FLVPlayback`. Däremot blir inte videospelaren synlig när du ställer in egenskapen `activeVideoPlayerIndex`. Om du vill göra videospelaren synlig anger du videospelarens nummer i egenskapen `visibleVideoPlayerIndex`. Mer information om hur de här egenskaperna samverkar med metoderna och egenskaperna i klassen `FLVPlayback` finns i avsnitten om egenskaperna `FLVPlayback.activeVideoPlayerIndex` och `FLVPlayback.visibleVideoPlayerIndex` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Följande ActionScript-kod laddar egenskapen `source` för att spela upp en videofil i standardvideospelaren och lägger till en referenspunkt för den. När `ready`-händelsen inträffar öppnar händelsehanteraren ytterligare en videospelare genom att ställa in egenskapen `activeVideoPlayerIndex` till 1. Den anger en videofil och en referenspunkt för den andra videospelaren och gör sedan standardspelaren (0) till den aktiva videospelaren igen.

```
/**
 * 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;
};
```

Om du vill växla till en annan FLV-fil när en fil spelas upp måste du växla i ActionScript-koden. Med referenspunkter kan du gå emellan vid angivna punkter i FLV-filen med hjälp av en `cuePoint`-händelse. Följande kod skapar en avlyssnare för `cuePoint`-händelsen och anropar en hanterarfunktion som pausar den aktiva videospelaren (0), växlar till den andra spelaren (1) och spelar upp dess videofil:

```
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);
    }
};
```

När du skapar en ny videospelare anger FLVPlayback-instansen standardvideospelarens värden för den nya videospelarens egenskaper, förutom egenskaperna `source`, `totalTime`, och `isLive` som FLVPlayback-instansen alltid ställer in standardvärdena för: tom sträng, 0 och `false`. Egenskapen `autoPlay`, som har värdet `true` för standardvideospelaren, får värdet `false`. Egenskapen `cuePoints` har ingen effekt, och den har ingen effekt på en följande hämtning till standardvideospelaren.

Metoderna och egenskaperna som styr volym, placering, mått, synlighet och gränssnittskontroller är alltid globala och deras beteende påverkas inte av att du ställer in egenskapen `activeVideoPlayerIndex`. Mer information om de här metoderna och egenskaperna, och om effekten av att ange egenskapen `activeVideoPlayerIndex`, finns i avsnittet om egenskapen `FLVPlayback.activeVideoPlayerIndex` i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*. De återstående egenskaperna och metoderna anger videospelaren, som identifieras av värdet på egenskapen `activeVideoPlayerIndex`, som mål.

Egenskaper och metoder som styr mått *fungerar dock tillsammans* med egenskapen `visibleVideoPlayerIndex`. Mer information finns i avsnittet om egenskapen `FLVPlayback.visibleVideoPlayerIndex` i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

Direktuppspela videofiler från Flash Media Server

Kraven för att direktuppspela FLV-filer från Flash Media Server är olika beroende på om inbyggd bandbreddsidentifiering är tillgänglig från din leverantör av Flash Video Streaming Service. Inbyggd bandbreddsidentifiering betyder att bandbreddsidentifieringen är inbyggd i direktuppspelningsservern och det ger bättre prestanda. Kontrollera med din leverantör om du inbyggd bandbreddsidentifiering är tillgänglig.

Om du vill komma åt dina videofiler på Flash Media Server använder du en URL av typen `rtmp://mitt_servernamn/mitt_program/streamflv`.

När du spelar upp ett direktflöde med Flash Media Server måste du ställa in värdet `true` för FLVPlayback-egenskapen `isLive`. Mer information finns i avsnittet om egenskapen `FLVPlayback.isLive` i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

Mer information om hur du administrerar Flash Media Server, inklusive hur du ställer in ett direktflöde, finns i dokumentationen till Flash Media Server på www.adobe.com/support/documentation/en/flashmediaserver/.

För inbyggd bandbreddsidentifiering eller ingen bandbreddsidentifiering

Klassen `NCManagerNative` är en underklass till `NCManager` som stöder inbyggd bandbreddsidentifiering, som vissa leverantörer av Flash Video Streaming Service kan stödja. När du använder `NCManagerNative` behövs inga särskilda filer på Flash Media Server. Med `NCManagerNative` kan du också ansluta till alla versioner av Flash Media Server utan en `main.asc`-fil, om bandbreddsidentifiering inte krävs.

Om du vill använda `NCManagerNative` i stället för den `NCManager`-klass som är standard lägger du till följande kodrader i den första bildrutan i din FLA-fil:

```
import fl.video*;  
VideoPlayer.INCManagerClass = fl.video.NCManagerNative;
```

För icke inbyggd bandbreddsidentifiering

Om inbyggd bandbreddsidentifiering inte går att få från din leverantör av Flash Video Streaming Service och du behöver bandbreddsidentifiering, måste du lägga till filen `main.asc` i ditt Flash Media Server FLV-program. Du hittar filen `main.asc` på följande plats: www.adobe.com/go/learn_fl_samples_se. Den finns i filen `Samples.zip`, i katalogen `Samples\ComponentsAS2\FLVPlayback`.

Så här ställer du in Flash Media Server för direktuppspelning av FLV-filer:

- 1 Skapa en mapp i Flash Media Server-programmappen och ge den ett namn av typen **mitt_program**.
- 2 Kopiera filen main.asc till mappen mitt_program.
- 3 Skapa en mapp med namnet **streams** i mappen mitt_program.
- 4 Skapa en mapp med namnet **_definst_** inuti mappen streams.
- 5 Placera dina FLV-filer i mappen **_definst_**.

Anpassa komponenten FLVPlayback

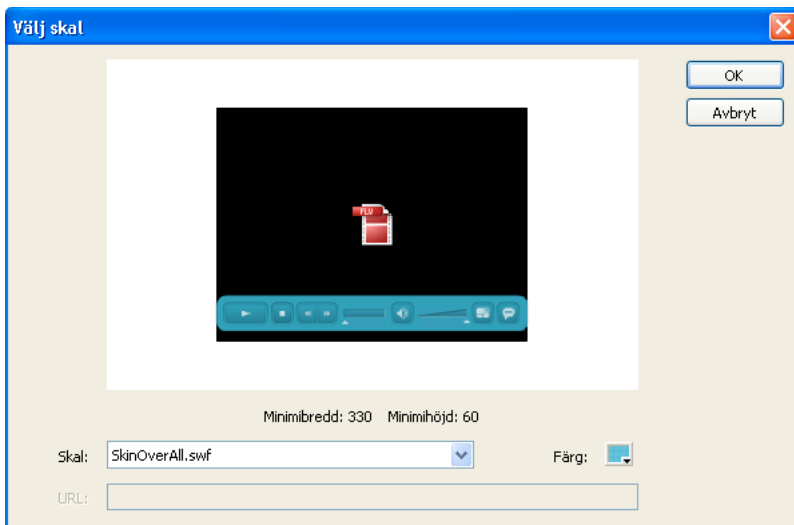
I det här avsnittet förklaras hur du anpassar komponenten FLVPlayback. De flesta metoder som används för att anpassa andra komponenter fungerar inte med komponenten FLVPlayback. Om du vill anpassa komponenten FLVPlayback använder du bara de tekniker som beskrivs i det här avsnittet.

Du har följande alternativ för att anpassa komponenten FLVPlayback: välj ett tidigare utformat skal, skala anpassade användargränssnittskomponenter för FLV-uppspelning individuellt eller skapa ett nytt skal. Du kan också använda FLVPlayback-egenskaper för att ändra ett skals beteende.

Obs! Du måste överföra skal-SWF-filen till webbservern tillsammans med program-SWF-filen för att skalet ska fungera med FLVPlayback-komponenten.

Välja ett tidigare utformat skal

Du kan välja ett skal för FLVPlayback-komponenten genom att klicka på cellen `value` för parametern `skin` i komponentinspektören. Sedan klickar du på förstöringsglaset för att öppna följande dialogruta Välj skal, där du kan välja ett skal eller ange URL som specificerar platsen där skal-SWF-filen finns.

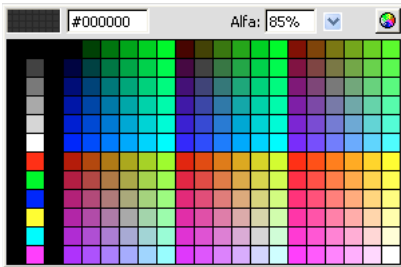


Dialogrutan Välj skal för FLVPlayback

Skalen som finns på popup-menyn Skal är placerade i programmappen för Flash /Flash Configuration/FLVPlayback Skins/ActionScript 3.0. Du kan göra nya skal tillgängliga för den här dialogrutan genom att skapa dem och placera dem i SWF-filen i mappen. Namnet visas på popup-menyn med tillägget `.swf`. Mer information om att skapa skal finns i ”Skapa ett nytt skal” på sidan 158.

Använda komponenten FLVPlayback

För skal som du tilldelar genom att ställa in egenskapen `skin`, antingen genom att ställa in `skin`-parametern vid utveckling eller med ActionScript vid körning, kan du tilldela färg- och alfavärden (genomskinlighet) oberoende av om du väljer skalet. Om du vill tilldela färg- och alfavärden vid utvecklingen öppnar du färgväljaren i dialogrutan Välj skal, som visas nedan.



Färgväljare i dialogrutan Välj skal

Om du vill välja en färg klickar du på en färgruta i panelen eller anger färgens numeriska värde i textrutan. Om du vill välja alfavärdet använder du reglaget eller skriv en procentsats i textrutan Alfa.

Om du vill tilldela färg- och alfavärden vid körning ställer du in egenskaperna `skinBackgroundColor` och `skinBackgroundAlpha`. Ställ in ett `0xRRGGBB`-värde (röd, grön, blå) för egenskapen `skinBackgroundColor`. Ställ in ett nummer mellan 0,0 och 1,0 för egenskapen `skinBackgroundAlpha`. Följande exempel ställer in värdet `0xFF0000` (röd) för `skinBackgroundColor` och 0,5 för `skinBackgroundAlpha`.

```
my_FLVPlybk.skinBackgroundColor = 0xFF0000;
my_FLVPlybk.skinBackgroundAlpha = .5;
```

Standardvärdena är de senaste värdena som väljs av användaren.

Om du vill skala komponenten FLVPlayback med anpassade användargränssnittskomponenter för FLV-uppspelning, väljer du Inget på popup-menyn.

Skala anpassade användargränssnittskomponenter för FLV-uppspelning individuellt

Med de anpassade användargränssnittskomponenterna för FLV-uppspelning kan du anpassa utseendet på FLVPlayback-kontrollerna i FLA-filen och göra så att du ser resultaten när du förhandsvisar webbsidan. Däremot är de här komponenterna inte utformade för att storleksändras. Du bör redigera ett filmklipp och dess innehåll så att de har en angiven storlek. Därför är det vanligtvis bäst att FLVPlayback-komponenten på scenen har den önskade storleken, med `scaleMode` inställd på `exactFit`.

Till att börja med drar du de anpassade användargränssnittskomponenter för FLV-uppspelning som du vill ha från panelen Komponenter, placerar dem där du vill ha dem på scenen och ger dem instansnamn.

De här komponenterna kan fungera utan ActionScript. Om du placerar dem på samma tidslinje och bildruta som FLVPlayback-komponenten och det inte finns något skal i komponenten, ansluter FLVPlayback-komponenten automatiskt till dem. Om du har flera FLVPlayback-komponenter på scenen, eller om den anpassade kontrollen och FLVPlayback-instansen inte finns på samma tidslinje, behövs ActionScript.

När komponenterna finns på scenen redigerar du dem på samma sätt som andra symboler. När du öppnar komponenterna ser du att alla har lite olika inställningar.

Button-komponent

Button-komponenterna har en liknande struktur. Följande Buttons finns: BackButton, ForwardButton, MuteButton, PauseButton, PlayButton, PlayPauseButton och StopButton. De flesta har ett enda filmklipp i bildruta 1 med instansnamnet placeholder_mc. Det är vanligtvis en instans av knappens normala läge, men behöver inte vara det. I bildruta 2 finns fyra filmklipp på scenen för varje visningsläge: normal, över, ned och inaktiverad. (Vid körning kommer komponenten egentligen aldrig till bildruta 2. De här filmklippen är placerade här för att göra redigeringen enklare och för att tvinga dem att läsas in i SWF-filen utan att du behöver markera kryssrutan Exportera i första bildrutan i dialogrutan Symbolegenskaper. Men du måste fortfarande markera alternativet Exportera för ActionScript.)

Om du vill ange ett skal för knappen redigerar du något de här filmklippen. Du kan ändra deras storlek och deras utseende.

Viss ActionScript visas vanligtvis på bildruta 1. Du ska inte behöva ändra det här skriptet. Det stoppar uppspelningshuvudet på bildruta 1 och anger vilket filmklipp som ska användas för olika lägen.

PlayPauseButton, MuteButton, FullScreenButton och CaptionButton

Knapparna PlayPauseButton, MuteButton, FullScreenButton och CaptionButton ställs in på ett annat sätt än de övriga knapparna. De har bara en ram med två lager och inget skript. I ramen finns två knappar ovanpå varandra: för PlayPauseButton finns en Play button och en Pause button, för MuteButton finns en Mute-on button och en Mute-off button, för FullScreenButton finns en full-screen-on button och en full-screen-off button, och för CaptionButton finns en caption-on button och en caption-off button. Om du vill ange skal för de här knapparna anger du skal för var och en av de två interna knapparna enligt beskrivningen i ”[Skala anpassade användargränssnittskomponenter för FLV-uppspelning individuellt](#)” på sidan 153. Ingen ytterligare åtgärd krävs.

CaptionButton är till för komponenten FLVPlaybackCaptioning och måste kopplas till den komponenten, och inte till FLVPlayback-komponenten.

BackButton och ForwardButton

Knapparna BackButton och ForwardButton ställs också in annorlunda än de övriga knapparna. På bildruta 2 har de extra filmklipp som du kan använda som en ram runt en av eller båda knapparna. De här filmklippen är inte obligatoriska och har inga speciella egenskaper. Det finns bara med för att underlätta. Om du vill använda dem drar du dem till scenen från panelen Bibliotek och placerar dem där du vill ha dem. Om du inte vill ha dem låter du bli att använda dem eller tar bort dem från panelen Bibliotek.

De flesta knapparna baseras på en gemensam uppsättning med filmklipp, så att du kan ändra utseendet på alla knapparna samtidigt. Du kan använda den här funktionen, eller du kan byta ut de gemensamma klippen och göra så att alla knappar ser olika ut.

Komponenten BufferingBar

Komponenten BufferingBar är enkel. Den består av en animering som blir synlig när komponenten hamnar i buffertläge, och kräver inget särskilt ActionScript för att konfigureras. Som standard är den ett randigt fält från vänster till höger med en rektangulär mask på för att ge den en ”polkagris”-effekt, men det är inget särskilt med den här konfigurationen.

Även om BufferingBars i skal-SWF-filer använder skalning med nio segment eftersom de måste skalas under körning, behöver den anpassade gränssnittskomponenten BufferingBar FLV inte och *kan inte* använda skalning med nio segment, eftersom den har kapslade filmklipp. Om du vill göra BufferingBar bredare och högre kan du ändra innehållet i stället för att skala den.

Komponenterna SeekBar och VolumeBar

Komponenterna SeekBar och VolumeBar liknar varandra, även om de har olika funktioner. Båda har handtag, använder samma mekanismer för handtagsspårning och har stöd för klipp som är kapslade i dem för att spåra förloppet och fullbordan.

Det finns många platser där ActionScript-koden i FLVPlayback-komponenten antar att registreringspunkten (även *origo* och *nollpunkt*) för SeekBar- eller VolumeBar-komponenten finns i det övre vänstra hörnet av innehållet, så det är viktigt att behålla den konventionen. Annars kan du få problem med handtag och progress- och fullness-filmklipp.

Även om SeekBars i skal-SWF-filer använder skalning med nio segment eftersom de måste skalas under körning, behöver den anpassade gränssnittskomponenten SeekBar FLV inte och *kan inte* använda skalning med nio segment, eftersom den har kapslade filmklipp. Om du vill göra SeekBar bredare och högre kan du ändra innehållet i stället för att skala den.

Handtag

En instans av handtagsfilmklippet finns i bildruta 2. Precis som med komponenterna BackButton och ForwardButton kommer komponenten egentligen aldrig till bildruta 2. De här filmklippen är placerade här för att göra redigeringen enklare och för att tvinga dem att läsas in i SWF-filen utan att du behöver markera kryssrutan Exportera i första bildrutan i dialogrutan Symbolegenskaper. Men du måste ändå markera alternativet Exportera för ActionScript.

Du kan se att handtagsfilmklippet har en rektangel i bakgrunden med värdet 0 för alfa. Den här rektangeln ökar storleken på handtagets träffområde, och gör det enklare att ta tag i det utan att ändra dess utseende, på liknanden sätt som en knappsträffläge. Eftersom handtaget skapas dynamiskt vid körning måste det vara ett filmklipp och inte en knapp. Den här rektangeln med värdet 0 för alfa behövs inte för något annat, och vanligtvis kan du byta ut handtagets insida mot en annan bild. Men det fungerar bäst om du behåller registreringspunkten centrerad i mitten av handtagsfilmklippet.

Följande ActionScript-kod finns i bildruta 1 för SeekBar-komponenten för att hantera handtaget:

```
stop();  
handleLinkageID = "SeekBarHandle";  
handleLeftMargin = 2;  
handleRightMargin = 2;  
handleY = 11;
```

Anropet till funktionen `stop()` är nödvändigt på grund av innehållet i bildruta 2.

Den andra raden anger vilken symbol som ska användas som handtag, och du behöver inte ändra den om du bara redigerar handtagsfilmklippets instans i bildruta 2. Vid körning skapar FLVPlayback-komponenten en instans av det angivna filmklippet på scenen som en jämställd komponent till förkomsten av Bar-komponenten. Det betyder att de har samma överordnade filmklipp. Om en Bar finns på rotnivå måste alltså handtaget också finnas på rotnivå.

Variabeln `handleLeftMargin` bestämmer handtagets ursprungliga plats (0%), och variabeln `handleRightMargin` bestämmer var det finns på slutet (100%). Numren anger förskjutningarna från vänster och höger ände av Bar-kontrollen, där positiva nummer markerar gränserna inuti fältet och negativa nummer markerar gränserna utanför fältet. De här förskjutningarna anger vart handtaget kan ta vägen, baserat på dess registreringspunkt. Om du placerar registreringspunkten i mitten av handtaget hamnar handtagets bitersta vänstra och högra sidor utanför marginalerna. Ett SeekBar-filmklipp måste ha registreringspunkten som innehållets övre vänstra hörn för att fungera korrekt.

Variabeln `handleY` bestämmer handtagets *y*-position i relation till Bar-instansen. Den baseras på varje filmklippets registreringspunkt. Registreringspunkten i exempelhandtaget ligger i toppen på triangeln för att placeras relativt till den synliga delen, och den osynliga träfflägesrektangeln ignoreras. Ett Bar-filmklipp måste också behålla registreringspunkten som innehållets övre vänstra hörn för att fungera korrekt.

Använda komponenten FLVPlayback

Om en Bar-kontroll, med de här gränserna, till exempel ställs in på (100, 100) och den är 100 pixlar bred, kan handtaget variera mellan 102 och 198 vågrätt och förbli 111 lodrätt. Om du ändrar `handleLeftMargin` och `handleRightMargin` till -2 och `handleY` till -11, kan handtaget variera mellan 98 och 202 vågrätt och förbli 89 lodrätt.

Progress- och fullness-filmklipp

Komponenten SeekBar har ett *progress*-filmklipp och VolumeBar har ett *fullness*-filmklipp, men i praktiken kan alla SeekBar och VolumeBar ha något, inget eller båda dessa filmklipp. De är strukturellt desamma och beter sig på liknande sätt, men spårar olika värden. Ett progress-filmklipp fylls när FLV-filen laddas ned (vilket bara är praktiskt för en HTTP-nedladdning, eftersom det alltid är fullt vid direktuppspelning från from FMS). Fullnes-filmklipp fylls när handtaget flyttas från vänster till höger.

Komponenten FLVPlayback hittar de här filmklippinstanserna genom att leta efter ett visst instansnamn, så instansen av progress-filmklippet måste ha ditt Bar-filmklipp som överordnat filmklipp samt instansnamnet `progress_mc`. Instansen av fullness-filmklippet måste ha instansnamnet `fullness_mc`.

Du kan ställa in progress- och fullness-filmklippen med eller utan filmklippinstansen `fill_mc` kapslad. Filmklippet VolumeBar `fullness_mc` visar metoden `med` filmklippet `fill_mc`, och filmklippet SeekBar `progress_mc` visar metoden `utan` filmklippet `fill_mc`.

Metoden med filmklippet `fill_mc` kapslat är användbar när du vill ha en fyllning som inte kan skalas utan att förvränga utseendet.

I filmklippet VolumeBar `fullness_mc` är den kapslade filmklippinstansen `fill_mc` maskerad. Du kan maskera den när du skapar filmklippet, eller också skapas en mask dynamiskt vid körning. Om du maskerar den med ett filmklipp ger du instansen namnet `mask_mc` och ställer in den så att `fill_mc` visas som det skulle när procentandelen är 100 %. Om du inte maskerar `fill_mc` blir den dynamiskt skapade masken rektangulär och får samma storlek som `fill_mc` vid 100 %.

Filmklippet `fill_mc` visas med masken på ett av två sätt, beroende på om `fill_mc.slideReveal` har värdet `true` eller `false`.

Om `fill_mc.slideReveal` har värdet `true` flyttas `fill_mc` från vänster till höger för att visas genom masken. Vid 0 % finns det längst till vänster, så att inget av det visas genom masken. När procentandelen ökar flyttas det åt höger tills det befinner sig vid 100 %, där det skapades på scenen.

Om `fill_mc.slideReveal` har värdet `false` eller är odefinierat (standardbeteendet), ändras maskens storlek från vänster till höger för att visa mer av `fill_mc`. När det är på 0 % skalas masken till 05 vågrätt, och när procentandelen ökar, ökar `scaleX` ända upp till 100 %, då hela `fill_mc` visas. Det behöver inte nödvändigtvis vara `scaleX = 100`, eftersom `mask_mc` kan ha skalats när den skapades.

Metoden utan `fill_mc` är enklare än metoden med `fill_mc`, men den förvränger fyllningen vågrätt. Om du inte vill ha den förvrängningen måste du använda `fill_mc`. SeekBar `progress_mc` illustrerar den här metoden.

Progress- eller fullness-filmklippet skalas vågrätt baserat på procentandelen. Vid 0 % har instansens `scaleX` värdet 0 och blir osynlig. När procentandelen växer justeras `scaleX` tills klippet, vid 100 %, har samma storlek som när det skapades på scenen. Det behöver inte nödvändigtvis vara `scaleX = 100`, eftersom klippinstansen kan ha skalats när den skapades.

Ansluta anpassade användargränssnittskomponenter för FLV-uppspelning

Om du placerar dina anpassade användargränssnittskomponenter på samma tidslinje och i samma bildruta som FLVPlayback-komponenten, och du inte har ställt in egenskapen `skin`, ansluter FLVPlayback dem automatiskt utan att något ActionScript behövs.

Om du har flera FLVPlayback-komponenter på scenen, eller om den anpassade kontrollen och FLVPlayback inte finns på samma tidslinje, måste du skriva ActionScript-kod för att ansluta de anpassade användargränssnittskomponenterna till din instans av FLVPlayback-komponenten. Först måste du ge FLVPlayback-instansen ett namn, och sedan använder du ActionScript för att tilldela instanserna av den anpassade användargränssnittskomponenten för FLV-uppspelning till motsvarande FLVPlayback-egenskaper. I följande exempel är FLVPlayback-instansen `my_FLVPlybk`, FLVPlayback-egenskapsnamnen kommer efter punkterna (.) och instanserna av de anpassade användargränssnittskontrollerna för FLV-uppspelning är placerade till höger om likhetstecknen (=):

```
//FLVPlayback instance = my_FLVPlybk
my_FLVPlybk.playButton = playbtn; // set playButton prop. to playbtn, etc.
my_FLVPlybk.pauseButton = pausebtn;
my_FLVPlybk.playPauseButton = playpausebtn;
my_FLVPlybk.stopButton = stopbtn;
my_FLVPlybk.muteButton = mutebtn;
my_FLVPlybk.backButton = backbtn;
my_FLVPlybk.forwardButton = forbtn;
my_FLVPlybk.volumeBar = volbar;
my_FLVPlybk.seekBar = seekbar;
my_FLVPlybk.bufferingBar = bufbar;
```

Med följande steg skapar du anpassade StopButton-, PlayPauseButton-, MuteButton- och SeekBar-kontroller:

- 1 Dra FLVPlayback-komponenten till scenen och ge den instansnamnet **my_FLVPlybk**.
- 2 Ställ in parametern `source` via komponentinspektören till **<http://www.helpexamples.com/flash/video/cuepoints.flv>**.
- 3 Ställ in parametern `Skal` på Inget.
- 4 Dra en StopButton, en PlayPauseButton och en MuteButton till scenen, och placera dem över FLVPlayback-instansen, så att de staplas lodrätt på vänster sida. Ge varje knapp ett instansnamn i egenskapsinspektören (till exempel **my_stopbtn**, **my_plypausbtn** och **my_mutebtn**).
- 5 I panelen Bibliotek öppnar du mappen med FLVPlayback-skal, och sedan öppnar du mappen SquareButton nedanför den.
- 6 Markera filmklippet SquareBgDown och dubbelklicka på det för att öppna det på scenen.
- 7 Högerklicka (Windows) eller Ctrl-klicka (Macintosh), välj Markera alla på menyn och ta bort symbolen.
- 8 Välj ovalverktyget, rita en oval på samma plats och ställ in blå fyllning (**#0033FF**).
- 9 I egenskapsinspektören ställer du in bredden (W:) till **40** och höjden (H:) till **20**. Ställ in x-koordinaten (X:) till **0.0** och y-koordinaten (Y:) till **0.0**.
- 10 Upprepa steg 6 till 8 för SquareBgNormal, men ändra fyllningen till gul (**#FFFF00**).
- 11 Upprepa steg 6 till 8 för SquareBgOver, men ändra fyllningen till grön (**#006600**).
- 12 Redigera filmklippen för de olika symbolerna i knapparna (PauseIcon, PlayIcon, MuteOnIcon, MuteOffIcon och StopIcon). Filmklippen finns i panelen Bibliotek under FLV Playback Skins/*Etikett* Button/Assets, där *Etikett* är namnet på knappen, till exempel Play, Pause, osv. Följ dessa steg för var och en:
 - a Välj alternativet Markera allt.
 - b Ändra färgen till röd (**#FF0000**).
 - c Ändra skalan till 300 %.
 - d Ändra innehållets X:- plats till **7.0** för att ändra den vågräta placeringen av ikonerna i alla knapplägen.

Obs! Genom att ändra placeringen på det här sättet undviker du att öppna varje knappläge och flytta ikonens filmklippinstans.

- 13 Klicka på den blå Tillbaka-pilen ovanför tidslinjen för att återgå till scen 1, bildruta 1.
- 14 Dra en SeekBar-komponent till scenen och placera den i det nedre högra hörnet av FLVPlayback-instansen.
- 15 I panelen Bibliotek dubbelklickar du på SeekBar för att öppna den på scenen.
- 16 Skala den till 400 %.
- 17 Markera konturen och ställ in röd färg (#FF0000).
- 18 Dubbelklicka på SeekBarProgress i mappen FLVPlayback Skins/Seek Bar och ställ in gul färg (#FFFF00).
- 19 Dubbelklicka på SeekBarHandle i mappen FLVPlayback Skins/Seek Bar och ställ in röd färg (#FF0000).
- 20 Klicka på den blå Tillbaka-pilen ovanför tidslinjen för att återgå till scen 1, bildruta 1.
- 21 Markera SeekBar-instansen på scenen och ge den instansnamnet **my_seekbar**.
- 22 Lägg till en `import`-sats för videoklasserna i panelen Åtgärder på i bildruta 1 i tidslinjen. Tilldela sedan Button- och SeekBar-namnen till motsvarande FLVPlayback-egenskaper, som i följande exempel:

```
import fl.video.*;
my_FLVPlybk.stopButton = my_stopbbtn;
my_FLVPlybk.playPauseButton = my_plypausbbtn;
my_FLVPlybk.muteButton = my_mutebbtn;
my_FLVPlybk.seekBar = my_seekbar;
```
- 23 Tryck på Ctrl+Enter för att testa filmen.

Skapa ett nytt skal

Det bästa sättet att skapa en skal-SWF-fil är att kopiera en av skalfilerna som finns i Flash, och använda den som utgångspunkt. Du hittar FLA-filerna för de här skalerna i Flash-programmappen i Configuration/FLVPlayback Skins/FLA/ActionScript 3.0/. Om du vill göra den färdiga skal-SWF-filen tillgänglig som ett alternativ i dialogrutan Välj skal placerar du den i mappen Configuration/FLVPlayback Skins/ActionScript 3.0, antingen i Flash-programfilen eller i en användares lokala mapp Configuration/FLVPlayback Skins/ActionScript 3.0.

Eftersom du kan ställa in färgen på ett skal utan att välja skalet, behöver du inte redigera FLA-filen för att ändra färgen. Om du skapar ett skal som har en viss färg och du inte vill att det ska gå att redigera i dialogrutan Välj skal, ställer du `in this.border_mc.colorMe = false;` i ActionScript-koden för skalets FLA-fil. Mer information om att ställa in ett skals färg finns i ”[Välja ett tidigare utformat skal](#)” på sidan 152.

När du tittar på de installerade FLA-filerna för Flash-skal kan det se ut som om vissa saker på scenen är onödiga, men många av dem placeras i stöddinjelager. Med direktförhandsvisning som använder skalning med nio segment, ser du snabbt vad som egentligen kommer att visas i SWF-filen.

Följande avsnitt beskriver de mer komplexa anpassningarna och ändringarna för filmklippen SeekBar, BufferingBar och VolumeBar.

Använda skallayouten

När du öppnar en FLA-fil för ett Flash-skal finns skalets filmklipp placerade längs huvudtidslinjen. De här klippen och ActionScript-koden som finns i samma bildruta definierar hur kontrollerna placeras vid körning.

Även om layoutlagret på många sätt liknar skalet som det ser ut vid körning, så är innehållet i det här lagret inte synligt vid körning. Det används bara för att beräkna vara kontrollerna ska placeras. De andra kontrollerna på scenen används vid körning.

I layoutlagret finns en platshållare för FLVPlayback-komponenten som heter video_mc. Alla övriga kontroller är utplacerade i förhållande till video_mc. Om du börjar med en av Flash FLA-filerna och ändrar storleken på kontrollerna kan du antagligen ordna layouten genom att flytta de här platshållarklippen.

Alla platshållarklipp har ett eget instansnamn. Namnen på platshållarklippen är 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 och volumeBarHandle_mc. Den del som får en ny färg när du väljer en skalfärg kallas border_mc.

Vilket klipp som används för att styra en kontroll är inte viktigt. Vanligtvis används normallägesklippet för knappar. För övriga kontroller används klippet för den kontrollen, men det är bara för att underlätta. Det enda som är viktigt är *x*- (vågrät) och *y*-placeringen (lodrät) samt höjden och bredden för platshållaren.

Du kan också ha så många extra klipp som du vill utöver standardkontrollerna. Det enda kravet på de här klippen är att deras bibliotekssymboler har Exportera för ActionScript markerat i dialogrutan Länkning. De anpassade klippen i layoutlagret kan ha vilket instansnamn som helst, förutom de reserverade namnen ovan. Ett instansnamn behövs bara för att ställa in ActionScript för klippet för att bestämma layouten.

Klipplet border_mc är speciellt. Om du anger värdet true för egenskapen FlvPlayback.skinAutoHide visas skalet när musen flyttas över border_mc clip. Det är viktigt för skal som visas utanför videospelarens konturer. Mer information om egenskapen skinAutoHide finns i ”[Ändra skalbeteende](#)” på sidan 162.

I Flash FLA-filer används border_mc för färg och för kantlinjen runt knapparna Framåt och Tillbaka.

Klipplet border_mc är också en del av skalet som får sin alfa och färg ändrad av egenskaperna skinBackgroundAlpha och skinBackgroundColor. För att tillåta anpassningsbar färg och alfa måste ActionScript i skalets FLA-fil innehålla:

```
border_mc.colorMe = true;
```

ActionScript och skallayout

Följande ActionScript-kod gäller generellt för alla kontroller. Vissa kontroller har ett specifikt ActionScript som definierar ytterligare beteenden, och som beskrivs i avsnittet om den kontrollen.

Det ursprungliga ActionScript är ett stort avsnitt som specificerar klassnamnen för varje komponents alla lägen. Alla de här klassnamnen finns i filen SkinOverAll fla. Koden ser till exempel ut så här för knapparna Pause och Play:

```
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";
```

Klassnamnen har inga egentliga externa klassfiler. De specificeras bara i dialogrutan Länkning för alla filmklipp i biblioteket.

I ActionScript 2.0-komponenten fanns det filmklipp på scens som faktiskt användes vid körning. I ActionScript 3.0-komponenten finns dessa filmklipp fortfarande i FLA-filen, men bara för att underlätta redigeringen. Nu finns alla i stödlinjelager och exporteras inte. Alla skalresurser i biblioteket är inställda för att exporteras i den första bildrutan, och de skapas dynamiskt med kod som den här:

```
new fl.video.skin.PauseButtonDisabled();
```

Efter det avsnittet finns ActionScript-kod som definierar den minsta bredden och höjden för skalet. Värdena visas i dialogrutan Välj skal, och de används vid körning för att hindra att skalet skalas under sin minimistorlek. Om du inte vill ange en minimivå låter du den vara odefinierad eller mindre än eller lika med noll.

```
// 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;
```

Alla platshållare kan ha följande egenskaper:

Egenskap	Beskrivning
anchorLeft	Booleskt. Placerar kontrollen i förhållande till FLVPlayback-instansens vänstra sida. Standardvärdet är true såvida inte anchorRight uttryckligen har fått värdet true, och då blir standardvärdet false.
anchorRight	Booleskt. Placerar kontrollen i förhållande till FLVPlayback-instansens högra sida. Standardvärdet är false.
anchorBottom	Booleskt. Placerar kontrollen i förhållande till FLVPlayback-instansens nederkant. Standardvärdet är true såvida inte anchorTop uttryckligen har fått värdet true, och då blir standardvärdet false
anchorTop	Booleskt. Placerar kontrollen i förhållande till FLVPlayback-instansens överkant. Standardvärdet är false.

Om både egenskapen anchorLeft och egenskapen anchorRight är true, skalas kontrollen vågrätt vid körning. Om både egenskapen anchorTop och egenskapen anchorBottom är true, skalas kontrollen lodrätt vid körning.

Om du vill se effekten av de här egenskaperna tittar du på hur de används i Flash-skalen. Kontrollerna BufferingBar och SeekBar är de enda som skalas, och de placeras ovanpå varandra och har båda värdet true för egenskaperna anchorLeft och anchorRight. Alla kontroller till vänster om BufferingBar och SeekBar har värdet true för anchorLeft, och alla kontroller till höger om dem har värdet true för anchorRight. Alla kontroller har värdet true för anchorBottom.

Du kan försöka att redigera filmklippen i layoutlagret för att skapa ett skal där kontrollerna finns högst upp i stället för längst ned. Du flyttar helt enkelt kontrollerna högst upp i förhållande till video_mc, och ställer anchorTop lika med true för alla kontroller.

Buffertkontroll

Buffertkontrollen har två filmklipp: bufferingBar_mc och bufferingBarFill_mc. Ett klipps placering på scenen i förhållande till det andra klippet är viktigt, eftersom den här relativa placeringen upprätthålls. Buffertkontrollen använder två separata klipp eftersom komponenten skalar bufferingBar_mc men inte bufferingBarFill_mc.

Klipppet för bufferingBar_mc har skalning med nio segment, så att kantlinjerna inte förvrängs vid skalningen. Klippet för bufferingBarFill_mc är mycket brett, så att det alltid är tillräckligt brett utan att behöva skalas. Det maskeras automatiskt vid körning så att endast delen ovanför den utsträckta bufferingBar_mc visas. Som standard upprätthåller maskens exakta mått en lika stor marginal till vänster och till höger i bufferingBar_mc, baserat på skillnaden mellan x-placeringen (vågrätt) för bufferingBar_mc och bufferingBarFill_mc. Du kan anpassa placeringen med ActionScript-kod.

Om buffertkontrollen inte behöver skalas, eller inte behöver skalning med nio segment, kan du ställa in den som den anpassade användargränssnittskomponenten BufferingBar för FLV-uppspelning. Mer information finns i ”Komponenten BufferingBar” på sidan 154.

Buffertkontrollen har följande extra egenskap:

Egenskap	Beskrivning
fill_mc:MovieClip	Anger instansnamnet för buffertkontrollens fyllning. Standardvärdet är bufferingBarFill_mc.

Sökfält och volymfält

Sökfältet har också två filmklipp: seekBar_mc och seekBarProgress_mc. Ett klipp placering i layoutlagret i förhållande till det andra klippet är viktig, eftersom den här relativa placeringen upprätthålls. Även om båda klippen skalas kan inte seekBarProgress_mc kapslas i seekBar_mc, eftersom seekBar_mc använder skalning med nio segment, som inte fungerar så bra med kapslade filmklipp.

Klipplet för seekBar_mc har skalning med nio segment, så att kantlinjerna inte förvrängs vid skalningen. Klippet seekBarProgress_mc skalas också, men det förvrängs. Det använder inte skalning med nio segment eftersom det är en fyllning, som ser bra ut även när den är förvrängd.

Klipplet seekBarProgress_mc fungerar utan en fill_mc, ungefär på samma sätt som ett progress_mc-klipp fungerar i anpassade användargränssnittskomponenter för FLV-uppspelning. Med andra ord är det inte maskerat och skalas vågrätt. De exakta måtten för seekBarProgress_mc vid 100 % definieras av vänster- och högermarginalerna i klippet seekBarProgress_mc. De här måtten är som standard likadana och baseras på skillnaden mellan *x*-placeringen (vågrät) av seekBar_mc och seekBarProgress_mc. Du kan anpassa måtten med ActionScript i sökfältets filmklipp, som i följande exempel:

```
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;
```

Du kan placera koden i tidslinjen för filmklippet SeekBar eller tillsammans med den andra ActionScript-koden på huvudtidslinjen. Om du anpassar med kod i stället för att ändra layouten behöver fyllningen inte finnas på scenen. Den behöver bara vara i biblioteket, inställd för export för ActionScript i bildruta 1 med rätt klassnamn.

Precis som med den anpassade användargränssnittskomponenten SeekBar för FLV-uppspelning går det att skapa ett fullness-filmklipp för sökfältet. Om sökfältet inte behöver skalas, eller om det skalas men inte använder skalning med nio segment, kan du ställa in progress_mc eller fullness_mc med någon av metoderna som används för anpassade användargränssnittskomponenter för FLV-uppspelning. Mer information finns i .

Eftersom volymfältet i Flash-skal inte skalas är det utformat på samma sätt den anpassade användargränssnittskomponenten VolumeBar för FLV-uppspelning. Mer information finns i "[Komponenterna SeekBar och VolumeBar](#)" på sidan 155. Undantaget är att handtaget implementeras på ett annat sätt.

Handtagen SeekBar och VolumeBar

Handtagen SeekBar och VolumeBar placeras i layoutlagret bredvid fältet. Som standard ställs handtagets vänstra och högra marginal samt *y*-axelvärdena in enligt dess placering i förhållande till fältets filmklipp. Den vänstra marginalen ställs in av skillnaden mellan handtagets *x*-placering (vågrät) och fältets *x*-placering (vågrät), och den högra marginalen är lika stor som den vänstra. Du kan anpassa de här värdena med ActionScript i filmklippen SeekBar och VolumeBar. Följande exempel är samma ActionScript-kod som används med de anpassade användargränssnittskomponenterna för FLV-uppspelning:

```
this.seekBar_mc.handleLeftMargin = 2;  
this.seekBar_mc.handleRightMargin = 2;  
this.seekBar_mc.handleY = 11;
```

Du kan placera koden i tidslinjen för filmklippet SeekBar eller tillsammans med den andra ActionScript-koden på huvudtidslinjen. Om du anpassar med kod i stället för att ändra layouten behöver handtaget inte finnas på scenen. Den behöver bara vara i biblioteket, inställd för export för ActionScript i bildruta 1 med rätt klassnamn.

Utöver de här egenskaperna är handtagen enkla filmklipp, som ställs in på samma sätt som i de anpassade användargränssnittskomponenterna för FLV-uppspelning. Båda har rektangulära bakgrunder med värdet 0 på `alpha`-egenskapen. Bakgrunderna är bara till för att öka träffområdet och är inte obligatoriska.

Bakgrunds- och förgrundsklipp

Filmklippen `chrome_mc` och `forwardBackBorder_mc` implementeras som bakgrundsklipp.

Av filmklippen `ForwardBackBorder`, `ForwardBorder` och `BackBorder` på scenen och platshållarens Framåt- och Tillbaka-knappar, är det bara `ForwardBackBorder` som *inte* finns i ett stödlinjelager. Det finns bara i skalen som faktiskt använder knapparna Framåt och Tillbaka.

De enda kraven för de här klippen är att de måste exporteras för Actionscript i bildruta 1 i biblioteket.

Ändra skalbeteende

Egenskaperna `bufferingBarHidesAndDisablesOthers` och `skinAutoHide` gör att du kan anpassa FLVPlayback-skalets beteende.

Om du ställer in värdet `true` för egenskapen `bufferingBarHidesAndDisablesOthers` döljer FLVPlayback-komponenten `SeekBar` och dess handtag, och inaktiverar knapparna `Play` och `Pause` när komponenten kommer i buffertläge. Det kan vara användbart när en FLV-fil direktuppspelas från FMS över en långsam anslutning med en hög inställning för egenskapen `bufferTime` (till exempel 10). I den här situationen kan en otålig användare försöka börja söka genom att klicka på knapparna `Play` och `Pause`, vilket kan fördröja uppspelningen av filen ännu mer. Du kan förhindra den här aktiviteten genom att ge `bufferingBarHidesAndDisablesOthers` värdet `true` och inaktivera `SeekBar`-elementet och knapparna `Pause` och `Play` med komponenten är i buffertläge.

Egenskapen `skinAutoHide` påverkar bara tidigare utformade skal-SWF-filer och inte kontroller som skapas från de anpassade användargränssnittskomponenterna för FLV-uppspelning. Om FLVPlayback-komponenten har värdet `true` döljer den skalet när musen inte befinner sig ovanför visningsområdet. Standardvärdet för den här egenskapen är `false`.

Använda en SMIL-fil

För att hantera fler strömmar för flera bandbredder använder `VideoPlayer`-klassen en hjälpklass (`NCManager`) som stöder en delmängd av SMIL. SMIL används för att identifiera videoflödets plats, layouten (bredd och höjd) för FLV-filen och FLV-källfiler som motsvarar de olika bandbredderna. Den kan även användas för att ange FLV-filens överföringshastighet och varaktighet.

Använd parametern `source` eller egenskapen `FLVPlayback.source` (ActionScript) för att ange var en SMIL-fil finns. Mer information finns i avsnitten om `FLVPlayback.source` i [Referenshandbok för ActionScript 3.0 i Adobe Flash Professional](#).

Följande exempel visar en SMIL-fil som direktuppspelar FLV-filer med flera bandbredder från en FMS med 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>
```

Taggen `<head>` kan innehålla taggarna `<meta>` och `<layout>`. Taggen `<meta>` stöder bara attributet `base`, som används för att ange URL:en för direktuppspelningsvideon (RTMP från en FMS).

Taggen `<layout>` stöder bara elementet `root-layout`, som används för att ställa in attributen `height` och `width`, och därför bestämmer storleken på fönstret där videofilen återges. De här attributen godtar bara pixelvärden, och inte procentandelar.

Inuti själva SMIL-filen kan du ta med en enda länk till en FLV-källfil. Om du direktuppspelar flera filer för olika bandbredder från en FMS (som i föregående exempel), kan du också använda taggen `<switch>` för att lista källfilerna.

Taggarna `video` och `ref` inuti taggen `<switch>` är synonyma. Båda kan använda attributet `src` för att ange videofiler. Dessutom kan du ange attributen `region`, `system-bitrate` och `dur` för att ange regionen, den minsta bandbredden som krävs och FLV-filens varaktighet.

I taggen `<body>` tillåts bara en förekomst av antingen `<video>`, `<src>` eller `<switch>`.

Följande exempel visar en progressiv nedladdning för en enstaka FLV-fil som inte använder bandbreddsidentifiering:

```
<smil>
    <head>
        <layout>
            <root-layout width="240" height="180" />
        </layout>
    </head>
    <body>
        <video src="myvideo.flv" />
    </body>
</smil>
```

<smil>

Tillgänglighet

Flash Professional 8.

Användning

```
<smil>
...
child tags
...
</smil>
```

Attribut

Ingen.

Underordnade taggar

<head>, <body>

Överordnad tagg

Ingen.

Beskrivning

Toppnivåtagg som identifierar en SMIL-fil.

Exempel

Följande exempel visar en SMIL-fil som anger FLV-filer:

```
<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>

Tillgänglighet

Flash Professional 8.

Användning

```
<head>
...
child tags
...
</head>
```

Attribut

Ingen.

Underordnade taggar

<meta>, <layout>

Överordnad tagg

`<smil>`

Beskrivning

Stöder taggarna `<meta>` och `<layout>`, anger platsen och standardlayouten (höjd och bredd) för FLV-källfilerna.

Exempel

Följande exempel ställer in rotlayouten till 240 x 180 pixlar:

```
<head>
  <meta base="rtmp://myserver/myapp/" />
  <layout>
    <root-layout width="240" height="180" />
  </layout>
</head>
```

<meta>

Tillgänglighet

Flash Professional 8.

Användning

`<meta/>`

Attribut

`base`

Underordnade taggar

`<layout>`

Överordnad tagg

Ingen.

Beskrivning

Innehåller attributet `base` som anger platsen (RTMP-URL) för FLV-källfilerna.

Exempel

Följande exempel visar en meta-tag för en basplacering på myserver:

```
<meta base="rtmp://myserver/myapp/" />
```

<layout>

Tillgänglighet

Flash Professional 8.

Användning

```
<layout>  
...  
child tags  
...  
</layout>
```

Attribut

Ingen.

Underordnade taggar

```
<root-layout>
```

Överordnad tagg

```
<meta>
```

Beskrivning

Anger FLV-filens bredd och höjd.

Exempel

Följande exempel anger layouten till 240 pixlar gånger 180 pixlar:

```
<layout>  
  <root-layout width="240" height="180" />  
</layout>
```

<root-layout>

Tillgänglighet

Flash Professional 8.

Användning

```
<root-layout...attributes.../>
```

Attribut

Bredd, höjd

Underordnade taggar

Ingen.

Överordnad tagg

```
<layout>
```

Beskrivning

Anger FLV-filens bredd och höjd.

Exempel

Följande exempel anger layouten till 240 pixlar gånger 180 pixlar:

```
<root-layout width="240" height="180" />
```

<brödtext>

Tillgänglighet

Flash Professional 8.

Användning

```
<body>  
...  
child tags  
...  
</body>
```

Attribut

Ingen.

Underordnade taggar

<video>, <ref>, <switch>

Överordnad tagg

<smil>

Beskrivning

Innehåller taggarna <video>, <ref> och <switch>, som anger namnet på FLV-källfilen, minimibandbredden och videofilens varaktighet. Attributet `system-bitrate` stöds bara när taggen <switch> används. I taggen <body> tillåts bara en instans av antingen <switch>, <video> eller <ref>.

Exempel

Följande exempel anger tre FLV-filer, två som använder taggen `video` och en som använder taggen `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>

Tillgänglighet

Flash Professional 8.

Användning

```
<video...attributes.../>
```

Attribut

`src`, `system-bitrate`, `dur`

Underordnade taggar

Ingen.

Överordnad tagg

<body>

Beskrivning

Synonym med taggen <ref>. Stöder attributen `src` och `dur`, som anger namnet på FLV-källfilen och dess varaktighet. Attributet `dur` stöder det fullständiga (00:03:00:01) och det ofullständiga (03:00:01) tidsformatet.

Exempel

Följande exempel ställer in källa och varaktighet för en video:

```
<video src="myvideo_mdm.flv" dur="3:00.1"/>
```

<ref>

Tillgänglighet

Flash Professional 8.

Användning

```
<ref...attributes.../>
```

Attribut

`src`, `system-bitrate`, `dur`

Underordnade taggar

Ingen.

Överordnad tagg

<body>

Beskrivning

Synonym med taggen <video>. Stöder attributen `src` och `dur`, som anger namnet på FLV-källfilen och dess varaktighet. Attributet `dur` stöder det fullständiga (00:03:00:01) och det ofullständiga (03:00:01) tidsformatet.

Exempel

Följande exempel ställer in källa och varaktighet för en video:

```
<ref src="myvideo_cable.flv" dur="3:00.1"/>
```

<switch>

Tillgänglighet

Flash Professional 8.

Använda komponenten FLVPlayback**Användning**

```
<switch>
...
child tags
...
</switch/>
```

Attribut

Ingen.

Underordnade taggar

<video>, <ref>

Överordnad tagg

<body>

Beskrivning

Används med någon av taggarna <video> eller <ref> för att lista FLV-filerna för videoflöden med flera bandbredder. Taggen <switch> stöder attributet `system-bitrate` som anger minimibandbredden och attributen `src` och `dur`.

Exempel

Följande exempel anger tre FLV-filer, två som använder taggen `video` och en som använder taggen `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>
```

Kapitel 7: Använda komponenten FLVPlayback Captioning

Med komponenten FLVPlayback kan du ta med en videospelare i Adobe Flash Professional CS5-programmet för att spela upp hämtade FLV- eller F4V-filer (Adobe Flash Video) och FLV-/F4V-direktuppspelningsfiler. Mer information om FLVPlayback finns i ”[Använda komponenten FLVPlayback](#)” på sidan 134.

Med komponenten FLVPlaybackCaptioning kan du ta med stöd för stängd bildtext för din video. Bildtextkomponenten stöder XML-standardtextformatet Timed Text för W3C och innehåller följande funktioner:

Bildtext med inbäddade händelsereferenspunkter Associera inbäddade händelsereferenspunkter i en FLV-fil med XML för att tillhandahålla bildtext i stället för att använda en TT-XML-fil.

Flera FLVPlayback Captioning-förekomster Skapa flera förekomster av FLVPlayback-bildtexter för flera FLVPlayback-förekomster.

Alternativknappsk kontroll Ger användarinteraktion med bildtext via en alternativknapp för bildtext.

Använda komponenten FLVPlaybackCaptioning

Du använder komponenten FLVPlaybackCaptioning med en eller flera FLVPlayback-komponenter. I det enklaste scenariot drar du en FLVPlayback-komponent till scenen, drar en FLVPlaybackCaptioning-komponent till samma scen, identifierar bildtexternas URL och ställer in bildtexter som ska visas. Dessutom kan du ställa in olika parametrar för att anpassa FLVPlayback-bildtexter.

Lägga till bildtexter i komponenten FLVPlayback

Du kan lägga till komponenten FLVPlaybackCaptioning i alla FLVPlayback-komponenter. Information om att lägga till FLVPlayback-komponenter i programmet finns i ”[Skapa ett program med komponenten FLVPlayback](#)” på sidan 135.

Lägga till komponenten FLVPlaybackCaptioning från panelen Komponenter:

- 1 I panelen Komponenter öppnar du mappen Video.
- 2 Dra (eller dubbelklicka) på komponenten FLVPlaybackCaptioning och lägg till den på samma scen som FLVPlayback-komponenten som du vill lägga till bildtext i.


Obs! Adobe tillhandahåller två exempelfiler för att du snabbt ska kunna lära dig FLVPlaybackCaptioning-komponenten: *caption_video.flv* (ett FLVPlayback-exempel) och *caption_video.xml* (ett bildtextexempel). Hämta dessa filer på www.helpexamples.com/flash/video/caption_video.flv och www.helpexamples.com/flash/video/caption_video.xml.

- 3 (Valfritt) Dra komponenten CaptionButton till samma scen som FLVPlayback- och FLVPlaybackCaptioning-komponenterna. Med komponenten CaptionButton kan användaren aktivera och inaktivera bildtext.

Obs! Om du vill aktivera CaptionButton-komponenten måste du dra den till samma scen som komponenterna FLVPlayback och FLVPlaybackCaptioning.

4 Med komponenten FLVPlaybackCaptioning markerad på scenen ange du följande obligatoriska information på fliken Parametrar i egenskapsinspektören:

- Ställ in värdet `true` för `showCaptions`.
- Ange `source` för TT-XML-filen som ska laddas ned.

 När du arbetar i Flash för att testa bildtexterna ska du ställa in värdet `true` för egenskapen `showCaptions`. Men om du tar med komponenten `CaptionButton` för att användarna ska kunna aktivera och inaktivera bildtext bör du ange värdet `false` för egenskapen `showCaptions`.

Det finns andra parametrar som du kan använda för att anpassa komponenten FLVPlaybackCaptioning. Mer information finns i ”[Anpassa komponenten FLVPlaybackCaptioning](#)” på sidan 180 och i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

5 Välj Kontroll > Testa filmen för att starta videon.

Skapa en instans dynamiskt med ActionScript:

- 1 Dra FLVPlayback-komponenten från panelen Komponenter till bibliotekspanelen (Fönster > Bibliotek).
- 2 Dra FLVPlaybackCaptioning-komponenten från panelen Komponenter till bibliotekspanelen.
- 3 Lägg till följande kod i åtgärdspanelen i den första bildrutan på tidslinjen.

```
import fl.video.*;
var my_FLVPlaybk = new FLVPlayback();
my_FLVPlaybk.x = 100;
my_FLVPlaybk.y = 100;
addChild(my_FLVPlaybk);
my_FLVPlaybk.skin = "install_drive:/Program Files/Adobe/Adobe Flash
CS5/en/Configuration/FLVPlayback Skins/ActionScript 3.0/SkinUnderPlaySeekCaption.swf";
my_FLVPlaybk.source = "http://www.helpexamples.com/flash/video/caption_video.flv";
var my_FLVPlaybkcap = new FLVPlaybackCaptioning();
addChild(my_FLVPlaybkcap);
my_FLVPlaybkcap.source = "http://www.helpexamples.com/flash/video/caption_video.xml";
my_FLVPlaybkcap.showCaptions = true;
```

4 Ändra `install_drive` till den enhet som du installerade Flash på, och ändra sedan sökvägen till sökvägen för mappen Skal i din installation:

Obs! Om du skapar en FLVPlayback-förekomst med ActionScript måste du också tilldela den ett skal dynamiskt genom att ställa in egenskapen `skin` med ActionScript. När du använder ett skal med ActionScript publiceras det inte automatiskt med SWF-filen. Kopiera skal-SWF-filen och program-SWF-filen till din server, annars kommer skal-SWF-filen inte att vara tillgänglig när användaren kör den.

Ställa in parametrar för komponenten FLVPlaybackCaptioning

För varje förekomst av komponenten FLVPlaybackCaptioning kan du ställa in följande parametrar i egenskapsinspektören eller komponentinspektören för att anpassa komponenten ytterligare. I följande lista identifieras egenskaperna och beskrivs kort:

autoLayout Bestämmer om komponenten FLVPlaybackCaptioning styr storleken på bildtextområdet. Standardvärdet är `true`.

captionTargetName Identifierar TextField- eller MovieClip-förekomstnamnet som innehåller bildtexter. Standardvärdet är `auto`.

flvPlaybackName Identifierar det FLVPlayback-förekomstnamn som du vill ge en bildtext. Standardvärdet är `auto`.

simpleFormatting Begränsar formateringsinstruktionerna för TT-XML-filen när värdet är true. Standardvärdet är false.

showCaptions Bestämmer om bildtexter visas. Standardvärdet är true.

källa Anger platsen för TT-XML-filen.

Mer information om alla FLVPlaybackCaptioning-parametrar finns i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

Ange parametern source

Använd parametern `source` för att ange namnet och platsen för TT-XML-filen som innehåller filmens bildtexter. Ange URL-sökvägen direkt i `source`-cellen i komponentinspektören.

Visa bildtexter

Om du vill visa bildtexter ställer du in värdet `true` för parametern `showCaptions`.

Mer information om alla parametrar för FLVPlaybackCaptioning-komponenten finns i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

I de föregående exemplen beskrivs hur du skapar och aktiverar komponenten FLVPlaybackCaptioning för att visa bildtexter. Det finns två källor som du kan använda för bildtexterna: (1) en TT-XML-fil som innehåller dina bildtexter, och (2) en XML-fil med bildtext som du associerar med inbäddade referenspunkter.

Använda Timed Text-bildtexter

Med komponenten FLVPlaybackCaptioning aktiveras bildtexter för den associerade FLVPlayback-komponenten genom att en TT-XML-fil (Timed Text) laddas ned. Mer information om TT-format (Timed Text) finns i informationen om Timed Text för ljud och video på www.w3.org.

I det här avsnittet finns en översikt över de TT-taggar som stöds, taggen för den obligatoriska bildtextfilen och ett exempel på en TT-XML-fil. Detaljerad information om alla TT-taggar som stöds finns i ”[Timed Text-taggar](#)” på sidan 173.

Komponenten FLVPlaybackCaptioning stöder följande TT-taggar:

Kategori	Uppgifter
Stöd för styckeformatering	Justera ett stycke åt höger, vänster eller centrerat
Stöd för textformatering	<ul style="list-style-type: none">Ange storleken på texten med absoluta pixelstorlekar eller delta-stil (till exempel +2, -4)Ställa in textfärg och teckensnittSkapa fetstil och kursiv stilStälla in textjustering
Annat formateringsstöd	<ul style="list-style-type: none">Ställa in bakgrundsfärgen på TextField för bildtexterStälla in bakgrundsfärgen på TextField för bildtexter till genomskinlig (alfa 0)Ställa in automatisk radbrytning i TextField för bildtexter (på eller av)

Komponenten FLVPlaybackCaptioning matchar tidskoden i FLV-filen. Alla bildtexter måste ha ett `begin`-attribut som anger när bildtexten ska visas. Om bildtexten inte har något `dur`- eller `end`-attribut försvinner bildtexten när nästa bildtext visas, eller när FLV-filen slutar.

Här nedan finns ett exempel på en TT-XML-fil. Den här filen (`caption_video.xml`) ger bildtext för filen `caption_video.flv`. Hämta dessa filer på www.helpexamples.com/flash/video/caption_video.flv och www.helpexamples.com/flash/video/caption_video.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
  <tt xml:lang="en"
  xmlns="http://www.w3.org/2006/04/ttaf1"xmlns:tts="http://www.w3.org/2006/04/ttaf1#styling">
  <head>
    <styling>
    <style id="1" tts:textAlign="right"/>
    <style id="2" tts:color="transparent"/>
    <style id="3" style="2" tts:backgroundColor="white"/>
    <style id="4" style="2 3" tts:fontSize="20"/>
    </styling>
  </head>
  <body>
    <div xml:lang="en">
    <p begin="00:00:00.00" dur="00:00:03.07">I had just joined <span
    tts:fontFamily="monospaceSansSerif,proportionalSerif,TheOther"tts:fontSize="+2">Macromedia</span>
    in 1996,</p>
    <p begin="00:00:03.07" dur="00:00:03.35">and we were trying to figure out what to do about the
    internet.</p>
    <p begin="00:00:06.42" dur="00:00:03.15">And the company was in dire straights at the time.</p>
    <p begin="00:00:09.57" dur="00:00:01.45">We were a CD-ROM authoring company,</p>
    <p begin="00:00:11.42" dur="00:00:02.00">and the CD-ROM business was going away.</p>
    <p begin="00:00:13.57" dur="00:00:02.50">One of the technologies I remember seeing was
    Flash.</p>
    <p begin="00:00:16.47" dur="00:00:02.00">At the time, it was called <span
    tts:fontWeight="bold" tts:color="#ccc333">FutureSplash</span>.</p>
    <p begin="00:00:18.50" dur="00:00:01.20">So this is where Flash got its start.</p>
    <p begin="00:00:20.10" dur="00:00:03.00">This is smart sketch running on the <span
    tts:fontStyle="italic">EU-pin computer</span>,</p>
    <p begin="00:00:23.52" dur="00:00:02.00">which was the first product that FutureWave did.</p>
    <p begin="00:00:25.52" dur="00:00:02.00">So our vision for this product was to</p>
    <p begin="00:00:27.52" dur="00:00:01.10">make drawing on the computer</p>
    <p begin="00:00:29.02" dur="00:00:01.30" style="1">as <span tts:color="#ccc333">easy</span>
    as drawing on paper.</p>
    </div>
  </body>
</tt>
```

Timed Text-taggar

Komponenten FLVPlaybackCaptioning stöder Timed Text-taggar för XML-filer med bildtext. Mer information om Timed Text-taggar för ljud och video finns på www.w3.org. I följande tabell finns de taggar som stöds och som inte stöds.

Funktion	Tagg/värde	Användning/beskrivning	Exempel
Ignorerade taggar	metadata	Ignorerad/tillåten på alla nivåer i dokumentet	
	set	Ignorerad/tillåten på alla nivåer i dokumentet	
	xml:lang	Ignorerad	
	xml:space	Ignorerad/beteende åsidosätter till: xml:space="default"	
	layout	Ignorerad/inklusive all region-taggar i ett avsnitt emd layout-tagga	
	br-tagga	Alla attribut och allt innehåll ignoreras.	
Media-timing för bildtexter	begin-attribut	Endast tillåtet i p-taggar. Krävs för mediatidsdistribution av bildtexter.	<p begin="3s">
	dur-attribut	Endast tillåtet i p-taggar. Rekommenderas. Om det inte inkluderas slutar bildtexten med FLV-filen eller när en annan bildtext börjar.	
	end-attribut	Endast tillåtet i p-taggar. Rekommenderas. Om det inte inkluderas slutar bildtexten med FLV-filen eller när en annan bildtext börjar.	
Klock-timing för bildtexter	00:03:00.1	Fullständigt klockformat	
	03:00.1	Ofullständigt klockformat	
	10	Förskjutna tidsperioder utan enheter. Förskjutning representerar sekunder.	
	00:03:00:05 00:03:00:05.1 30f 30t	Stöds inte. Tidsformat som inkluderar ramar eller skalmarkeringar stöds inte.	
Body-tagga	brödtext	Obligatorisk/stöd för endast en body-tagga.	<body><div>...</div></body>
Content-tagga	div-tagga	Noll eller mer tillåtet. Den första taggan används.	
	p-tagga	Noll eller mer tillåtet..	
	span-tagga	En logisk behållare för en sekvens med textuella innehållsenheter. Inget stöd för kapslade span. Stöd för attributstiltaggar.	
	br-tagga	Anger en explicit radbrytning.	
Stiltaggar (Alla stiltaggar används i p-taggen)	style	Refererar ett eller flera stilelement. Kan användas som en tagga och som attribut. Som en tagga krävs ett ID-attribut (stilen kan återanvändas i dokumentet). Stöd för en eller flera stiltaggar inuti stiltaggen.	

Funktion	Tagg/värde	Användning/beskrivning	Exempel
	tts:background Color	Anger en stilegenskap som definierar bakgrundsfärgen för ett område. Alfa ignoreras såvida det inte har värdet noll (alfa 0) för att göra bakgrunden genomskinlig. Färgformatet är #RRGGBBAA.	
	tts:color	Anger en stilegenskap som definierar förgrundsfärgen. Alfa har inte stöd för några färger. Värdet <code>transparent</code> omvandlas till svart.	<pre><style id="3" style="2" tts:backgroundColor="white"/> "transparent" = #00000000 "black"=#000000FF "silver"=#C0C0C0FF "grey"=#808080FF "white"=#FFFFFFFF "maroon"=#800000FF "red"=#FF0000FF "purple"=#800080FF "fuchsia"("magenta")= #FF00FFFF "green"=#008000FF "lime"=#00FF00FF "olive"=#808000FF "yellow"=#FFFF00FF "navy"=#000080FF "blue"=#0000FFFF "teal"=#008080FF "aqua"("cyan")=#00FFFFFF</pre>
	tts:fontFamily	Anger en stilegenskap som definierar teckensnittsfamiljen.	<pre>"default" = _serif "monospace" = _typewriter "sansSerif" = _sans "serif" = _serif "monospaceSansSerif" = _typewriter "monospaceSerif" = _typewriter "proportionalSansSerif" = _sans</pre>
	tts:fontSize	Anger en stilegenskap som definierar teckensnittsstorleken. Bara det första (lodräta) värdet används om två anges. Procentvärden och procentenheter ignoreras. Stöd för storlekar i absolut pixel (till exempel 12) och relativ stil (till exempel +2).	

Funktion	Tagg/värde	Användning/beskrivning	Exempel
	tts:fontStyle	Anger en stilegenskap som definierar teckensnittsstilen.	"normal" "italic" "inherit"* * Standardbeteendet: ärver stilen från omgivande taggar.
	tts:fontWeight	Anger en stilegenskap som definierar teckensnittsbredden.	"normal" "bold" "inherit"* * Standardbeteendet: ärver stilen från omgivande taggar.

Funktion	Tagg/värde	Användning/beskrivning	Exempel
	tts:textAlign	Anger en stilegenskap som definierar hur områden på raden justeras inom ett innehållande blockområde.	"left" "right" "center" "start" ("=left") "end" ("=right") "inherit"* * Ärver stilen från den omgivande taggen. Om ingen textAlign-taggen ställs in är standardvärdet "left".
	tts:wrapOption	Anger en stilegenskap som definierar om automatisk radbrytning gäller i det påverkade elementets kontext. Den här inställningen påverkar alla stycken i bildtextelementet.	"wrap" "noWrap" "inherit"* * Ärver stilen från den omgivande taggen. Om ingen wrapOption-taggen ställs in är standardvärdet "wrap".
Attribut som inte stöds	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		

Använda referenspunkter med bildtexter

Med referenspunkter kan du interagera med en video. Du kan till exempel påverka uppspelningen av en FLV-fil eller visa text vid angivna tidpunkter i videon. Om du inte har någon TT-XML-fil att använda med en FLV-fil kan du bädda in referenspunkter i en FLV-fil och sedan associera dem med text. I det här avsnittet finns information om referenspunktsstandarderna för komponenten FLVPlaybackCaptioning, samt en kort översikt över hur du associerar referenspunkterna med text för bildtexter. Mer information om hur du bäddar in händelsereferenspunkter med guiden Importera video eller Flash Video encoder finns i Kapitel 16, "Arbeta med video" i *Använda Flash*.

Förstå referenspunktsstandarder för FLVPlaybackCaptioning

I FLV-filens metadata representeras en referenspunkt av ett objekt med följande egenskaper: `name`, `time`, `type` och `parameters`. ActionScript-referenspunkter för FLVPlaybackCaptioning har följande attribut:

name Egenskapen `name` är en sträng som innehåller referenspunktens tilldelade namn. Egenskapen `name` måste börja med prefixet `fl.video.caption.2.0.` och prefixet måste följas av en sträng. Strängen är en serie med positiva heltal som ökas stegvis varje gång för att varje namn ska vara unikt. Prefixet innehåller versionsnumret som också matchar versionsnumret för FLVPlayback. För Adobe Flash CS4 och senare måste du ange versionsnumret 2.0.

time Egenskapen `time` är tidpunkten när bildtexten ska visas.

type Egenskapen `type` är en sträng vars värde är "event".

parameters Egenskapen `parameters` är en array som stöder följande par med namn och värde:

- **text:String** Den HTML-formaterade texten för bildtexten. Den här texten skickas till egenskapen `TextField.htmlText` direkt. Komponenter FLVPlaybackCaptioning stöder den valfria egenskapen `text:n` som stöder användningen av flera språkspår. Mer information finns i ”[Stöda flera språkspår med inbäddade referenspunkter](#)” på sidan 180.
- **endTime:Number** Tidpunkten när bildtexten ska försvinna. Om du inte anger den här egenskapen antar FLVPlaybackCaptioning-komponenten att den inte är ett nummer (NaN), och bildtexten visas tills FLV-filen är klar (FLVPlayback-förekomsten skickar `VideoEvent.COMPLETE`-händelsen). Ange egenskapen `endTime:Number` i sekunder.
- **backgroundColor:uint** Den här parametern ställer in `TextField.backgroundColor`. Den här egenskapen är valfri.
- **backgroundColorAlpha:Boolean** Om `backgroundColor` har en alfa på 0 % anger parametern `TextField.background = !backgroundColor`. Den här egenskapen är valfri.
- **wrapOption:Boolean** Den här parametern ställer in `TextField.wordWrap`. Den här egenskapen är valfri.

Förstå händelseinbäddade referenspunkter för bildtext

Om du inte har en Timed Text XML-fil som innehåller bildtexter för FLV-filen kan du skapa bildtexter genom att associera en XML-fil som innehåller bildtext med händelseinbäddade referenspunkter. XML-exemplet förutsätter att du har utfört följande steg för att skapa en händelseinbäddad referenspunkt i videon:

- Lägg till de händelseinbäddade referenspunkterna (enligt standarden för FLVPlaybackCaptioning) och koda videon.
- I Flash drar du en FLVPlayback-komponent och en FLVPlaybackCaptioning-komponent till scenen.
- Ställ in FLVPlayback- och FLVPlaybackCaptioning-komponentens källegenskaper (FLV- och XML-filernas placering).
- Publicera.

Följande exempel importerar XML till kodaren:

```
<?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>
```

Komponenten FLVPlaybackCaptioning stöder också flera språkspår med inbäddad referenspunkt. Mer information finns i ["Stöda flera språkspår med inbäddade referenspunkter"](#) på sidan 180.

Stöda flera språkspår med inbäddade referenspunkter

Egenskapen `track` i `FLVPlaybackCaptioning` stöder flera språkspår med inbäddade referenspunkter, så länge TT-XML-filen följer referenspunktstandarderna för `FLVPlaybackCaptioning`. (Mer information finns i ”[Förstå referenspunktstandarderna för FLVPlaybackCaptioning](#)” på sidan 178.) Med komponenten `FLVPlaybackCaptioning` stöder inte flera språkspår i separata XML-filer. Om du vill använda egenskapen `track` ställer du in ett värde som inte är lika med 0 för egenskapen. Om du till exempel anger värdet 1 för egenskapen `track` (`track == 1`), söker komponenten `FLVPlaybackCaptioning` bland referenspunktparametrarna. Om det inte finns någon matchning används text-egenskapen i referenspunktparametrarna. Mer information finns i avsnittet om egenskapen `track` i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.

Spela upp flera videofiler med bildtext

Du kan öppna flera videospelare i en enda förekomst av `FLVPlayback`-komponenten för att spela upp flera videor och växla mellan dem när de spelas upp. Du kan också associera bildtext med varje videospelare i `FLVPlayback`-komponenten. Mer information om hur du öppnar flera videospelare finns i ”[Använda flera videospelare](#)” på sidan 149. Om du vill använda bildtext i flera videospelare skapar du en förekomst av komponenten `FLVPlaybackCaptioning` för varje `VideoPlayer` och ställer in `FLVPlaybackCaptioning` `videoPlayerIndex` på motsvarande index. Indexet `VideoPlayer` har standardvärdet 0 om det bara finns en `VideoPlayer`.

Nedan följer ett exempel på kod som tilldelar unik bildtext till unika videor. Om du vill köra det här exemplet måste alla exempel-URL i exemplet bytas ut mot fungerande URL:er.

```
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");
```

Anpassa komponenten FLVPlaybackCaptioning

Om du vill börja använda komponenten `FLVPlaybackCaptioning` fort kan du använda standardvärdena för `FLVPlaybackCaptioning`, som placerar bildtexten direkt ovanför `FLVPlayback`-komponenten. Du kan anpassa komponenten `FLVPlaybackCaptioning` så att bildtexten flyttas bort från videon.

Följande kod visar hur du skapar ett `FLVPlayback`-objekt dynamiskt med en alternativknapp för bildtext:

- 1 Placera `FLVPlayback`-komponenten på scenen vid 0,0 och ange förekomstnamnet **player**.
- 2 Placera `FLVPlaybackCaptioning`-komponenten på scenen vid 0,0 och ange förekomstnamnet **captioning**.
- 3 Placera `CaptionButton`-komponenten på scenen.
- 4 I följande kodexempel anger du variabeln `testVideoPath:String` till en FLV-fil (använd en absolut eller relativ sökväg).

Obs! Kodexemplet ställer in variabeln `testVideoPath` till Flash-videoexemplet `caption_video.flv`. Ändra den här variabeln till den sökväg för bildtextvideokomponenten som du lägger till en bildtextknappkomponent i.

- 5 I följande kodexempel anger du variabeln `testCaptioningPath:String` till en lämplig Timed Text XML-fil (använd en absolut eller relativ sökväg).

Obs! Kodexemplet anger variabeln `testCaptioningPath` till TT-XML-filen `caption_video.xml`. Ändra variabeln till sökvägen till TT-XML-filen som innehåller bildtexter till videon.

6 Spara följande kod som `FLVPlaybackCaptioningExample.as` i samma katalog som din FLA-fil.

7 Ställ in `FLVPlaybackCaptioningExample` som `DocumentClass` i FLA-filen.

```
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;
        }
    }
}
```

Mer information om alla `FLVPlaybackCaptioning`-parametrar finns i *Referenshandbok för ActionScript 3.0 i Adobe Flash Professional*.