

Verwenden von ACTIONSCRIPT® 3.0-Komponenten

Rechtliche Hinweise

Rechtliche Hinweise finden Sie unter http://help.adobe.com/de_DE/legalnotices/index.html.

Inhalt

Kapitel 1: Einführung

Zielpublikum	1
Systemanforderungen	1
Informationen zur Dokumentation	2
Typografische Konventionen	2
Begriffserläuterungen	2
Weitere Ressourcen	2

Kapitel 2: Hintergrundinformationen zu ActionScript 3.0-Komponenten

Vorteile von Komponenten	4
Komponentenarten	6
Hinzufügen und Löschen in einem Dokument	8
Ermitteln der Komponentenversion	9
ActionScript 3.0-Ereignisverarbeitungsmodell	10
Einfache Anwendung	11

Kapitel 3: Arbeiten mit Komponenten

Komponenten-Architektur	18
Arbeiten mit Komponentendateien	20
Fehlersuche in Komponentenanwendungen	22
Festlegen von Parametern und Eigenschaften	23
Arbeiten mit der Bibliothek	24
Größe der Komponenten ändern	24
Live-Vorschau	25
Verarbeiten von Ereignissen	25
Arbeiten mit der Anzeigeliste	27
Arbeiten mit dem FocusManager	29
Arbeiten mit List-basierten Komponenten	31
Arbeiten mit einem Datenprovider	31
Arbeiten mit einem Zellrenderer	39
Barrierefreiheit von Komponenten	46

Kapitel 4: Verwenden der UI-Komponenten

Arbeiten mit der Button-Komponente	47
Arbeiten mit der CheckBox-Komponente	50
Arbeiten mit der ColorPicker-Komponente	53
Arbeiten mit der ComboBox-Komponente	56
Arbeiten mit der DataGrid-Komponente	60
Arbeiten mit der Label-Komponente	66
Arbeiten mit der List-Komponente	68
Arbeiten mit der NumericStepper-Komponente	73
Arbeiten mit der ProgressBar-Komponente	76
Arbeiten mit der RadioButton-Komponente	81

Inhalt

Arbeiten mit der ScrollPane-Komponente	84
Arbeiten mit der Slider-Komponente	87
Arbeiten mit der TextArea-Komponente	90
Arbeiten mit der TextInput-Komponente	93
Arbeiten mit der TileList-Komponente	96
Arbeiten mit der UILoader-Komponente	99
Arbeiten mit der UIScrollBar-Komponente	101
Kapitel 5: Anpassen der UI-Komponenten	
Überblick über die Anpassung von UI-Komponenten	104
Festlegen von Stilen	104
Überblick über Skins	107
Anpassen der Button-Komponente	110
Anpassen der CheckBox-Komponente	113
Anpassen der ColorPicker-Komponente	114
Anpassen der ComboBox-Komponente	116
Anpassen der DataGrid-Komponente	118
Anpassen der Label-Komponente	123
Anpassen der List-Komponente	124
Anpassen der NumericStepper-Komponente	127
Anpassen der ProgressBar-Komponente	129
Anpassen der RadioButton-Komponente	130
Anpassen der ScrollPane-Komponente	132
Anpassen der Slider-Komponente	133
Anpassen der TextArea-Komponente	135
Anpassen der TextInput-Komponente	137
Anpassen der TileList-Komponente	138
Anpassen der UILoader-Komponente	140
Anpassen der UIScrollBar-Komponente	141
Kapitel 6: Verwenden der FLVPlayback-Komponente	
Verwenden der FLVPlayback-Komponente	143
Anpassen der FLVPlayback-Komponente	163
Verwenden einer SMIL-Datei	174
Kapitel 7: Verwenden der FLVPlaybackCaptioning-Komponente	
Arbeiten mit der FLVPlaybackCaptioning-Komponente	182
Verwenden von Timed Text-Untertiteln	184
Verwenden von Cue-Points bei der Untertitelung	191
Abspielen mehrerer FLV-Dateien mit Untertiteln	193
Anpassen der FLVPlaybackCaptioning-Komponente	194

Kapitel 1: Einführung

Adobe® Flash® CS5 Professional ist das professionelle Standard-Authoring-Tool für die Erstellung eindrucksvoller Websites. Diese Rich-Internet-Anwendungen basieren auf Komponenten. Bei einer *Komponente* handelt es sich um einen Movieclip, dessen Parameter während der Erstellung in Flash oder zur Laufzeit mit Adobe® ActionScript®-Methoden, -Eigenschaften und -Ereignissen festgelegt werden. Entwickler haben mit Komponenten die Möglichkeit, Code wiederzuverwenden und gemeinsam zu nutzen. Außerdem können komplexe Funktionen eingebunden werden, die von Designern verwendet und individuell angepasst werden können, ohne ActionScript verwenden zu müssen.

Mithilfe von Komponenten können Sie schnell und einfach robuste Anwendungen mit einheitlicher Gestaltungs- und Verhaltensweise erstellen. In diesem Handbuch wird beschrieben, wie Sie mit Adobe ActionScript 3.0-Komponenten Anwendungen erstellen. Im *Komponenten-Referenzhandbuch für Adobe® ActionScript® 3.0* werden die Anwendungsprogrammierschnittstellen (APIs) der einzelnen Komponenten beschrieben.

Sie können von Adobe® entwickelte Komponenten verwenden, Komponenten anderer Entwickler herunterladen oder Ihre eigenen Komponenten erstellen.

Zielpublikum

Dieses Handbuch richtet sich an Entwickler, die Flash-Anwendungen erstellen und mithilfe von Komponenten die Entwicklung beschleunigen möchten. Sie sollten bereits Erfahrung mit der Entwicklung von Flash-Anwendungen und mit dem Schreiben von ActionScript haben.

Wenn Sie mit dem Schreiben von ActionScript-Code weniger vertraut sind, können Sie einem Dokument Komponenten hinzufügen, ihre Parameter im Eigenschafteninspektor oder im Komponenteninspektor festlegen und die entsprechenden Ereignisse im Bedienfeld „Verhalten“ verarbeiten. Beispielsweise könnten Sie einer Button-Komponente für eine Schaltfläche, mit der eine URL in einem Webbrowser geöffnet wird, das Verhalten „Gehe zu Webseite“ zuweisen, ohne ActionScript-Code zu schreiben.

Wenn Sie dagegen robuste Anwendungen programmieren möchten, können Sie Komponenten dynamisch erstellen, ActionScript zum Festlegen von Eigenschaften und Aufrufen von Methoden zur Laufzeit einsetzen und das Ereignis-Listener-Modell zum Verarbeiten von Ereignissen verwenden.

Weitere Informationen finden Sie unter „[Arbeiten mit Komponenten](#)“ auf Seite 18.

Systemanforderungen

Für Flash-Komponenten gelten außer den Flash-Systemanforderungen keine weiteren Systemanforderungen.

Alle SWF-Dateien, die Komponenten aus Flash CS3 oder höher verwenden, müssen mit Adobe® Flash® Player 9.0.28.0 oder höher angezeigt werden. Außerdem müssen sie für ActionScript 3.0 veröffentlicht werden (wählen Sie dazu „Datei“ > „Einstellungen für Veröffentlichungen“, Registerkarte „Flash“).

Informationen zur Dokumentation

In diesem Dokument finden Sie detaillierte Informationen dazu, wie Sie mit Komponenten Flash-Anwendungen entwickeln können. Dabei werden allgemeine Kenntnisse in Flash und ActionScript 3.0 vorausgesetzt. Dokumentation konkret für Flash und zugehörige Produkte steht separat zur Verfügung.

Dieses Dokument ist als PDF-Datei und als Onlinehilfe verfügbar. Zum Anzeigen der Onlinehilfe starten Sie Flash und wählen „Hilfe“ > „Flash-Hilfe“ > „Verwenden von Adobe ActionScript 3.0-Komponenten“.

Weitere Informationen über Flash finden Sie in den folgenden Dokumenten:

- *Verwenden von Flash*
- *ActionScript 3.0 Entwicklerhandbuch*
- *Adobe ActionScript 3.0 Referenzhandbuch für Flash Professional*

Typografische Konventionen

Folgende typografische Konventionen werden in diesem Handbuch verwendet:

- *Kursive Schriftart*: steht für einen zu ersetzenden Wert (z. B. in einem Verzeichnispfad).
- *Codeschriftart*: zeigt ActionScript-Code an (einschließlich Methoden- und Eigenschaftsnamen).
- *Kursive Codeschriftart*: zeigt eine Codekomponente, die ersetzt werden muss (beispielsweise ein ActionScript-Parameter).
- **Fettdruck**: zeigt einen Wert an, den Sie eingeben.

Begriffserläuterungen

Folgende Begriffe werden in diesem Handbuch verwendet:

Laufzeit Dieser Begriff wird verwendet, wenn der Code in Flash Player ausgeführt wird.

Authoring-Zeit Dieser Begriff bezieht sich auf die Arbeit in der Flash-Authoring-Umgebung.

Weitere Ressourcen

Zusätzlich zu den Informationen in diesen Handbüchern stellt Adobe im Adobe Developer Center und im Adobe Design Center regelmäßig aktualisierte Artikel, Entwurfsideen und Beispiele bereit.

Weitere Komponentenbeispiele finden Sie unter www.adobe.com/go/learn_fl_samples_de.

Adobe Developer Center

Das Adobe Developer Center ist Ihre Quelle für aktuelle Informationen zu ActionScript, Artikel über Anwendungsentwicklung aus der Praxis und Informationen über wichtige neue Themen. Besuchen Sie das Developer Center unter www.adobe.com/go/flash_devcenter_de.

Adobe Design Center

Hier finden Sie die neuesten Informationen zu digitalem Design und animierten Grafiken. Sie können die Arbeiten führender Künstler einsehen, neue Entwurfstrends entdecken und Ihre eigenen Fertigkeiten mit Tutorials, wichtigen Workflows und fortgeschrittenen Techniken verbessern. Prüfen Sie zweimal pro Monat, ob neue Tutorials und Artikel sowie inspirierende Galerieeinträge vorliegen. Besuchen Sie das Design Center unter www.adobe.com/go/fl_designcenter_de.

Kapitel 2: Hintergrundinformationen zu ActionScript 3.0-Komponenten

Bei Adobe® Flash® Professional CS5-Komponenten handelt es sich um Movieclips mit Parametern, mit denen Sie die Darstellung und das Verhalten des Movieclips ändern können. Bei einer Komponente kann es sich um ein einfaches Element der Benutzeroberfläche handeln, wie ein Optionsfeld oder ein Kontrollkästchen. Eine Komponente kann aber auch Inhalt enthalten, wie bei den List- und DataGrid-Komponenten.

Mithilfe von Komponenten können Sie schnell und einfach robuste Flash-Anwendungen mit einheitlicher Gestaltungs- und Verhaltensweise erstellen. Anstatt benutzerdefinierte Schaltflächen, Kombinationsfelder und Listen manuell zu erstellen, können Sie diese Elemente mithilfe von Flash-Komponenten implementieren. Ziehen Sie sie einfach aus dem Bedienfeld „Komponenten“ in das Anwendungsdokument. Außerdem können Sie die Darstellung und das Verhalten der Komponenten ganz einfach an Ihr Anwendungsdesign anpassen.

Alle diese Aufgaben lassen sich ohne fundierte ActionScript-Kenntnisse erledigen. Sie können ActionScript 3.0 aber auch verwenden, um das Verhalten einer Komponente zu ändern oder um ein neues Verhalten zu implementieren. Jede Komponente verfügt über einen eigenen Satz von ActionScript-Methoden, -Eigenschaften und -Ereignissen, welche die API (Application Programming Interface, Anwendungsprogrammierschnittstelle) der Komponente bilden. Mithilfe der API können Sie Komponenten erstellen und bearbeiten, während die Anwendung ausgeführt wird.

Die API bietet Ihnen auch die Möglichkeit, benutzerdefinierte Komponenten völlig neu zu erstellen. Von der Adobe Exchange-Website unter www.adobe.com/go/flash_exchange_de können Sie außerdem Komponenten herunterladen, die von Mitgliedern der Flash-Community erstellt wurden. Informationen zum Erstellen von Komponenten finden Sie unter www.adobe.com/go/learn_fl_creating_components_de.

Zur ActionScript 3.0-Komponentenarchitektur gehören Klassen, auf denen alle Komponenten basieren, Skins und Stile zum Anpassen der Darstellung, ein Ereignisverarbeitungsmodell, die Fokusverwaltung, eine Eingabehilfen-Schnittstelle und weitere Merkmale.

Hinweis: Adobe Flash CS5 umfasst ActionScript 2.0-Komponenten und ActionScript 3.0-Komponenten. Diese beiden Komponentensätze können nicht kombiniert werden. Sie müssen für jede Anwendung entweder den einen oder den anderen Komponentensatz verwenden. Flash CS5 zeigt entweder ActionScript 2.0-Komponenten oder ActionScript 3.0-Komponenten, je nachdem, ob Sie eine ActionScript 2.0-Datei oder eine ActionScript 3.0-Datei öffnen. Wenn Sie ein neues Flash-Dokument erstellen, müssen Sie entweder eine Flash-Datei (ActionScript 3.0) oder eine Flash-Datei (ActionScript 2.0) auswählen. Beim Öffnen eines vorhandenen Dokuments bestimmt Flash anhand der Einstellungen für die Veröffentlichung, welcher Komponentensatz verwendet wird. Weitere Informationen über ActionScript 2.0-Komponenten finden Sie in der Dokumentation „Adobe® ActionScript® 2.0-Komponenten verwenden“.

Eine ausführliche Liste der Flash ActionScript 3.0-Komponenten finden Sie unter „[Komponentenarten](#)“ auf Seite 6.

Vorteile von Komponenten

Mit Komponenten können Sie Design und Kodierung voneinander trennen. Komponenten ermöglichen Entwicklern die Erstellung von Funktionen, die Designer in Anwendungen verwenden können. Entwickler können häufig verwendete Funktionen in Komponenten einbinden und Designer können Größe, Position und Verhalten der Komponenten anpassen, indem sie die Parameter ändern. Außerdem können sie die Darstellung einer Komponente ändern, indem sie ihre grafischen Elemente oder Skins bearbeiten.

Komponenten verfügen über wichtige gemeinsame Funktionen, wie etwa Stile, Skins und Fokusverwaltung. Wenn Sie einer Anwendung die erste Komponente hinzufügen, machen diese gemeinsamen Funktionen ungefähr 20 KB der Größe aus. Weitere Komponenten, die Sie hinzufügen, nutzen denselben zugewiesenen Speicherplatz, sodass eine unnötige Zunahme der Anwendungsgröße vermieden wird.

In diesem Abschnitt werden einige Vorteile von ActionScript 3.0-Komponenten beschrieben.

Leistungsfähige ActionScript 3.0-Funktionalität bietet eine vielseitige, objektorientierte Programmiersprache, die bei der Entwicklung der Flash Player-Funktionen einen wichtigen Meilenstein darstellt. Mithilfe dieser Programmiersprache lassen sich Rich-Internet-Anwendungen auf Grundlage von wiederverwendbarem Code erstellen. ActionScript 3.0 basiert auf ECMAScript, der international standardisierten Skriptsprache, und entspricht der Sprachspezifikation ECMAScript (ECMA-262) Version 3. Eine umfassende Einführung in ActionScript 3.0 finden Sie im *ActionScript 3.0-Entwicklerhandbuch*. Referenzinformationen zur Sprache finden Sie im [ActionScript 3.0 Referenzhandbuch](#).

FLA-basierte Benutzeroberflächenkomponenten ermöglichen den einfachen Zugriff auf Skins und damit eine einfache Anpassung zur Authoring-Zeit. Diese Komponenten bieten auch Stile, darunter Skinstile, mit denen Sie das Erscheinungsbild der Komponente anpassen und zur Laufzeit Skins laden können. Weitere Informationen finden Sie unter „[Anpassen der UI-Komponenten](#)“ auf Seite 104 und im [ActionScript 3.0 Referenzhandbuch](#).

Neue FLVPlayback-Komponente integriert FLVPlaybackCaptioning sowie Vollbildmodus-Unterstützung, eine verbesserte Live-Vorschau, Skins zum Hinzufügen von Farb- und Alpha-Einstellungen sowie verbesserte Funktionen für FLV-Download und Layout.

Der Eigenschafteninspektor und der Komponenten-Inspektor ermöglichen die Änderung von Komponentenparametern beim Authoring in Flash. Weitere Informationen finden Sie unter „[Arbeiten mit Komponentendateien](#)“ auf Seite 20 und „[Festlegen von Parametern und Eigenschaften](#)“ auf Seite 23.

Ein neues Sammlungsdialogfeld für die ComboBox-, List- und TileList-Komponenten ermöglicht das Ausfüllen ihrer `dataProvider`-Eigenschaft über die Benutzeroberfläche. Weitere Informationen finden Sie unter „[Erstellen eines Datenproviders](#)“ auf Seite 31.

Über das ActionScript 3.0-Ereignismodell können Anwendungen auf Ereignisse warten und als Reaktion Ereignisprozeduren aufrufen. Weitere Informationen finden Sie unter „[ActionScript 3.0-Ereignisverarbeitungsmodell](#)“ auf Seite 10 und „[Verarbeiten von Ereignissen](#)“ auf Seite 25.

Managerklassen bieten eine einfache Möglichkeit zur Fokus- und Stilverwaltung in einer Anwendung. Weitere Informationen finden Sie im [ActionScript 3.0 Referenzhandbuch](#).

Die UIComponent-Basisklasse bietet Kernmethoden, -eigenschaften und -ereignisse für Komponenten, die sie erweitern. Alle ActionScript 3.0-Benutzeroberflächenkomponenten erben von der UIComponent-Klasse. Weitere Informationen finden Sie im Abschnitt zur UIComponent-Klasse im [ActionScript 3.0 Referenzhandbuch](#).

Die Verwendung einer SWC in FLA-basierten Benutzeroberflächenkomponenten bietet ActionScript-Definitionen als Element in der Zeitleiste der Komponente, um die Kompilierung zu beschleunigen.

Eine leicht erweiterbare Klassenhierarchie ermöglicht es Ihnen, mit ActionScript 3.0 eigene Namespaces zu erstellen, Klassen bei Bedarf zu importieren und Komponenten durch Unterklassen zu erweitern.

Weitere Informationen finden Sie im [ActionScript 3.0 Referenzhandbuch](#).

Hinweis: Flash CS5 unterstützt sowohl FLA-basierte als auch SWC-basierte Komponenten. Weitere Informationen finden Sie unter „[Komponenten-Architektur](#)“ auf Seite 18.

Komponentenarten

Sie installieren die Flash-Komponenten, wenn Sie Flash CS5 installieren.

Zu den ActionScript 3.0-Komponenten zählen die folgenden UI-Komponenten (Benutzeroberflächenkomponenten):

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

Zusätzlich zu den UI-Komponenten umfasst Flash ActionScript 3.0 die folgenden Komponenten und zugehörigen Klassen:

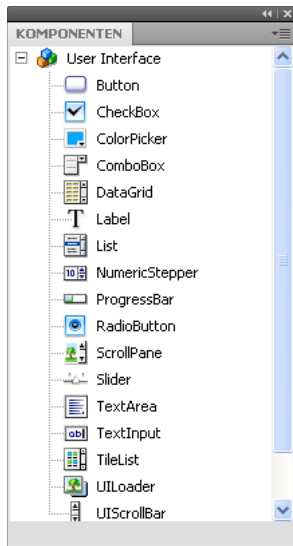
- FLVPlayback-Komponente (fl.video.FLVPlayback), eine SWC-basierte Komponente.
Mit der FLVPlayback-Komponente können Sie einen Video-Player in Ihre Flash-Anwendung einschließen, um progressives Streaming-Video über HTTP, von einem Adobe® Flash® Video Streaming Service (FVSS) oder vom Adobe Macromedia® Flash® Media Server (FMS) abzuspielen. Weitere Informationen finden Sie unter „[Verwenden der FLVPlayback-Komponente](#)“ auf Seite 143.
- Die FLVPlayback Custom UI-Komponenten, die auf FLA basieren und sowohl mit der ActionScript 2.0-Version als auch mit der ActionScript 3.0-Version der FLVPlayback-Komponente verwendet werden können. Weitere Informationen finden Sie unter „[Verwenden der FLVPlayback-Komponente](#)“ auf Seite 143.
- Die FLVPlayback Captioning-Komponente, die optionale Untertitel für FLVPlayback bereitstellt. Einzelheiten finden Sie unter „[Verwenden der FLVPlaybackCaptioning-Komponente](#)“ auf Seite 182.

Eine vollständige Liste der ActionScript 3.0-Komponenten und der zugehörigen Klassen finden Sie im [ActionScript 3.0 Referenzhandbuch](#).

Anzeigen der Flash-Komponenten

Sie können die Flash ActionScript 3.0-Komponenten im Bedienfeld „Komponenten“ anzeigen, indem Sie die folgenden Schritte ausführen.

- 1 Starten Sie Flash.
- 2 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0) oder öffnen Sie ein vorhandenes Flash-Dokument, in dessen Einstellungen für Veröffentlichungen ActionScript 3.0 festgelegt wurde.
- 3 Wählen Sie „Fenster“ > „Komponenten“, um das Bedienfeld „Komponenten“ aufzurufen.



Bedienfeld „Komponenten“ mit Benutzeroberflächenkomponenten

Sie können weitere Komponenten auch von der Adobe Exchange-Website unter www.adobe.com/go/flash_exchange_de herunterladen. Wenn Sie von dieser Website heruntergeladene Komponenten installieren möchten, müssen Sie den Adobe® Extension Manager von www.adobe.com/go/exchange_de herunterladen und installieren. Klicken Sie dazu auf den Link „EXTENSION MANAGER“.

Jede Komponente kann im Bedienfeld „Komponenten“ in Flash angezeigt werden. Im Folgenden wird beschrieben, wie Sie Komponenten auf einem Windows®- oder Macintosh®-Computer installieren.

Installieren von Komponenten auf einem Windows- oder Macintosh-Computer

- 1 Beenden Sie Flash.
- 2 Legen Sie die SWC- oder FLA-Datei, in der die Komponente enthalten ist, im folgenden Ordner auf Ihrer Festplatte ab.
 - Windows:
C:\Programme\Adobe\Adobe Flash CS5\Sprache\Configuration\Components
 - Macintosh:
Macintosh HD:Applications:Adobe Flash CS5:Configuration:Components
- 3 Starten Sie Flash.
- 4 Wählen Sie „Fenster“ > „Komponenten“, um die Komponente im Bedienfeld „Komponenten“ anzuzeigen, falls dieses Bedienfeld nicht bereits geöffnet ist.

Weitere Informationen über Komponentendateien finden Sie unter „[Arbeiten mit Komponentendateien](#)“ auf Seite 20.

Hinzufügen und Löschen in einem Dokument

Wenn Sie eine FLA-basierte Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne ziehen, importiert Flash einen bearbeitbaren Movieclip in die Bibliothek. Wenn Sie eine SWC-basierte Komponente auf die Bühne ziehen, importiert Flash einen kompilierten Clip in die Bibliothek. Nachdem eine Komponente in die Bibliothek importiert wurde, können Sie Instanzen davon entweder aus dem Bedienfeld „Bibliothek“ oder aus dem Bedienfeld „Komponenten“ auf die Bühne ziehen.

Hinzufügen von Komponenten zur Authoring-Zeit

Sie können einem Dokument eine Komponente hinzufügen, indem Sie sie aus dem Bedienfeld „Komponenten“ ziehen. Die Eigenschaften der einzelnen Instanzen einer Komponente werden im Eigenschafteninspektor oder auf der Registerkarte „Parameter“ im Komponenten-Inspektor eingestellt.

- 1 Wählen Sie „Fenster“ > „Komponenten“.
- 2 Doppelklicken Sie im Bedienfeld „Komponenten“ auf die Komponente oder ziehen Sie die Komponente auf die Bühne.
- 3 Wählen Sie die Komponente auf der Bühne aus.
- 4 Sollte der Eigenschafteninspektor nicht zu sehen sein, wählen Sie „Fenster“ > „Eigenschaften“ > „Eigenschaften“.
- 5 Geben Sie im Eigenschafteninspektor einen Instanznamen für die Komponenteninstanz ein.
- 6 Wählen Sie „Fenster“ > „Komponenten-Inspektor“ und klicken Sie auf die Registerkarte „Parameter“, um die Parameter für die Instanz anzugeben.

Weitere Informationen finden Sie unter [„Festlegen von Parametern und Eigenschaften“](#) auf Seite 23.

- 7 Ändern Sie bei Bedarf die Größe der Komponente, indem Sie die Werte für Breite (B) und Höhe (H) ändern.
Weitere Informationen zum Einstellen der Größe für bestimmte Komponententypen finden Sie unter [„Anpassen der UI-Komponenten“](#) auf Seite 104.
- 8 Wählen Sie „Steuerung“ > „Film testen“ oder drücken Sie Strg+Eingabe, um das Dokument zu kompilieren und die Ergebnisse Ihrer Einstellungen zu sehen.

Sie können auch die Farbe und die Textformatierung für eine Komponente über Stileigenschaften ändern. Außerdem können Sie die Darstellung einer Komponente anpassen, indem Sie ihre Skins ändern. Weitere Informationen zu diesen Themen finden Sie unter [„Anpassen der UI-Komponenten“](#) auf Seite 104.

Wenn Sie während des Authorings eine Komponente auf die Bühne ziehen, können Sie über den Instanznamen auf die Komponente verweisen (beispielsweise `myButton`).

Hinzufügen von Komponenten zur Laufzeit mit ActionScript

Damit Sie einem Dokument zur Laufzeit mit ActionScript eine Komponente hinzufügen können, muss die Komponente in der Bibliothek der Anwendung vorhanden sein („Fenster“ > „Bibliothek“), wenn die SWF-Datei kompiliert wird. Sie können der Bibliothek eine Komponente hinzufügen, indem Sie die Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ ziehen. Weitere Informationen zur Bibliothek finden Sie unter [„Arbeiten mit der Bibliothek“](#) auf Seite 24.

Sie müssen auch die Klassendatei der Komponente importieren, damit die jeweilige API in der Anwendung zur Verfügung steht. Klassendateien von Komponenten sind in *Paketen* installiert, die eine oder mehrere Klassen enthalten. Zum Importieren einer Komponentenkategorie verwenden Sie die `import`-Anweisung und geben den Namen des Pakets sowie den Namen der Klasse an. Die `Button`-Kategorie wird beispielsweise mit der folgenden `import`-Anweisung importiert:

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

Einzelheiten dazu, in welchem Paket eine Komponente enthalten ist, finden Sie im [ActionScript 3.0 Referenzhandbuch](#). Informationen zum Speicherort der Komponentensourcenfiles finden Sie unter „[Arbeiten mit Komponentensourcenfiles](#)“ auf Seite 20.

Um eine Instanz einer Komponente zu erstellen, müssen Sie die ActionScript-Konstruktormethode der Komponente aufrufen. Mit der folgenden Anweisung wird zum Beispiel eine Instanz einer Schaltfläche mit dem Namen `aButton` erstellt:

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

Im letzten Schritt wird die statische `addChild()`-Methode aufgerufen, um die Komponenteninstanz der Bühne oder dem Anwendungscontainer hinzuzufügen. Mit der folgenden Anweisung wird zum Beispiel die Instanz `aButton` hinzugefügt:

```
addChild(aButton);
```

Nun können Sie die API der Komponente verwenden, um dynamisch die Größe und die Bühnenposition der Komponente festzulegen, auf Ereignisse zu warten und das Verhalten mithilfe von Eigenschaften zu ändern. Weitere Informationen zur API für eine bestimmte Komponente finden Sie im [ActionScript 3.0 Referenzhandbuch](#).

Weitere Informationen zur `addChild()`-Methode finden Sie unter „[Arbeiten mit der Anzeigeliste](#)“ auf Seite 27.

Löschen einer Komponente

Wenn Sie zur Authoring-Zeit eine Komponenteninstanz von der Bühne löschen möchten, wählen Sie die Instanz aus und drücken Sie die `Entf`-Taste. Dadurch wird die Instanz von der Bühne entfernt, nicht aber die Komponente aus der Anwendung.

Um eine Komponente aus einem Flash-Dokument zu löschen, nachdem Sie sie auf der Bühne oder in der Bibliothek platziert haben, müssen Sie die Komponente und die dazugehörigen Elemente aus der Bibliothek löschen. Es ist nicht ausreichend, die Komponente von der Bühne zu entfernen. Wenn Sie die Komponente nicht aus der Bibliothek entfernen, ist sie beim Kompilieren noch in der Anwendung enthalten.

- 1 Wählen Sie im Bedienfeld „Bibliothek“ das Symbol der Komponente aus.
- 2 Klicken Sie am unteren Rand des Bedienfelds „Bibliothek“ auf die Schaltfläche „Löschen“ oder wählen Sie im Menü des Bedienfelds „Bibliothek“ den Befehl „Löschen“.

Wiederholen Sie diese Schritte, um alle zugehörigen Elemente der Komponente zu löschen.

Unter „[Entfernen von Komponenten aus der Anzeigeliste](#)“ auf Seite 28 wird beschrieben, wie Sie eine Komponente aus ihrem Container entfernen, während die Anwendung ausgeführt wird.

Ermitteln der Komponentenversion

Flash ActionScript 3.0-Komponenten verfügen über eine `Version`-Eigenschaft, die Sie anzeigen können, wenn sie für den technischen Support von Adobe erforderlich ist oder wenn Sie wissen müssen, welche Version der Komponente Sie verwenden.

Anzeigen der Versionsnummer für eine UI-Komponente

- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie die Komponente auf die Bühne und geben Sie ihr einen Instanznamen. Ziehen Sie beispielsweise eine ComboBox-Komponente auf die Bühne und nennen Sie sie **aCb**.
- 3 Drücken Sie die Taste **F9** oder wählen Sie „Fenster“ > „Aktionen“, um das Bedienfeld „Aktionen“ zu öffnen.
- 4 Klicken Sie auf Bild 1 der Hauptzeitleiste und fügen Sie dem Bedienfeld „Aktionen“ den folgenden Code hinzu:

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

Die Versionsnummer sollte ähnlich wie in der folgenden Abbildung im Bedienfeld „Ausgabe“ angezeigt werden.

Für die FLVPlayback- und FLVPlaybackCaptioning-Komponenten müssen Sie auf den Klassennamen anstatt auf den Instanznamen zurückgreifen, da die Versionsnummer in einer Klassenkonstante gespeichert wird.

Anzeigen der Versionsnummer der FLVPlayback- und FLVPlaybackCaptioning-Komponenten

- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie die FLVPlayback- und FLVPlaybackCaptioning-Komponenten in das Bedienfeld „Bibliothek“.
- 3 Drücken Sie die Taste **F9** oder wählen Sie „Fenster“ > „Aktionen“, um das Bedienfeld „Aktionen“ zu öffnen.
- 4 Klicken Sie auf Bild 1 der Hauptzeitleiste und fügen Sie dem Bedienfeld „Aktionen“ den folgenden Code hinzu:

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

Die Versionsnummern werden im Bedienfeld „Ausgabe“ angezeigt.

ActionScript 3.0-Ereignisverarbeitungsmodell

Mit ActionScript 3.0 wird ein einziges Ereignisverarbeitungsmodell eingeführt, das die unterschiedlichen Ereignisverarbeitungsmechanismen der ActionScript-Vorgängerversionen ersetzt. Das neue Ereignismodell beruht auf der DOM3-Ereignisspezifikation (Document Object Model Level 3).

Für Entwickler, die bereits Erfahrungen mit der `addListener()`-Methode in ActionScript 2.0 gesammelt haben, ist eine Übersicht der Unterschiede zwischen dem Ereignis-Listener-Modell von ActionScript 2.0 und dem Ereignismodell von ActionScript 3.0 hilfreich. Im Folgenden sind daher einige der wichtigsten Unterschiede zwischen den beiden Ereignismodellen aufgeführt:

- Zum Hinzufügen von Ereignis-Listnern in ActionScript 2.0 wird in einigen Fällen `addListener()` und in anderen `addEventListener()` verwendet. In ActionScript 3.0 wird in allen Fällen `addEventListener()` eingesetzt.
- In ActionScript 2.0 gibt es keinen Ereignisablauf, d. h., die `addListener()`-Methode kann nur für das Objekt aufgerufen werden, das das Ereignis überträgt. Im Gegensatz dazu kann in ActionScript 3.0 die `addEventListener()`-Methode für jedes Objekt aufgerufen werden, das Bestandteil des Ereignisablaufs ist.
- In ActionScript 2.0 können Ereignis-Listener entweder Funktionen, Methoden oder Objekte sein, während in ActionScript 3.0 als Ereignis-Listener nur Funktionen oder Methoden möglich sind.
- Die Syntax `on(event)` wird in ActionScript 3.0 nicht mehr unterstützt, sodass Sie einem Movieclip keinen ActionScript-Ereigniscode anhängen können. Sie können nur `addEventListener()` verwenden, um einen Ereignis-Listener hinzuzufügen.

Im folgenden Beispiel, in dem das `MouseEvent.CLICK`-Ereignis für die Button-Komponente `aButton` abgefragt wird, wird das grundlegende Ereignisverarbeitungsmodell von ActionScript 3.0 veranschaulicht:

```
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);  
}
```

Weitere Informationen zur Ereignisverarbeitung in ActionScript 3.0 finden Sie unter *Programmieren mit ActionScript 3.0*. Einzelheiten zur ActionScript 3.0-Ereignisverarbeitung für Komponenten finden Sie unter „[Verarbeiten von Ereignissen](#)“ auf Seite 25.

Einfache Anwendung

In diesem Abschnitt wird Schritt für Schritt erläutert, wie Sie mit den Flash-Komponenten und dem Flash-Authoring-Tool eine einfache ActionScript 3.0-Anwendung erstellen. Das Beispiel steht als FLA-Datei mit dem ActionScript-Code in der Zeitleiste und auch als externe ActionScript-Klassendatei mit einer FLA-Datei, die nur die Komponenten in der Bibliothek enthält, zur Verfügung. In der Regel empfiehlt es sich, größere Anwendungen mithilfe von externen Klassendateien zu erstellen, damit Sie den Code zwischen Klassen und Anwendungen wieder verwenden und Ihre Anwendungen einfacher verwalten können. Weitere Informationen zur Programmierung mit ActionScript 3.0 finden Sie unter *Programmieren mit ActionScript 3.0*.

Anwendungsdesign

Das erste Beispiel einer ActionScript-Komponentenanwendung ist eine Abwandlung der Standardanwendung „Hello World“ und weist deshalb ein relativ einfaches Design auf.

- Die Anwendung wird „Greetings“ heißen.
- In einer TextArea-Komponente wird eine Begrüßung angezeigt, die zunächst „Hello World“ lautet.
- Über eine ColorPicker-Komponente kann die Textfarbe geändert werden.
- Die Anwendung enthält drei RadioButton-Komponenten, über die Sie die Textgröße auf klein (small), größer (larger) oder am größten (largest) einstellen können.
- Über eine ComboBox-Komponente kann eine andere Begrüßung in einer Dropdownliste ausgewählt werden.
- Für die Anwendung werden Komponenten aus dem Bedienfeld „Komponenten“ verwendet und es werden Anwendungselemente mithilfe von ActionScript-Code erstellt.

Anhand dieser Kurzbeschreibung können Sie mit dem Erstellen der Anwendung beginnen.

Erstellen der Greetings-Anwendung

In den folgenden Schritten wird die Greetings-Anwendung mit dem Flash-Authoring-Tool erstellt. Sie erstellen die FLA-Datei, platzieren Komponenten auf der Bühne und fügen der Zeitleiste ActionScript-Code hinzu.

Erstellen der Greetings-Anwendung in einer FLA-Datei:

- 1 Wählen Sie „Datei“ > „Neu“.
- 2 Wählen Sie im Dialogfeld „Neues Dokument“ die Option „Flash-Datei (ActionScript 3.0)“ aus und klicken Sie dann auf „OK“.

Ein neues Flash-Fenster wird geöffnet.

- 3 Wählen Sie „Datei“ > „Speichern“, nennen Sie die Flash-Datei **Greetings fla** und klicken Sie auf „Speichern“.
- 4 Wählen Sie im Flash-Bedienfeld „Komponenten“ eine TextArea-Komponente aus und ziehen Sie sie auf die Bühne.
- 5 Wählen Sie die TextArea-Komponente auf der Bühne aus und geben Sie dann im Fenster „Eigenschaften“ den Instanznamen **aTa** sowie die folgenden Informationen ein:
 - **230** für den Wert „B“ (Breite).
 - **44** für den Wert „H“ (Höhe).
 - **165** für den Wert „X“ (horizontale Position).
 - **57** für den Wert „Y“ (vertikale Position).
 - Geben Sie auf der Registerkarte „Parameter“ als Textparameter **Hello World!** ein.
- 6 Ziehen Sie eine ColorPicker-Komponente auf die Bühne, platzieren Sie sie links von der TextArea-Komponente und geben Sie ihr den Instanznamen **txtCp**. Geben Sie im Eigenschafteninspektor die folgenden Informationen ein:
 - **96** für den Wert „X“.
 - **72** für den Wert „Y“.
- 7 Ziehen Sie nacheinander drei RadioButton-Komponenten auf die Bühne und geben Sie ihnen die Instanznamen **smallRb**, **largerRb** und **largestRb**. Geben Sie im Eigenschafteninspektor die folgenden Informationen ein:
 - Jeweils **100** für den Wert „B“ und **22** für den Wert „H“.
 - **155** für den Wert „X“.
 - **120** für den Wert „Y“ von **smallRb**, **148** für **largerRb** und **175** für **largestRb**.
 - Jeweils **fontRbGrp** für den groupName-Parameter.
 - Geben Sie auf der Registerkarte „Parameter“ die Bezeichnungen **Small**, **Larger**, **Largest** ein.
- 8 Ziehen Sie eine ComboBox-Komponente auf die Bühne und geben Sie ihr den Instanznamen **msgCb**. Geben Sie im Eigenschafteninspektor die folgenden Informationen ein:
 - **130** für den Wert „B“.
 - **265** für den Wert „X“.
 - **120** für den Wert „Y“.
 - Geben Sie auf der Registerkarte „Parameter“ als prompt-Parameter **Greetings** ein.
 - Doppelklicken Sie auf das Textfeld für den dataProvider-Parameter, um das Dialogfeld „Werte“ zu öffnen.
 - Klicken Sie auf das Pluszeichen und ersetzen Sie den label-Wert durch **Hello World!**
 - Wiederholen Sie den vorigen Schritt und fügen Sie diese Bezeichnungen hinzu: **Have a nice day!** und **Top of the Morning!**
 - Klicken Sie auf „OK“, um das Dialogfeld „Werte“ zu schließen.
- 9 Speichern Sie die Datei.
- 10 Falls das Bedienfeld „Aktionen“ noch nicht angezeigt wird, öffnen Sie es, indem Sie **F9** drücken oder „Fenster“ > „Aktionen“ wählen. Klicken Sie auf Bild 1 der Hauptzeitleiste und geben Sie im Bedienfeld „Aktionen“ den folgenden Code ein:


```
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);
```

Mit den ersten drei Zeilen werden die von der Anwendung verwendeten Ereignisklassen importiert. Ein Ereignis tritt bei einer Benutzerinteraktion mit einer der Komponenten auf. Mit den nächsten fünf Zeilen werden die Ereignisprozeduren für die Ereignisse registriert, auf die die Anwendung wartet. Ein `click`-Ereignis für eine `RadioButton`-Komponente tritt auf, wenn der Benutzer auf die Komponente klickt. Ein `change`-Ereignis tritt auf, wenn der Benutzer eine andere Farbe in der `ColorPicker`-Komponente auswählt. Ein `change`-Ereignis für eine `ComboBox`-Komponente tritt auf, wenn der Benutzer eine andere Begrüßung in der Dropdownliste auswählt.

Mit der vierten Zeile wird die `RadioButtonGroup`-Klasse importiert, damit die Anwendung die Ereignisprozedur einer `RadioButton`-Gruppe anstatt den einzelnen Optionsfeldern zuweisen kann.

- 11 Fügen Sie dem Bedienfeld „Aktionen“ die folgende Codezeile hinzu, um das `TextFormat`-Objekt `tf` zu erstellen, das die Anwendung verwendet, um die Stileigenschaften `size` und `color` für den Text der `TextArea`-Komponente zu ändern.

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

- 12 Fügen Sie den folgenden Code hinzu, um die `rbHandler`-Ereignisverarbeitungsfunktion zu erstellen. Diese Funktion verarbeitet ein `click`-Ereignis, wenn der Benutzer auf eine der `RadioButton`-Komponenten klickt.

```
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);
}
```

Die Funktion überprüft über eine `switch`-Anweisung die `target`-Eigenschaft des `event`-Objekts, um festzustellen, welche `RadioButton`-Komponente das Ereignis ausgelöst hat. Die `currentTarget`-Eigenschaft enthält den Namen des Objekts, das das Ereignis ausgelöst hat. Je nachdem, auf welches Optionsfeld der Benutzer geklickt hat, wird die Größe des Textes in der `TextArea`-Komponente in 14, 18 oder 24 Punkt geändert.

- 13 Fügen Sie den folgenden Code ein, um die `cpHandler()`-Funktion zu implementieren, die eine Änderung des Wertes in der `ColorPicker`-Komponente verarbeitet:

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

Diese Funktion setzt die `color`-Eigenschaft des `TextFormat`-Objekts `tf` auf die in der `ColorPicker`-Komponente ausgewählte Farbe und ruft dann `setStyle()` auf, um die Farbe auf den Text in der `TextArea`-Instanz `aTa` anzuwenden.

- 14 Fügen Sie den folgenden Code ein, um die `cbHandler()`-Funktion zu implementieren, die eine Änderung der Auswahl in der `ComboBox`-Komponente verarbeitet:

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

Diese Funktion ersetzt den Text in der `TextArea`-Komponente durch den ausgewählten Text in der `ComboBox`-Komponente, `event.target.selectedItem.label`.

- 15 Wählen Sie „Steuerung“ > „Film testen“ oder drücken Sie Strg+Eingabe, um den Code zu kompilieren und die `Greetings`-Anwendung zu testen.

Im folgenden Abschnitt wird beschrieben, wie Sie dieselbe Anwendung mit einer externen `ActionScript`-Klasse und einer `FLA`-Datei erstellen, die nur über die erforderlichen Komponenten in der Bibliothek verfügt.

Erstellen der `Greetings2`-Anwendung mit einer externen Klassendatei:

- 1 Wählen Sie „Datei“ > „Neu“.
- 2 Wählen Sie im Dialogfeld „Neues Dokument“ die Option „Flash-Datei (ActionScript 3.0)“ aus und klicken Sie dann auf „OK“.
Ein neues `Flash`-Fenster wird geöffnet.
- 3 Wählen Sie „Datei“ > „Speichern“, nennen Sie die `Flash`-Datei **Greetings2.fla** und klicken Sie auf „Speichern“.
- 4 Ziehen Sie jede der folgenden Komponenten aus dem Bedienfeld „Komponenten“ in die Bibliothek:
 - `ColorPicker`
 - `ComboBox`
 - `RadioButton`
 - `TextArea`Die kompilierte `SWF`-Datei verwendet jedes dieser Elemente, deshalb müssen Sie sie der Bibliothek hinzufügen. Ziehen Sie die Komponenten in den unteren Bereich des Bedienfelds „Bibliothek“. Wenn Sie diese Komponenten der Bibliothek hinzufügen, werden andere Elemente (wie `List`, `TextInput` und `UI ScrollBox`) automatisch hinzugefügt.
- 5 Geben Sie im Fenster „Eigenschaften“ für die Dokumentklasse **Greetings2** ein.
Wenn `Flash` eine Warnung anzeigt, dass für die Dokumentklasse keine Definition gefunden wurde, ignorieren Sie sie. Sie definieren die `Greetings2`-Klasse in den folgenden Schritten. Diese Klasse definiert die Hauptfunktionalität für die Anwendung.
- 6 Speichern Sie die Datei „Greetings2.fla“.
- 7 Wählen Sie „Datei“ > „Neu“.
- 8 Wählen Sie im Dialogfeld „Neues Dokument“ die Option „ActionScript-Datei“ aus und klicken Sie dann auf „OK“.
Ein neues Skriptfenster wird geöffnet.
- 9 Fügen Sie den folgenden Code im Skriptfenster hinzu:

```
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() {
```

Das Skript definiert eine ActionScript 3.0-Klasse namens „Greetings2“. Das Skript führt folgende Aufgaben aus:

- Es importiert Klassen, die in der Datei verwendet werden. Normalerweise fügen Sie diese `Import`-Anweisungen beim Verweis auf unterschiedliche Klassen im Code hinzu, in diesem Beispiel werden sie jedoch der Einfachheit halber in einem Schritt importiert.
- Es deklariert Variablen für die verschiedenen Arten von Komponentenobjekten, die Sie dem Code hinzufügen werden. Eine andere Variable erstellt das `TextFormat`-Objekt `tf`.
- Es definiert die Konstrukturfunktion `Greetings2()` für die Klasse. In den folgenden Schritten fügen Sie dieser Funktion Zeilen hinzu. Außerdem fügen Sie der Klasse weitere Methoden hinzu.

10 Wählen Sie „Datei“ > „Speichern“, nennen Sie die Datei **Greetings2.as** und klicken Sie auf „Speichern“.

11 Fügen Sie der Funktion `Greeting2()` die folgenden Codezeilen hinzu:

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

Die Funktion sollte nun folgendermaßen aussehen:

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

12 Fügen Sie hinter der schließenden Klammer der Methode `Greeting2()` die folgenden Codezeilen hinzu:

```
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);
}
```

Diese Zeilen haben folgende Wirkung:

- Die in der Anwendung verwendeten Komponenten werden instanziiert.
- Größe, Position und Eigenschaften der einzelnen Komponenten werden eingestellt.

- Jede Komponente wird mithilfe der `addChild()`-Methode der Bühne hinzugefügt.

13 Fügen Sie nach der schließenden Klammer der `buildRadioButtons()`-Methode den folgenden Code für die `setUpHandlers()`-Methode hinzu:

```
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;
}
}
```

Diese Funktionen definieren Ereignis-Listener für die Komponenten.

14 Wählen Sie „Datei“ > „Speichern“, um die Datei zu speichern.

15 Wählen Sie „Steuerung“ > „Film testen“ oder drücken Sie Strg+Eingabe, um den Code zu kompilieren und die Anwendung „Greetings2“ zu testen.

Entwickeln und Ausführen von weiteren Beispielen

Nachdem Sie die Greetings-Anwendung entwickelt und ausgeführt haben, verfügen Sie nun über die erforderlichen Grundkenntnisse, um auch die anderen Codebeispiele in diesem Handbuch auszuführen. Der relevante ActionScript 3.0-Code in jedem Beispiel wird hervorgehoben und erläutert. Sie sollten in der Lage sein, die einzelnen Codebeispiele in diesem Handbuch in eine FLA-Datei einzufügen und diese Datei zu kompilieren und auszuführen.

Kapitel 3: Arbeiten mit Komponenten

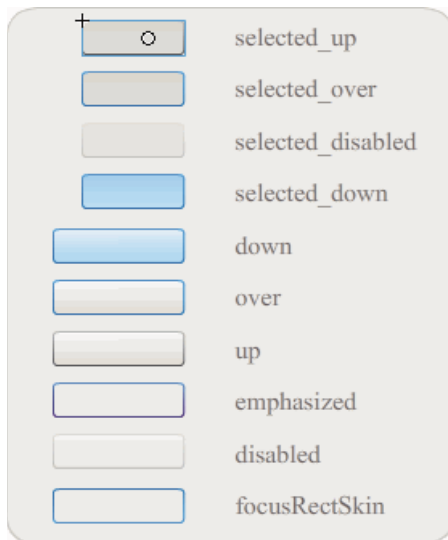
Komponenten-Architektur

Adobe® ActionScript® 3.0-Komponenten werden von Adobe® Flash Player Version 9.0.28.0 und höher unterstützt. Diese Komponenten sind nicht kompatibel mit Komponenten, die vor CS4 erstellt worden sind. Informationen zur Verwendung von Adobe® ActionScript® 2.0-Komponenten finden Sie in den Handbüchern *Adobe® ActionScript® 2.0-Komponenten verwenden* und *Adobe® ActionScript® 2.0 Referenzhandbuch*.

Die Adobe ActionScript 3.0-Benutzeroberflächenkomponenten (oder UI-Komponenten f. engl. „User Interface“) wurden als FLA-basierte Komponenten implementiert, doch unterstützt Flash CS5 sowohl SWC- als auch FLA-basierte Komponenten. Die Komponenten „FLVPlayback“ und „FLVPlaybackCaptioning“ etwa sind SWC-basierte Komponenten. Sie können beide Komponententypen im Ordner „Components“ ablegen, damit sie im Bedienfeld „Komponenten“ angezeigt werden. Diese beiden Komponententypen werden auf unterschiedliche Weise erstellt, daher sind sie hier gesondert beschrieben.

FLA-basierte ActionScript 3.0-Komponenten

Die UI-Komponenten von ActionScript 3.0 sind FLA-basierte Dateien (.fla-Dateien) mit integrierten Skins, die Sie bearbeiten können, indem Sie auf der Bühne auf die Komponente doppelklicken. Die Skins und die anderen Elemente der Komponente befinden sich in Bild 2 auf der Zeitleiste. Wenn Sie auf die Komponente doppelklicken, wird von Flash automatisch das Bild 2 angesteuert und es wird eine Palette mit den Skins der Komponente geöffnet. In der folgenden Abbildung sehen Sie die Palette der Skins, die für die Button-Komponente angezeigt werden.



Skins für die Button-Komponente

Weitere Informationen zu Komponentenskins und zum Anpassen von Komponenten finden Sie im Kapitel „Anpassen der UI-Komponenten“ auf Seite 104 und im Abschnitt „Anpassen der FLVPlayback-Komponente“ auf Seite 163.

Um die Kompilierung für Ihre Anwendungen zu beschleunigen und um Konflikte mit Ihren ActionScript 3.0-Einstellungen zu vermeiden, enthalten die Flash CS5 FLA-basierten UI-Komponenten auch eine SWC, die den bereits kompilierten ActionScript-Code der Komponente enthält. Die ComponentShim-SWC befindet sich auf der Bühne in Bild 2 jeder UI-Komponente, damit die vorkompilierten Definitionen verfügbar sind. Damit eine Komponente für ActionScript zur Verfügung steht, muss sie sich entweder auf der Bühne oder in der Bibliothek befinden, wobei die Option „In erstes Bild exportieren“ in den Verknüpfungseigenschaften ausgewählt sein muss. Wenn Sie eine Komponente mithilfe von ActionScript-Code erstellen, müssen Sie die Klasse auch mit einer `import`-Anweisung importieren, damit Sie darauf zugreifen können. Informationen zur `import`-Anweisung finden Sie im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

SWC-basierte Komponenten

SWC-basierte Komponenten verfügen ebenfalls über eine FLA-Datei und eine ActionScript-Klassendatei, sie wurden jedoch als SWC-Datei kompiliert und exportiert. Eine SWC-Datei ist ein Paket mit vorkompilierten Flash-Symbolen und ActionScript-Code, das Ihnen eine Neukompilierung von Symbolen und Code, die sich nicht ändern, erspart.

FLVPlayback und FLVPlaybackCaptioning sind SWC-basierte Komponenten. Anstelle von integrierten Skins besitzen sie externe Skins. Die FLVPlayback-Komponente verfügt über eine Standardskin, die Sie ändern können, indem Sie aus einer Sammlung vordefinierter Skins eine andere auswählen, die Steuerelemente im Bedienfeld „Komponenten“ anpassen (BackButton, BufferingBar usw.) oder eine benutzerdefinierte Skin erstellen. Weitere Informationen finden Sie im Abschnitt [„Anpassen der FLVPlayback-Komponente“](#) auf Seite 163.

In Flash können Sie wie folgt einen Movieclip in einen kompilierten Clip konvertieren:

Kompilieren eines Movieclips

- Klicken Sie mit der rechten Maustaste (Windows) bzw. bei gedrückter Ctrl-Taste (Macintosh) auf den gewünschten Movieclip im Bedienfeld „Bibliothek“ und wählen Sie die Option „In kompilierten Clip konvertieren“.

Der kompilierte Clip verhält sich genauso wie der Movieclip, aus dem er kompiliert wurde. Kompilierte Clips lassen sich jedoch wesentlich schneller anzeigen und veröffentlichen als gewöhnliche Movieclips. Kompilierte Clips können zwar nicht bearbeitet werden, aber ihre Eigenschaften können im Eigenschaften- und Komponenten-Inspektor angezeigt werden.

SWC-Komponenten enthalten einen kompilierten Clip, die vorkompilierten ActionScript-Definitionen sowie weitere Dateien, die die Komponente beschreiben. Wenn Sie eigene Komponenten erstellen, können Sie sie als SWC-Datei exportieren, um sie zu verteilen.

Exportieren einer SWC-Datei

- Klicken Sie mit der rechten Maustaste (Windows) bzw. bei gedrückter Ctrl-Taste (Macintosh) auf den gewünschten Movieclip im Bedienfeld „Bibliothek“ und wählen Sie die Option „SWC-Datei exportieren“ aus.

Hinweis: Das Format einer SWC-Datei ab Version Flash CS4 ist kompatibel mit dem Flex-SWC-Format, sodass SWC-Dateien zwischen den beiden Produkten ausgetauscht werden können, aber nicht notwendigerweise ohne Modifikationen.

Informationen über die Erstellung SWC-basierter Komponenten finden Sie unter www.adobe.com/go/learn_fl_creating_components_de.

Arbeiten mit der API für ActionScript 3.0-Komponenten

Jede ActionScript 3.0-Komponente wurde aus einer ActionScript 3.0-Klasse erstellt, die sich in einem Paketordner befindet und einen Namen im Format „fl.Paketname.Klassenname“ besitzt. Die Button-Komponente etwa ist eine Instanz der Button-Klasse und hat den Paketnamen `fl.controls.Button`. Sie müssen auf den Paketnamen verweisen, wenn Sie eine Komponentenkategorie in Ihre Anwendung importieren. Die Button-Klasse würden Sie mit der folgenden Anweisung importieren:

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

Weitere Information über den Speicherort von Komponentendateien finden Sie im Abschnitt „[Arbeiten mit Komponentendateien](#)“ auf Seite 20.

Eine Komponentenkategorie definiert die Methoden, Eigenschaften, Ereignisse und Stile, die Ihnen eine Interaktion mit der Komponenteinstanz in einer Anwendung ermöglichen. Die ActionScript 3.0-UI-Komponenten sind Unterklassen der Sprite- und UIComponent-Klassen und erben von diesen auch ihre Methoden, Eigenschaften und Ereignisse. Die Sprite-Kategorie ist der Grundbaustein einer Anzeigeliste und ähnelt einem Movieclip, besitzt aber keine Zeitleiste. Die UIComponent-Kategorie ist die Basiskategorie für alle visuellen Komponenten, sowohl interaktive als auch nicht interaktive. Der Vererbungspfad der einzelnen Komponenten sowie deren Eigenschaften, Methoden, Ereignisse und Stile werden im *Adobe ActionScript 3.0 Referenzhandbuch für Flash Professional* beschrieben.

Alle ActionScript 3.0-Komponenten verwenden das Ereignisverarbeitungsmodell von ActionScript 3.0. Weitere Informationen hierzu finden Sie im Abschnitt „[Verarbeiten von Ereignissen](#)“ auf Seite 25 und im Handbuch *Programmieren mit ActionScript 3.0*.

Arbeiten mit Komponentendateien

In diesem Abschnitt wird beschrieben, wo Komponentendateien gespeichert werden, wo die ActionScript-Quelldateien zu finden sind und wie Komponenten zum Bedienfeld „Komponenten“ hinzugefügt bzw. daraus entfernt werden.

Pfad der Komponentendateien

Flash-Komponenten werden im Anwendungsordner „Configuration“ gespeichert.

Hinweis: Informationen zu diesen Ordnern finden Sie im Handbuch „Verwenden von Flash“ im Abschnitt „Mit Flash installierte Konfigurationsordner“.

Komponenten werden in den unten angeführten Ordnern installiert.

- Windows 2000 und Windows XP: `C:\Programme\Adobe\Adobe Flash CS5\de\Configuration\Components`
- Mac OS X: `Macintosh HD:Applications:Adobe Flash CS5:Configuration:Components`

Innerhalb des Ordners „Components“ befinden sich die UI-Komponenten in der Datei „User Interface fla“, und die Komponenten FLVPlayback (FLVPlaybackAS3.swc) und FLVPlaybackCaptioning befinden sich im Ordner „Video“.

Sie können Komponenten auch in den unten angeführten benutzerabhängigen Ordnern speichern.

- Windows 2000 und Windows XP: `C:\C:\Dokumente und Einstellungen\Benutzername\Lokale Einstellungen\Anwendungsdaten\Adobe\Flash CS 5\de\Configuration\Components`
- Windows Vista: `C:\Benutzer\Benutzername\Lokale Einstellungen\Anwendungsdaten\Adobe\Adobe Flash CS5\de\Configuration\Components`

Hinweis: In Windows ist der Ordner „Application Data“ standardmäßig ausgeblendet. Um verborgene Ordner anzuzeigen, wählen Sie „Start“ > „Arbeitsplatz“, um den Windows-Explorer zu öffnen, und wählen darin „Extras“ > „Ordneroptionen“. Aktivieren Sie auf der Registerkarte „Ansicht“ das Kontrollkästchen „Alle Dateien und Ordner anzeigen“.

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

Pfad der Quelldateien von Komponenten

Die ActionScript-Klassendateien (oder *Quelldateien*) von Komponenten (mit der Endung .as) sind unter Windows 2000 und Windows XP in den unten angeführten Anwendungsordnern gespeichert.

Benutzeroberflächenkomponenten: C:\Programme\Adobe\Adobe Flash CS5\de\Configuration\Component Source\ActionScript 3.0\User Interface\fl

FLVPlayback: C:\Programme\Adobe\Adobe Flash CS5\de\Configuration\Component Source\ActionScript 3.0\FLVPlayback\fl\video

FLVPlaybackCaptioning: C:\Programme\Adobe\Adobe Flash CS5\de\Configuration\Component Source\ActionScript 3.0\FLVPlaybackCaptioning\fl\video

Unter Mac OS X sind die Quelldateien der Komponenten in den unten angeführten Ordnern zu finden.

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

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

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

Quelldateien der Komponenten und Klassenpfad

Da der Code der ActionScript 3.0-Komponenten in die Komponenten einkompiliert ist, sollten Sie den Pfad der ActionScript-Klassendateien normalerweise nicht in der Classpath-Variable angeben. Denn wenn Sie deren Speicherort in den Klassenpfad einbinden, dauert es länger, Ihre Anwendungen zu kompilieren. Wenn Flash aber Komponentenklassendateien in der Classpath-Einstellung findet, hat die Klassendatei immer Vorrang vor dem einkompilierten Komponentencode.

Eine Situation, in der es sich empfiehlt, den Pfad der Komponentenquelldateien in die Classpath-Einstellung einzubinden, ist bei der Fehlersuche in einer Anwendung mit Komponenten. Weitere Informationen finden Sie im Abschnitt „[Fehlersuche in Komponentenanwendungen](#)“ auf Seite 22.

Ändern der Komponentendateien

Wenn Sie SWC-basierte Komponenten aktualisieren, hinzufügen oder entfernen oder wenn Sie neue FLA-basierte Komponenten zu Flash hinzufügen, müssen Sie sie neu in das Bedienfeld „Komponenten“ laden, damit sie verfügbar sind. Sie können Komponenten neu laden, indem Sie Flash neu starten oder indem Sie im Menü des Bedienfelds „Komponenten“ den Menüpunkt „Neu laden“ wählen. Flash lädt dann alle Komponenten, die Sie in den Ordner „Components“ eingefügt haben.


Neuladen von Komponenten in das Bedienfeld „Komponenten“ (Flash wird ausgeführt):

- Wählen Sie im Bedienfeld „Komponenten“ den Befehl „Neu laden“ aus dem Menü.

Entfernen einer Komponente aus dem Bedienfeld „Komponenten“:

- Entfernen Sie die FLA-, SWC- oder MXP-Datei aus dem Ordner „Components“ und starten Sie Flash neu, oder wählen Sie im Menü des Bedienfelds „Komponenten“ den Menüpunkt „Neu laden“. Eine MXP-Datei ist eine Komponentendatei, die von Adobe Exchange heruntergeladen wurde.

Sie können SWC-basierte Komponenten entfernen und ersetzen, während Flash ausgeführt wird, und diese Änderungen sind nach dem Neuladen sichtbar. Wenn Sie jedoch FLA-basierte Komponenten ändern oder löschen, sind die Änderungen erst dann wirksam, wenn Sie Flash beenden und neu starten. Sie können jedoch FLA-basierte Komponenten hinzufügen und mit dem Befehl „Neu laden“ laden.

 *Adobe empfiehlt Ihnen, von jeder Flash-Komponentendatei (.fla oder .as), die Sie ändern wollen, erst einmal eine Sicherungskopie anzulegen. Auf diese Weise können Sie sie bei Bedarf wieder herstellen.*

Fehlersuche in Komponentenanwendungen

Die ActionScript 3.0-Komponenten enthalten ihren gesamten Quellcode, um die Kompilierungszeit von Anwendungen möglichst gering zu halten. Der Flash-Debugger kann jedoch keinen Code innerhalb von kompilierten Clips untersuchen. Wenn Sie daher die Fehlersuche in Ihren Anwendungen bis zum Komponentenquellcode hinunter durchführen wollen, müssen Sie die Komponentenquelldateien in die Classpath-Einstellung einbinden.

Der Komponentenpaketordner ist relativ zum Speicherort der Quelldateien für den betreffenden Komponententyp angeordnet. Um auf alle ActionScript 3.0-Quelldateien für alle UI-Komponenten zu verweisen, fügen Sie zum Klassenpfad für die UI-Pakete folgenden Pfad hinzu:

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

Hinweis: *Dadurch haben die Quelldateien Vorrang vor dem einkompilierten Code aller UI-Komponenten, allerdings wird auch die Kompilierungszeit für Ihre Anwendung erhöht. Wenn Sie aus irgendeinem Grund die Quelldatei einer Komponente geändert haben, hat die Komponente als Folge möglicherweise ein anderes Verhalten.*

Um den Klassenpfad festzulegen, wählen Sie „Bearbeiten“ > „Voreinstellungen“ und wählen in der Kategorienliste „ActionScript“ aus. Klicken Sie dann auf die Schaltfläche „ActionScript 3.0-Einstellungen“. Um einen neuen Eintrag hinzuzufügen, klicken Sie auf das Pluszeichen über dem Fenster, in dem die aktuellen Einstellungen angezeigt werden.

Die Variable \$(AppConfig) bezieht sich auf den Flash CS5-Ordner „Configuration“ im Installationsverzeichnis von Flash CS5. Typischerweise ist der Pfad wie unten angegeben.

- Windows 2000 und Windows XP: C:\Programme\Adobe\Adobe Flash CS5\de\Configuration\
- Mac OS X: Macintosh HD:Applications:Adobe Flash CS5:Configuration

Hinweis: *Wenn Sie eine Komponentenquelldatei ändern müssen, ist es stark anzuraten, die Originalquelldatei in einen anderen Ordner zu kopieren, und den neuen Pfad zum Klassenpfad hinzuzufügen.*

Weitere Informationen über den Pfad der Quelldateien von Komponenten finden Sie im Abschnitt „[Pfad der Quelldateien von Komponenten](#)“ auf Seite 21.

Festlegen von Parametern und Eigenschaften

Jede Komponente verfügt über Parameter, die Sie ändern können, um das Aussehen und Verhalten der Komponente zu ändern. Ein Parameter ist eine Eigenschaft der Komponentenkategorie und wird im Eigenschaften- oder Komponenteninspektor angezeigt. Die am häufigsten verwendeten Eigenschaften werden als Authoring-Parameter angezeigt, während Sie andere mit ActionScript festlegen müssen. Alle Parameter, die beim Authoring eingestellt werden können, lassen sich auch mit ActionScript festlegen. Wenn Sie einen Parameter mit ActionScript festlegen, wird der beim Authoring festgelegte Wert außer Kraft gesetzt.

Die meisten ActionScript 3.0-UI-Komponenten übernehmen Eigenschaften und Methoden von der UIComponent-Klasse sowie von einer Basisklasse. Die Klassen „Button“ und „CheckBox“ beispielsweise erben Eigenschaften sowohl von der UIComponent-Klasse als auch von der BaseButton-Klasse. Die geerbten Eigenschaften einer Klasse sowie ihre eigenen Klasseneigenschaften sind für Sie zugänglich. Die ProgressBar-Komponente erbt etwa die Eigenschaft `ProgressBar.enabled` von der UIComponent-Klasse, hat aber seine eigene `ProgressBar.percentComplete`-Eigenschaft. Um mit einer Instanz der ProgressBar-Komponente zu interagieren, können Sie auf beide Eigenschaften zugreifen. Weitere Informationen zu den Eigenschaften einer Komponente finden Sie im entsprechenden Klasseneintrag im [ActionScript 3.0 Referenzhandbuch](#).

Die Parameter einer Komponenteninstanz werden entweder im Eigenschafteninspektor oder auf der Registerkarte „Parameter“ im Komponenten-Inspektor eingestellt.

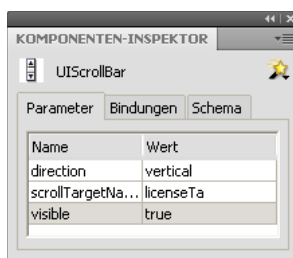
Eingeben des Instanznamens einer Komponente im Eigenschafteninspektor:

- 1 Wählen Sie „Fenster“ > „Eigenschaften“ > „Eigenschaften“.
- 2 Wählen Sie auf der Bühne die Instanz einer Komponente aus.
- 3 Geben Sie einen Namen für die Komponenteninstanz in das Feld mit der Angabe „<Instanzname>“ (über der Dropdownliste „Movieclip“) ein. Alternativ dazu können Sie auch auf die Registerkarte „Parameter“ klicken und den Namen in das Feld unter *Komponente* eingeben. Geben Sie Werte für jeden Parameter ein, den Sie festlegen wollen.

Es empfiehlt sich, dem Instanznamen ein Suffix hinzuzufügen, das die Art der Komponente kennzeichnet. Dies erleichtert das Lesen des ActionScript-Codes. Der Instanzname **licenseSb** bezeichnet z. B. eine Bildlaufleiste (ScrollBar), mit der der Benutzer durch eine Lizenzvereinbarung im Textbereich **licenseTa** blättern kann.

Eingeben von Parametern einer Komponenteninstanz im Komponenten-Inspektor:

- 1 Wählen Sie „Fenster“ > „Komponenten-Inspektor“.
- 2 Wählen Sie auf der Bühne die Instanz einer Komponente aus.
- 3 Klicken Sie auf die Registerkarte „Parameter“ und geben Sie Werte für beliebige der aufgeführten Parameter ein.



Komponenten-Parameter im Komponenten-Inspektor

Festlegen von Komponenteneigenschaften in ActionScript

In ActionScript verwenden Sie einen Punktoperator (.) (Punktsyntax), um auf Eigenschaften und Methoden zuzugreifen, die zu einem Objekt oder einer Instanz auf der Bühne gehören. Ein in der Punktsyntax geschriebener Ausdruck beginnt mit dem Namen der Instanz, gefolgt von einem Punkt, und endet mit dem Element, das Sie angeben möchten. Der folgende ActionScript-Code legt beispielsweise die Eigenschaft `width` der `CheckBox`-Instanz `aCh` so fest, dass das Kontrollkästchen 50 Pixel breit ist:

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

Die folgende `if`-Anweisung überprüft, ob der Benutzer das Kontrollkästchen ausgewählt hat:

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

Arbeiten mit der Bibliothek

Wenn Sie eine Komponente zum ersten Mal einem Dokument hinzufügen, importiert Flash sie als Movieclip in das Bedienfeld „Bibliothek“. Sie können eine Komponente aus dem Bedienfeld „Komponenten“ auch direkt in das Bedienfeld „Bibliothek“ ziehen und der Bühne dann Instanzen der Komponente hinzufügen. Sie müssen eine Komponente auf jeden Fall zur Bibliothek hinzufügen, bevor Sie auf die Klassenelemente zugreifen können.

Wenn Sie der Bibliothek eine Komponente hinzufügen und mithilfe von ActionScript eine Instanz davon erstellen, müssen Sie zunächst mit der `import`-Anweisung die Klasse der Komponente importieren. In der `import`-Anweisung müssen Sie sowohl den Paketnamen als auch den Klassennamen der Komponente angeben. Mit der folgenden Anweisung wird zum Beispiel die `Button`-Klasse importiert:

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

Wenn Sie eine Komponente in der Bibliothek platzieren, importiert Flash auch einen Ordner mit ihren Elementen, der die Skins für die verschiedenen Zustände enthält. Die *Skins* einer Komponente beinhalten die Sammlung an Symbolen für die grafische Darstellung der Komponente in der Anwendung. Eine einzelne Skin ist eine grafische Repräsentation (oder Movieclip) die einen bestimmten Zustand der Komponente bezeichnet.

Der Inhalt des Bibliothekordners „Component Assets“ ermöglicht Ihnen, bei Bedarf die Skin der Komponente zu ändern. Weitere Informationen finden Sie im Kapitel „[Anpassen der UI-Komponenten](#)“ auf Seite 104.

Nachdem eine Komponente zur Bibliothek hinzugefügt wurde, können Sie dem Dokument weitere Instanzen davon hinzufügen, indem Sie das Symbol der Komponente aus dem Bedienfeld „Komponenten“ oder aus dem Bedienfeld „Bibliothek“ auf die Bühne ziehen.

Größe der Komponenten ändern

Die Größe der Komponenteninstanzen kann mit dem Werkzeug „Frei transformieren“ oder mit der Methode `setSize()` angepasst werden. Die Methode `setSize()` kann von jeder Komponenteninstanz aufgerufen werden, um die Größe zu ändern (siehe `UIComponent.setSize()`). Mit dem folgenden Code wird die Größe der Instanz einer `List`-Komponente auf eine Breite von 200 Pixel und eine Höhe von 300 Pixel eingestellt:

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

Die Größe einer Komponente wird nicht automatisch an die Komponentenbeschriftung angepasst. Wenn die Komponenteninstanz, die in ein Dokument eingefügt wurde, für die Darstellung ihrer Beschriftung nicht ausreicht, wird der Text in der Anzeige abgeschnitten. Sie müssen die Größe der Komponente so ändern, dass sie der Beschriftung entspricht.

Weitere Informationen zum Anpassen der Größe von Komponenten finden Sie unter den jeweiligen Einträgen im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

Live-Vorschau

Mithilfe der standardmäßig aktivierten Live-Vorschau können Sie Komponenten so auf der Bühne anzeigen, wie sie nach der Veröffentlichung in Flash aussehen werden. Die Komponenten werden mit ihrer ungefähren Größe angezeigt.

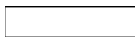
Ein- und Ausschalten der Live-Vorschau

- Wählen Sie „Steuerung“ > „Live-Vorschau aktivieren“. Ein Häkchen neben der Option zeigt an, dass sie aktiviert ist.

Die Live-Vorschau gibt je nach Komponente verschiedene Parameter wieder. Informationen dazu, welche Komponentenparameter in der Live-Vorschau dargestellt werden, finden Sie in den Einträgen zu den einzelnen Komponenten im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).



Button-Komponente bei aktivierter Live-Vorschau



Button-Komponente bei deaktivierter Live-Vorschau

Die Funktionsweise einer Komponente lässt sich in der Live-Vorschau nicht überprüfen. Um die Funktion zu testen, müssen Sie den Befehl „Steuerung > „Film testen“ verwenden.

Verarbeiten von Ereignissen

Jede Komponente sendet Ereignisse aus, wenn ein Benutzer mit ihr in Interaktion tritt. Wenn ein Benutzer zum Beispiel auf eine Schaltfläche klickt, löst die Button-Komponente das Ereignis `MouseEvent.CLICK` aus, und wenn ein Benutzer einen Eintrag in einer Liste auswählt, löst die List-Komponente das Ereignis `Event.CHANGE` aus. Ein Ereignis kann auch eintreten, wenn mit einer Komponente etwas signifikantes passiert, z. B. wenn ein Ladevorgang in einer `UI Loader`-Instanz abgeschlossen ist. In diesem Fall würde ein `Event.COMPLETE`-Ereignis erzeugt werden. Zum Verarbeiten eines Ereignisses schreiben Sie ActionScript-Code, der ausgeführt wird, wenn das Ereignis eintritt.

Eine Komponente umfasst zunächst einmal die Ereignisse aller Klassen, von denen die Komponente erbt. Das bedeutet, dass alle ActionScript 3.0-UI-Komponenten Ereignisse von der `UIComponent`-Klasse übernehmen, da diese die Basisklasse für die ActionScript 3.0-UI-Komponenten ist. Eine Liste der Ereignisse, die von einer Komponente gesendet werden, finden Sie im Abschnitt „Ereignisse“ des Klasseneintrags der jeweiligen Komponente im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

Eine vollständige Erläuterung zur Verarbeitung von Ereignissen in ActionScript 3.0 finden Sie im Handbuch *Programmieren mit ActionScript 3.0*.

Hintergrundinformationen zu Ereignis-Listnern

Die folgenden zentralen Punkte gelten generell für die Verarbeitung von Ereignissen für ActionScript 3.0-Komponenten:

- Alle Ereignisse werden von einer Instanz einer Komponentenklasse ausgesendet. Die Komponenteninstanz ist dabei der Sender oder *Broadcaster*.
- Sie registrieren einen Ereignis-*Listener*, indem Sie für die Komponenteninstanz die `addEventListener()`-Methode aufrufen. Die folgende Codezeile etwa fügt einen Listener für das Ereignis `MouseEvent.CLICK` zur Button-Instanz `aButton` hinzu:

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

Der zweite Parameter der `addEventListener()`-Methode registriert den Namen der Funktion, `clickHandler`, die aufgerufen werden soll, wenn das Ereignis eintritt. Diese Funktion wird auch als *Rückruffunktion* bezeichnet.

- Sie können mehrere Listener mit einer Komponenteninstanz registrieren.

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

- Sie können einen Listener mit mehreren Komponenteninstanzen registrieren.

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

- Die Ereignisprozedur wird an ein Ereignisobjekt übergeben, das Informationen zum Ereignistyp und zu der Instanz, die das Ereignis überträgt, enthält. Weitere Informationen finden Sie im Abschnitt „[Hintergrundinformationen zum Ereignisobjekt](#)“ auf Seite 26.
- Der Listener bleibt aktiv, bis die Anwendung beendet wird oder bis Sie ihn mithilfe der Methode `removeEventListener()` explizit entfernen. Die folgende Codezeile etwa entfernt den Listener für das `MouseEvent.CLICK`-Ereignis zu `aButton`:

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

Hintergrundinformationen zum Ereignisobjekt

Das Ereignisobjekt erbt von der Event-Klasse und verfügt über Eigenschaften, die Informationen zum eingetretenen Ereignis enthalten, darunter die Eigenschaften `target` und `type`, die wichtige Informationen zum Ereignis zur Verfügung stellen:

Eigenschaft	Beschreibung
<code>type</code>	Ein String, der den Ereignistyp angibt.
<code>target</code>	Ein Verweis auf die Komponenteninstanz, die das Ereignis per Broadcast aussendet.

Wenn ein Ereignis zusätzliche Eigenschaften aufweist, sind diese in der Klassenbeschreibung des Ereignisses im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#) aufgeführt.

Das Ereignisobjekt wird automatisch erzeugt und an die Ereignisprozedurfunktion übergeben, wenn das Ereignis eintritt.

Sie können das Ereignisobjekt in der Funktion verwenden, um auf den Namen des übertragenen Ereignisses oder auf den Instanznamen der Komponente, die das Ereignis übertragen hat, zuzugreifen. Vom Instanznamen aus können Sie auf die anderen Komponenteneigenschaften zugreifen. Im folgenden Code wird beispielsweise die Eigenschaft `target` des Ereignisobjekts `evtObj` verwendet, um auf die Eigenschaft `label` der Instanz `aButton` zuzugreifen und den Wert im Bedienfeld „Ausgabe“ anzuzeigen.

```
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");
}
```

Arbeiten mit der Anzeigeliste

Alle ActionScript 3.0-Komponenten erben von der DisplayObject-Klasse und haben daher Zugang zu den Methoden und Eigenschaften dieser Klasse, die zur Interaktion mit der Anzeigeliste dienen. Die *Anzeigeliste* repräsentiert die Hierarchie der angezeigten Objekte und visuellen Elemente in einer Anwendung. Diese Hierarchie umfasst die folgenden Elemente:

- Die Bühne, die den Container auf oberster Ebene darstellt
- Anzeigobjekte, also u. a. Formen, Movieclips und Textfelder
- Anzeigobjektcontainer, also besondere Anzeigobjekte, die untergeordnete Anzeigobjekte enthalten können

Die Reihenfolge der Objekte in der Anzeigeliste bestimmt deren Tiefe innerhalb des übergeordneten Containers. Die Tiefe eines Objekts bezieht sich auf seine Position von oben nach unten bzw. von vorne nach hinten auf der Bühne oder seinem Anzeigcontainer. Die Tiefenreihenfolge ist offensichtlich, wenn sich die Objekte überschneiden, sie existiert aber auch, wenn dies nicht der Fall ist. Jedes Objekt in der Anzeigeliste hat eine entsprechende Tiefe auf der Bühne. Wenn Sie die Tiefe eines Objekts ändern möchten, indem Sie es vor oder hinter andere Objekte verschieben, müssen Sie die Position des Objekts in der Anzeigeliste ändern. Die Standardreihenfolge von Objekten in der Anzeigeliste ist die Reihenfolge, in der sie auf der Bühne platziert wurden. Position 0 in der Anzeigeliste ist das Objekt ganz unten in der Tiefenreihenfolge.

Einfügen von Komponenten in die Anzeigeliste

Sie können in ein DisplayObjectContainer-Objekt ein Objekt einfügen, indem Sie die Methode `addChild()` oder `addChildAt()` des Containers aufrufen. Im Fall der Bühne können Sie ein Objekt auch beim Authoring in die Anzeigeliste einfügen, indem Sie es erstellen, oder, wenn es sich um Komponenten handelt, indem Sie es aus dem Bedienfeld „Komponenten“ auf die Bühne ziehen. Um ein Objekt mit ActionScript in einen Container einzufügen, erstellen Sie zunächst eine Instanz, indem Sie mit dem Operator `new` seinen Konstruktor aufrufen und dann die Methode `addChild()` oder `addChildAt()` aufrufen, um es auf der Bühne und somit in der Anzeigeliste zu platzieren. Die Methode `addChild()` platziert das Objekt auf der nächsten Position in der Anzeigeliste, während Sie mit `addChildAt()` die Position angeben können, auf der das Objekt platziert werden soll. Wenn Sie eine Position angeben, die bereits besetzt ist, werden das Objekt auf dieser Position und alle darüber liegenden Objekte um eine Position nach oben gerückt. Die Eigenschaft `numChildren` eines DisplayObjectContainer-Objekts bezeichnet die Anzahl der enthaltenen Objekte. Sie können ein Objekt aus der Anzeigeliste abrufen, indem Sie die Methode `getChildAt()` aufrufen und die Position des Objekts angeben oder, wenn Sie seinen Namen kennen, indem Sie `getChildByName()` aufrufen.

Hinweis: Wenn Sie eine Komponente mithilfe von ActionScript-Code einfügen, müssen Sie seiner `name`-Eigenschaft einen Wert zuweisen, um sie später per Name aus der Anzeigeliste abrufen zu können.

Mit dem folgenden Beispiel werden die Namen und Positionen von drei Komponenten in der Anzeigeliste ausgegeben. Ziehen Sie zuerst eine NumericStepper-, eine Button- und eine ComboBox-Instanz so auf die Bühne, dass sie sich überschneiden, und weisen Sie ihnen die Instanznamen **aNs**, **aButton** und **aCb** zu. Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Zeitleiste den folgenden Code hinzu.

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

Im Bedienfeld „Ausgabe“ werden die folgenden Zeilen angezeigt:

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

Verschieben von Komponenten innerhalb der Anzeigeliste

Sie können die Position eines Objekts innerhalb der Anzeigeliste, und somit seine Anzeigtiefe, ändern, indem Sie die Methode `addChildAt()` aufrufen und als Methodenparameter den Namen des Objekts und die Position, auf der es platziert werden soll, angeben. Fügen Sie dem vorstehenden Beispiel den folgenden Code hinzu, um das NumericStepper-Objekt nach ganz oben zu verschieben, und wiederholen Sie die Schleife, um die neuen Positionen der Komponenten in der Anzeigeliste anzuzeigen:

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

Im Bedienfeld „Ausgabe“ sollten Sie Folgendes sehen:

```
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
```

Die NumericStepper-Komponente sollte auch vor den anderen Komponenten auf dem Bildschirm angezeigt werden.

Beachten Sie, dass die Eigenschaft `numChildren` die Nummer (oder Tiefenreihenfolge) eines Objekts innerhalb der Anzeigeliste bezeichnet (von 1 bis n), während die erste Position bei 0 liegt. Wenn sich in der Liste also drei Objekte befinden, lautet die Indexposition des dritten Objekts 2. Das bedeutet, dass Sie mit `numChildren - 1` auf die letzte Position in der Anzeigeliste, d. h., auf das oberste Objekt, verweisen können.

Entfernen von Komponenten aus der Anzeigeliste

Sie können eine Komponente mit den Methoden `removeChild()` und `removeChildAt()` aus einem Anzeigebereichscontainer und seiner Anzeigeliste entfernen. Im folgenden Beispiel werden drei Button-Komponenten übereinander auf der Bühne abgelegt, und für jede Komponente wird ein Ereignis-Listener hinzugefügt. Wenn der Benutzer auf die einzelnen Schaltflächen klickt, werden sie von der Ereignisprozedur aus der Anzeigeliste und von der Bühne entfernt.

- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie eine Button-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.

- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und fügen Sie den folgenden Code ein:

```
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);
}
```

Eine vollständige Erläuterung der Anzeigeliste finden Sie im Abschnitt „Programmierung von Anzeigeobjekten“ im Handbuch *Programmieren mit ActionScript 3.0*.

Arbeiten mit dem FocusManager

Wenn ein Benutzer zur Navigation in einer Flash-Anwendung die Tabulatortaste drückt, bestimmt der FocusManager, welche Komponente den Eingabefokus erhält. Sie brauchen eine FocusManager-Instanz nur dann in eine Anwendung einfügen oder Code schreiben, um den FocusManager zu aktivieren, wenn Sie eine Komponente erstellen wollen.

Wenn ein RadioButton-Objekt den Fokus erhält, wird dieses Objekt sowie alle Objekte mit demselben `groupName`-Wert vom FocusManager überprüft, und der Fokus wird demjenigen Objekt gegeben, für das die Eigenschaft `selected` auf `true` gesetzt ist.

Jede modale Window-Komponente enthält eine FocusManager-Instanz, sodass die Steuerelemente in diesem Fenster eine eigene und in sich geschlossene Tabulatorreihenfolge erhalten. Hierdurch wird verhindert, dass ein Benutzer mit der Tabulatortaste versehentlich zu Komponenten in anderen Fenstern gelangt.

Der FocusManager verwendet die Tiefenebene (oder *z*-Reihenfolge) der Elemente im Container als Standardnavigationsschema oder *Tabulatorschleife*. Benutzer navigieren normalerweise mit der Tabulatortaste durch die Tabulatorschleife, wobei sich der Fokus von der ersten Komponente, die den Fokus hat, bis zur letzten bewegt und dann wieder zu ersten. Die Tiefenebenen werden in erster Linie von der Reihenfolge festgelegt, in der Komponenten auf die Bühne gezogen werden. Sie können die endgültige *z*-Reihenfolge jedoch auch mithilfe der Befehle „Modifizieren“ > „Anordnen“ > „In den Vordergrund“ bzw. „In den Hintergrund“ festlegen. Weitere Informationen zur Tiefenreihenfolge finden Sie im Abschnitt [„Arbeiten mit der Anzeigeliste“](#) auf Seite 27.

Sie können die Methode `setFocus()` aufrufen, um einer Komponenteninstanz in einer Anwendung den Fokus zu geben. Im folgenden Beispiel wird eine FocusManager-Instanz für den aktuellen Container (`this`) erstellt. Den Fokus erhält die Button-Instanz `aButton`.

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

Sie können feststellen, welche Komponente den Fokus hat, indem Sie die Methode `getFocus()` aufrufen. Um festzustellen, welche Komponente in der Tabulatorschleife als nächste den Fokus erhält, rufen Sie die Methode `getNextFocusManagerComponent()` auf. Im folgenden Beispiel befinden sich eine `CheckBox`-, eine `RadioButton`- und eine `Button`-Komponente auf der Bühne. Jede dieser Komponenten hat einen Listener für die Ereignisse `MouseEvent.CLICK` und `FocusEvent.MOUSE_FOCUS_CHANGE`. Wenn das `MouseEvent.CLICK`-Ereignis eintritt, weil der Benutzer auf die Komponente geklickt hat, ruft die Funktion `showFocus()` die Methode `getNextFocusManagerComponent()` auf, um festzustellen, welche Komponente in der Tabulatorschleife als nächste den Fokus erhalten würde. Dann wird die Methode `setFocus()` aufgerufen, um dieser Komponente den Fokus zu geben. Wenn das Ereignis `FocusEvent.MOUSE_FOCUS_CHANGE` eintritt, zeigt die `fc()`-Funktion den Namen der Komponente an, bei der dieses Ereignis aufgetreten ist. Dieses Ereignis wird ausgelöst, wenn der Benutzer auf eine Komponente klickt, die nicht die nächste in der Tabulatorschleife ist.

```
// 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);
}
```

Sie können eine Schaltfläche erstellen, die beim Drücken der Eingabetaste (Windows) bzw. der Return-Taste (Macintosh) den Fokus erhält, indem Sie die Eigenschaft `FocusManager.defaultButton` auf die Instanz der als Standardschaltfläche gewünschten Schaltfläche setzen. Dies wird im folgenden Code gezeigt:

```
import fl.managers.FocusManager;

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

Die `FocusManager`-Klasse überschreibt das Standardfokusrechteck von Flash Player und zeichnet ein angepasstes Fokusrechteck mit abgerundeten Ecken.

Weitere Informationen zum Erstellen eines Fokusschemas in einer Flash-Anwendung finden Sie im Abschnitt zur `FocusManager`-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#). Um einen benutzerdefinierten Fokusmanager zu erstellen, müssen Sie eine Klasse erstellen, die die `IFocusManager`-Schnittstelle implementiert. Weitere Informationen finden Sie unter `IFocusManager` im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

Arbeiten mit List-basierten Komponenten

Die List-, DataGrid- und TileList-Komponenten erben alle von der SelectList-Basisklasse. Aus diesem Grund werden diese Komponenten als listengestützte Komponenten betrachtet. Die ComboBox-Komponente besteht aus einem Textfeld und einer Liste, sodass es sich hier ebenfalls um eine listengestützte Komponente handelt.

Eine Liste besteht aus Zeilen. Ein Datenraster (DataGrid) und eine gekachelte Liste (TileList) setzen sich aus Zeilen zusammen, die sich in mehrere Spalten aufteilen lassen. Die Schnittmenge einer Zeile und einer Spalte bildet eine Zelle. In einer Liste, die im Grunde eine einzelne Spalte mit Zeilen ist, ist jede Zeile eine Zelle. Eine Zelle hat zwei wichtige Aspekte:

- Die Datenwerte in den Zellen heißen Elemente. Ein *Element* ist ein ActionScript-Objekt, das zum Speichern der Informationseinheiten in der Liste dient. Eine Liste kann als ein Array aufgefasst werden; jede indizierte Position des Arrays entspricht einem Element. In einer Liste ist ein Element ein Objekt, das in der Regel eine Eigenschaft `label` (die angezeigt wird) und eine Eigenschaft `data` (die zur Datenspeicherung verwendet wird) besitzt. Ein *Datenprovider* ist ein Datenmodell der Elemente in einer Liste. Eine listengestützte Komponente kann ihren Inhalt aus einem Datenprovider beziehen, indem dieser der `dataProvider`-Eigenschaft der Komponente zugewiesen wird.
- Eine Zelle kann ganz unterschiedliche Arten von Daten aufnehmen, von Text über Bilder, Movieclips bis hin zu jeder Klasse, die Sie erstellen können. Aus diesem Grund muss eine Zelle auf eine Weise gezeichnet oder gerendert werden, die ihrem Inhalt entspricht. Und daher verfügen listengestützte Komponenten zum Rendern der Zellen über einen *Zellrenderer*. Bei einem Datenraster ist jede Spalte ein `DataGridColumn`-Objekt, das ebenfalls eine `cellRenderer`-Eigenschaft besitzt, sodass jede Spalte gemäß ihrem Inhalt gerendert werden kann.

Alle List-basierten Komponenten besitzen die Eigenschaften `cellRenderer` und `dataProvider`, die Sie für das Laden und Rendern der Zellen dieser Komponenten einstellen können. Wie Sie mit diesen Eigenschaften und mit List-basierten Komponenten arbeiten, erfahren Sie in den Abschnitten „Arbeiten mit einem Datenprovider“ auf Seite 31 und „Arbeiten mit einem Zellrenderer“ auf Seite 39.

Arbeiten mit einem Datenprovider

Ein Datenprovider ist eine Datenquelle, aus der Sie die Daten für die Komponenten „ComboBox“, „DataGrid“, „List“ und „TileList“ beziehen können. Jede dieser Komponentenklassen besitzt eine `dataProvider`-Eigenschaft, der Sie ein `DataProvider`-Objekt zuweisen können, um die Zellen der Komponente mit Daten zu füllen. In der Regel ist ein solcher Datenprovider eine Sammlung von Daten, wie etwa ein Array oder ein XML-Objekt.

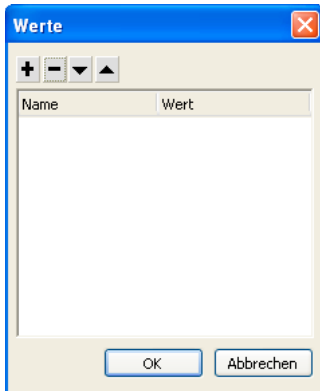
Erstellen eines Datenproviders

In der Authoring-Umgebung können Sie für die Komponenten „ComboBox“, „List“ und „TileList“ mithilfe des Parameters `dataProvider` einen Datenprovider erstellen. Die Komponente „DataGrid“ verfügt im Eigenschafteninspektor nicht über den Parameter „dataProvider“, da sie mehrere Spalten haben kann und der Datenprovider deshalb komplexer sein muss. Mithilfe von ActionScript können Sie für diese Komponenten und auch für ein Datenraster einen Datenprovider erstellen.

Verwenden des Parameters „dataProvider“

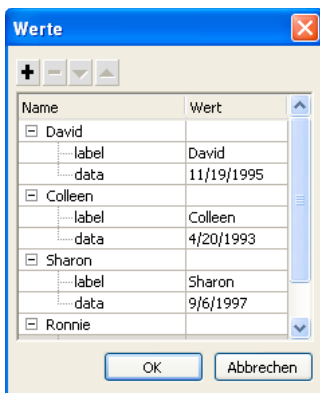
Sie können einen einfachen Datenprovider für die Komponenten „ComboBox“, „List“ und „TileList“ erstellen, indem Sie auf der Registerkarte „Parameter“ des Eigenschaften- oder Komponenten-Inspektors auf den Parameter „dataProvider“ klicken.

Wenn Sie auf die Zelle „Wert“, die anfangs ein leeres Array zeigt, doppelklicken, wird das Dialogfeld „Werte“ geöffnet, in dem Sie mehrere Beschriftungs- und Datenwerte eingeben können, um den Datenprovider zu erstellen.



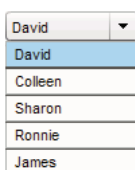
Im Dialogfeld „Werte“ können Sie einen Datenprovider definieren

Klicken Sie auf das Pluszeichen, um in den Datenprovider ein Element einzufügen. Klicken Sie auf das Minuszeichen, um ein Element zu löschen. Klicken Sie auf den nach oben weisenden Pfeil, um ein ausgewähltes Element innerhalb der Liste nach oben zu verlagern bzw. auf den nach unten weisenden Pfeil, um das Element um eine Position nach unten zu verlagern. In der folgenden Abbildung sehen Sie das Dialogfeld „Werte“, mit dem eine Liste mit Namen und Geburtstagen von Kindern erstellt wird.



Das Dialogfeld „Werte“ mit Daten

Das Array, das Sie hier erstellen, setzt sich aus je einem Beschriftungs- und einem Wertefeld zusammen. Zur Beschriftung dienen die Felder `label` und `data` und in den Wertefeldern sind die Namen der Kinder und ihre Geburtstage eingetragen. Das Feld „label“ identifiziert den Inhalt, der in der Liste angezeigt wird, in diesem Fall die Namen der Kinder. Das entstandene Kombinationsfeld sieht wie folgt aus:



Das durch den Datenprovider gefüllte Kombinationsfeld

Arbeiten mit Komponenten

Wenn Sie alle Daten hinzugefügt haben, klicken Sie auf „OK“, um das Dialogfeld zu schließen. Das Array im Parameter „dataProvider“ ist jetzt mit den von Ihnen erstellten Elementen gefüllt.

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

Die Daten als Array im Parameter „dataProvider“

Mithilfe von ActionScript können Sie auf die Beschriftungen und Datenwerte der dataProvider-Eigenschaft der Komponente zugreifen.

Erstellen eines Datenproviders mithilfe von ActionScript-Code

Sie können einen Datenprovider erstellen, indem Sie die Daten in einem Array- oder XML-Objekt erstellen und das Objekt im Parameter value dem DataProvider-Konstruktor zur Verfügung stellen.

Hinweis: In ActionScript 3.0 können Sie ein Array- oder XML-Objekt direkt einer dataProvider-Eigenschaft zuweisen, da die Eigenschaft als DataProvider-Objekt definiert ist und nur ein Objekt vom Typ „DataProvider“ annehmen kann.

Im folgenden Beispiel wird eine List-Komponente, also eine einzelne Spalte mit Zeilen, mit den Namen mehrerer Kinder und ihrer Geburtstage gefüllt. Im Beispiel wird die Liste im items-Array definiert und als Parameter übergeben, wenn die DataProvider-Instanz erstellt wird (`new DataProvider(items)`). Die Liste wird dann der dataProvider-Eigenschaft der List-Komponente zugewiesen.

```
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);
```

Das Array setzt sich aus Paaren mit je einem Beschriftungs- und einem Wertefeld zusammen. Zur Beschriftung dienen die Felder label und data und in den Wertefeldern sind die Namen der Kinder und ihre Geburtstage eingetragen. Das Feld „label“ identifiziert den Inhalt, der in der Liste angezeigt wird, in diesem Fall die Namen der Kinder. Die entstandene Liste sieht wie folgt aus:

David
Colleen
Sharon
Ronnie
James

Eine durch einen Datenprovider gefüllte Liste

Der Wert des Datenfeld ist verfügbar, wenn der Benutzer ein Element aus der Liste auswählt und dadurch ein change-Ereignis auslöst. Im nächsten Beispiel wird das vorige Beispiel durch den Textbereich aTa und die Ereignisprozedur changeHandler ergänzt, sodass bei Auswahl eines Namens in der Liste der Geburtstag des Kindes angezeigt wird.

```
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);

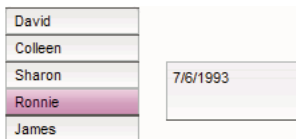
addChild(aList);
addChild(aTa);

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

aList.addEventListener(Event.CHANGE, changeHandler);

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

Wenn nun ein Benutzer den Namen eines Kindes aus der Liste auswählt, wird dessen Geburtstag im Textbereich angezeigt (siehe Abbildung). Dies wird durch die `changeHandler()`-Funktion erreicht, wenn diese die `text`-Eigenschaft der `TextArea`-Instanz (`aTa.text`) auf den Wert des Datenfelds im ausgewählten Element setzt (`event.target.selectedItem.data`). Die `event.target`-Eigenschaft ist das Objekt, welches das Ereignis ausgelöst hat, in diesem Fall die `List`-Instanz.



Der Wert des Datenproviderfelds „data“ wird nun ebenfalls angezeigt

Sie können nicht nur Text in einen Datenprovider einbinden. Im folgenden Beispiel werden Movieclips in einen Datenprovider eingebunden, der die Daten für eine gekachelte Liste (`TileList`) liefert. Der Datenprovider wird durch Aufruf von `addItem()` erstellt, d. h., jedes Element wird nach der Erstellung des Movieclips, eines farbigen Felds, hinzugefügt.

```
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();
}
```

Anstelle eines Arrays können Sie auch XML-Daten verwenden, um ein DataProvider-Objekt zu füllen. Durch den folgenden Code etwa werden Daten in dem XML-Objekt `employeesXML` gespeichert, und anschließend wird dieses Objekt als `value`-Parameter der Konstrukturfunktion `DataProvider()` übergeben:

```
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;
```

Sie können, wie im vorigen Code, Daten als Attribute von XML-Daten bereitstellen oder als Eigenschaften der XML-Daten, wie im folgenden Code:

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

Der Datenprovider verfügt ebenfalls über ein Set an Methoden und Eigenschaften, auf die Sie zugreifen und die Sie ändern können. Sie können mithilfe der DataProvider-API Elemente zu einem Datenprovider hinzufügen, daraus entfernen bzw. sie ersetzen, sortieren und verbinden.

Ändern eines Datenproviders

Mit den Methoden `addItem()` und `addItemAt()` können Sie neue Elemente in einen Datenprovider einfügen. Im folgenden Beispiel werden Elemente hinzugefügt, die ein Benutzer in das Textfeld eines bearbeitbaren Kombinationsfelds einfügt. Es wird davon ausgegangen, dass ein Kombinationsfeld auf die Bühne gezogen wurde und den Instanznamen `aCb` erhielt.

```
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);
}
```

Sie können Elemente in einer Komponente auch mithilfe ihres Datenproviders entfernen. Im folgenden Beispiel werden zwei separate List-Komponenten, `listA` und `listB`, implementiert und es wird eine Schaltfläche mit der Beschriftung „Sync“ bereitgestellt. Klickt ein Benutzer auf die Schaltfläche, wird im Beispiel die Methode `replaceItemAt()`, verwendet, um die Elemente in `listB` durch die Elemente in `listA` zu ersetzen. Ist `listA` länger als `listB`, so wird die Methode `addItem()` aufgerufen, um die zusätzlichen Elemente in `listB` einzufügen. Ist `listB` länger als `listA`, so wird die Methode `removeItemAt()` aufgerufen, um die zusätzlichen Elemente aus `listB` zu entfernen.


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

Mit den Methoden `merge()`, `sort()` und `sortOn()` ist das Zusammenführen mit einem anderen Datenprovider und das Sortieren der Elemente möglich. Im folgenden Beispiel werden zwei `DataGrid`-Instanzen (`aDg` und `bDg`) mit den Mitgliederlisten von zwei Softballteams gefüllt. Außerdem wird die Schaltfläche mit der Beschriftung „Merge“ (Zusammenführen) hinzugefügt. Wenn der Benutzer darauf klickt, verbindet die Ereignisprozedur `mrgHandler` die Mitgliederliste aus `bDg` mit der Mitgliederliste aus `aDg` und sortiert die Daten der `DataGrid`-Komponente anhand der Spalte „Name“.

```
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");
}
```

Weitere Informationen finden Sie im Abschnitt zur DataProvider-Klasse im [ActionScript 3.0 Referenzhandbuch](#).

Arbeiten mit einem Zellrenderer

„CellRenderer“ ist eine Klasse, die von List-basierten Komponenten wie „List“, „DataGrid“, „TileList“ und „ComboBox“ verwendet wird, um benutzerdefinierten Zellinhalt für deren Zeilen zu ändern und anzuzeigen. Benutzerdefinierte Zellen können Text, vorgefertigte Komponenten, wie ein Kontrollkästchen, oder jede sonstige Anzeigeklassen, die Sie erstellen können, enthalten. Um Daten mithilfe eines benutzerdefinierten Zellrenderers (CellRenderer) zu rendern, können Sie entweder die CellRenderer-Klasse erweitern oder die ICellRenderer-Schnittstelle implementieren, um eine eigene benutzerdefinierte CellRenderer-Klasse zu erstellen.

Die List-, DataGrid-, TileList- und ComboBox-Klassen sind Unterklassen der SelectableList-Klasse. Die SelectableList-Klasse umfasst ein cellRenderer-Format. Dieses Format definiert das Anzeigeklassen, das die Komponente für das Rendern der Zellen verwendet.

Sie können die Formatierung der vom Zellrenderer verwendeten Formate ändern, indem Sie die Methode `setRendererStyle()` des List-Objekts aufrufen (siehe „[Formatieren von Zellen](#)“ auf Seite 39). Sie können auch eine benutzerdefinierte Klasse als Zellrenderer definieren (siehe „[Definieren benutzerdefinierter CellRenderer-Klassen](#)“ auf Seite 40).

Formatieren von Zellen

Die CellRenderer-Klasse umfasst eine Reihe von Formaten, mit denen Sie die Formatierung der Zellen steuern können.

Die folgenden Formate erlauben die Definition der für die verschiedenen Zustände der Zellen („disabled“, „down“, „over“ und „up“) verwendeten Skins:

- `disabledSkin` und `selectedDisabledSkin`
- `downSkin` und `selectedDownSkin`
- `overSkin` und `selectedOverSkin`

- `upSkin` und `selectedUpSkin`

Die folgenden Formate dienen der Textformatierung:

- `disabledTextFormat`
- `textFormat`
- `textPadding`

Sie können diese Formate festlegen, indem Sie die Methode `setRendererStyle()` des `List`-Objekts oder die Methode `setStyle()` des `CellRenderer`-Objekts aufrufen. Sie können diese Formate abrufen, indem Sie die Methode `getRendererStyle()` des `List`-Objekts oder die Methode `getStyle()` des `CellRenderer`-Objekts aufrufen. Über die Eigenschaft `rendererStyles` des `List`-Objekts oder die Methode `getStyleDefinition()` des `CellRenderer`-Objekts können Sie auch auf ein Objekt zugreifen, das alle `Renderer`formate als benannte Eigenschaften des Objekts definiert.

Sie können die Methode `clearRendererStyle()` aufrufen, um ein Format auf seinen Standardwert zurückzusetzen.

Um die Höhe der Zeilen in der Liste abzurufen oder festzulegen, verwenden Sie die Eigenschaft `rowHeight` des `List`-Objekts.

Definieren benutzerdefinierter `CellRenderer`-Klassen

Definition durch Erstellen einer Klasse, die die `CellRenderer`-Klasse erweitert

Der folgende Code enthält zum Beispiel zwei Klassen. Die `ListSample`-Klasse instanziiert eine `List`-Komponente und verwendet die andere Klasse, `CustomRenderer`, um den für die `List`-Komponente zu verwendenden `Zellrenderer` zu definieren. Die `CustomRenderer`-Klasse erweitert die `CellRenderer`-Klasse.

- 1 Wählen Sie „Datei“ > „Neu“.
- 2 Wählen Sie im Dialogfeld „Neues Dokument“ die Option „Flash-Datei“ (ActionScript 3.0) aus und klicken Sie dann auf „OK“.
- 3 Wählen Sie „Fenster“ > „Komponenten“, um das Bedienfeld „Komponenten“ zu öffnen.
- 4 Ziehen Sie die `List`-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
- 5 Wenn der `Eigenschafteninspektor` nicht angezeigt wird, wählen Sie „Fenster“ > „Eigenschaften“ > „Eigenschaften“.
- 6 Während die `List`-Komponente ausgewählt ist, legen Sie die `Eigenschaften` im `Eigenschafteninspektor` fest:
 - Instanzname: `myList`
 - B (Breite): 200
 - H (Höhe): 300
 - X: 20
 - Y: 20
- 7 Wählen Sie in der Zeitleiste das Bild 1 der Ebene 1 aus und wählen Sie „Fenster“ > „Aktionen“.
- 8 Geben Sie im Bedienfeld „Aktionen“ den folgenden Code ein:

```
myList.setStyle("cellRenderer", CustomCellRenderer);
myList.addItem({label:"Burger -- $5.95"});
myList.addItem({label:"Fries -- $1.95"});
```
- 9 Wählen Sie „Datei“ > „Speichern“. Geben Sie einen Dateinamen ein und klicken Sie auf „OK“.
- 10 Wählen Sie „Datei“ > „Neu“.
- 11 Wählen Sie im Dialogfeld „Neues Dokument“ die Option „ActionScript-Datei“ aus und klicken Sie dann auf „OK“.

12 Geben Sie den folgenden Code in das Skriptfenster ein, um die CustomCellRenderer-Klasse zu definieren.

```
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 Wählen Sie „Datei“ > „Speichern“. Geben Sie der Datei den Namen „CustomCellRenderer.as“, legen Sie sie im selben Verzeichnis wie die FLA-Datei ab und klicken Sie auf „OK“.

14 Wählen Sie „Steuerung“ > „Film testen“.

Definition durch eine Klasse, die die ICellRenderer-Schnittstelle implementiert

Sie können einen Zellrenderer auch mithilfe jeder Klasse definieren, die die DisplayObject-Klasse erbt und die ICellRenderer-Schnittstelle definiert. Mit dem folgenden Code werden zum Beispiel zwei Klassen definiert. Die ListSample2-Klasse fügt ein List-Objekt in die Anzeigeliste ein und definiert seinen Zellrenderer so, dass er die CustomRenderer-Klasse verwendet. Die CustomRenderer-Klasse erweitert die CheckBox-Klasse (welche die DisplayObject-Klasse erweitert) und implementiert die ICellRenderer-Schnittstelle. Beachten Sie, dass die CustomRenderer-Klasse Get-/Set-Methoden für die Eigenschaften data und listData definiert, die in der ICellRenderer-Schnittstelle definiert sind. Andere in der ICellRenderer-Schnittstelle definierte Eigenschaften und Methoden (die Eigenschaft selected und die Methode setSize()) sind bereits in der CheckBox-Klasse definiert:

- 1 Wählen Sie „Datei“ > „Neu“.
- 2 Wählen Sie im Dialogfeld „Neues Dokument“ die Option „Flash-Datei“ (ActionScript 3.0) aus und klicken Sie dann auf „OK“.
- 3 Wählen Sie „Fenster“ > „Komponenten“, um das Bedienfeld „Komponenten“ zu öffnen.
- 4 Ziehen Sie die List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
- 5 Wenn der Eigenschafteninspektor nicht angezeigt wird, wählen Sie „Fenster“ > „Eigenschaften“ > „Eigenschaften“.
- 6 Während die List-Komponente ausgewählt ist, legen Sie die Eigenschaften im Eigenschafteninspektor fest:
 - Instanzname: myList
 - B (Breite): 100
 - H (Höhe): 300
 - X: 20
 - Y: 20
- 7 Wählen Sie in der Zeitleiste das Bild 1 der Ebene 1 aus und wählen Sie „Fenster“ > „Aktionen“.
- 8 Geben Sie im Bedienfeld „Aktionen“ den folgenden Code ein:

```
myList.setStyle("cellRenderer", CustomCellRenderer);
myList.addItem({name:"Burger", price:"$5.95"});
myList.addItem({name:"Fries", price:"$1.95"});
```
- 9 Wählen Sie „Datei“ > „Speichern“. Geben Sie einen Dateinamen ein und klicken Sie auf „OK“.

10 Wählen Sie „Datei“ > „Neu“.

11 Wählen Sie im Dialogfeld „Neues Dokument“ die Option „ActionScript-Datei“ aus und klicken Sie dann auf „OK“.

12 Geben Sie den folgenden Code in das Skriptfenster ein, um die CustomCellRenderer-Klasse zu definieren.

```
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 Wählen Sie „Datei“ > „Speichern“. Geben Sie der Datei den Namen „CustomCellRenderer.as“, legen Sie sie im selben Verzeichnis wie die FLA-Datei ab und klicken Sie auf „OK“.

14 Wählen Sie „Steuerung“ > „Film testen“.

Definition durch ein Bibliothekssymbol

Sie können einen Zellrenderer auch mithilfe eines Symbols in der Bibliothek definieren. Das Symbol muss für ActionScript exportiert werden und der Klassenname für das Bibliothekssymbol muss mit einer Klassendatei verknüpft sein, die entweder die ICellRenderer-Schnittstelle implementiert oder die CellRenderer-Klasse (oder eine ihrer Unterklassen) erweitert.

Im folgenden Beispiel wird ein benutzerdefinierter Zellrenderer mithilfe eines Bibliothekssymbols definiert.

1 Wählen Sie „Datei“ > „Neu“.

2 Wählen Sie im Dialogfeld „Neues Dokument“ die Option „Flash-Datei“ (ActionScript 3.0) aus und klicken Sie dann auf „OK“.

3 Wählen Sie „Fenster“ > „Komponenten“, um das Bedienfeld „Komponenten“ zu öffnen.

4 Ziehen Sie die List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.

5 Wenn der Eigenschafteninspektor nicht angezeigt wird, wählen Sie „Fenster“ > „Eigenschaften“ > „Eigenschaften“.

6 Während die List-Komponente ausgewählt ist, legen Sie die Eigenschaften im Eigenschafteninspektor fest:

- Instanzname: myList
- B (Breite): 100

- H (Höhe): 400
 - X: 20
 - Y: 20
- 7 Klicken Sie auf die Registerkarte „Parameter“ und doppelklicken Sie auf die zweite Spalte in der Zeile „dataProvider“.
- 8 Klicken Sie im Dialogfeld „Werte“ zwei Mal auf das Pluszeichen, um zwei Datenelemente einzufügen (deren Beschriftungen Sie auf „label0“ und „label1“ setzen), und klicken Sie auf „OK“.
- 9 Zeichnen Sie mit dem Textwerkzeug ein Textfeld auf der Bühne.
- 10 Während das Textfeld ausgewählt ist, legen Sie die Eigenschaften im Eigenschafteninspektor fest:
- Texttyp: Dynamischer Text
 - Instanzname: textField
 - B (Breite): 100
 - Schriftgröße: 24
 - X: 0
 - Y: 0
- 11 Während das Textfeld noch ausgewählt ist, wählen Sie „Modifizieren“ > „In Symbol konvertieren“.
- 12 Legen Sie im Dialogfeld „In Symbol konvertieren“ die folgenden Einstellungen fest und klicken Sie auf „OK“.
- Name: MyCellRenderer
 - Typ: MovieClip
 - Export für ActionScript: ausgewählt
 - Export in Bild 1: ausgewählt
 - Klasse: MyCellRenderer
 - Basisklasse: flash.display.MovieClip
- Falls Flash eine ActionScript-Klasse-Warnung anzeigt, klicken Sie im Dialogfeld mit der Warnmeldung auf „OK“.
- 13 Löschen Sie die Instanz des neuen Movieclipsymbols von der Bühne.
- 14 Wählen Sie in der Zeitleiste das Bild 1 der Ebene 1 aus und wählen Sie „Fenster“ > „Aktionen“.
- 15 Geben Sie im Bedienfeld „Aktionen“ den folgenden Code ein:
- ```
myList.setStyle("cellRenderer", MyCellRenderer);
```
- 16 Wählen Sie „Datei“ > „Speichern“. Geben Sie einen Dateinamen ein und klicken Sie auf „OK“.
- 17 Wählen Sie „Datei“ > „Neu“.
- 18 Wählen Sie im Dialogfeld „Neues Dokument“ die Option „ActionScript-Datei“ aus und klicken Sie dann auf „OK“.
- 19 Geben Sie in das Skriptfenster den folgenden Code ein, um die MyCellRenderer-Klasse zu definieren.

```
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 Wählen Sie „Datei“ > „Speichern“. Geben Sie der Datei den Namen „MyCellRenderer.as“, legen Sie sie im selben Verzeichnis wie die FLA-Datei ab und klicken Sie auf „OK“.

21 Wählen Sie „Steuerung“ > „Film testen“.



## Eigenschaften der CellRenderer-Klasse

Die Eigenschaft `data` ist ein Objekt, das alle Eigenschaften enthält, die für den Zellrenderer festgelegt worden sind. In der folgenden Klasse etwa, die einen benutzerdefinierten Zellrenderer definiert, der die `CheckBox`-Klasse erweitert, übergibt die `set`-Funktion für die Eigenschaft `data` den Wert `data.label` an die Eigenschaft `label`, die von der `CheckBox`-Klasse geerbt wurde:

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

Die Eigenschaft `selected` definiert, ob eine Zelle in der Liste ausgewählt ist oder nicht.

## Anwenden von Zellrenderern auf Datenrasterzellen

Ein `DataGrid`-Objekt kann mehrere Spalten enthalten und Sie können für jede Spalte verschiedene `CellRenderer` festlegen. Jede Spalte einer `DataGrid`-Komponente wird von einem `DataGridColumn`-Objekt repräsentiert. Die `DataGridColumn`-Klasse enthält die Eigenschaft `cellRenderer`, für die Sie einen `CellRenderer` für die Spalte definieren können.

## Definieren von Zellrenderern für bearbeitbare Zellen

Die `DataGridCellEditor`-Klasse definiert einen `Renderer`, der für bearbeitbare Zellen in einem `DataGrid`-Objekt verwendet wird. Er wird zum `Renderer` für eine Zelle, wenn die Eigenschaft `editable` des `DataGrid`-Objekts auf `true` gesetzt ist und der Benutzer auf die betreffende Zelle klickt. Um einen `CellRenderer` für die bearbeitbare Zelle zu definieren, setzen Sie die Eigenschaft `itemEditor` für jedes Element des `columns`-Arrays des `DataGrid`-Objekts.

## Verwenden eines Bilds, einer SWF-Datei oder eines Movieclips als Zellrenderer

Die `ImageCell`-Klasse eine Unterklasse von `CellRenderer`, definiert ein Objekt für das Rendern von Zellen, in denen der Hauptinhalt der Zelle ein Bild, eine SWF-Datei oder ein `Movieclip` ist. Die `ImageCell`-Klasse umfasst die folgenden Formate für die Definition der Zelldarstellung:

- `imagePadding`: Der Rand zwischen Zelle und Bild, also der Abstand zwischen Zell- und Bildbegrenzung (in Pixel).
- `selectedSkin`: Die `Skin`, die verwendet wird, um den ausgewählten Zustand anzuzeigen.

- `textOverlayAlpha`: Die Deckkraft der Überlagerung hinter der Zellenbeschriftung.
  - `textPadding`: Der Rand zwischen Zelle und Text, also der Abstand zwischen Zell- und Textbegrenzung (in Pixel).
- Die `ImageCell`-Klasse ist der Standardzellrenderer für die `TileList`-Klasse.

## Barrierefreiheit von Komponenten

Sie können visuellen Inhalt in Ihren Flash-Anwendungen auch sehbehinderten Menschen zugänglich machen, und zwar über Bildschirmleseprogramme, die eine Beschreibung des Bildschirminhalts vorlesen. Wie Sie Ihre Flash-Anwendung einem Bildschirmleseprogramm zugänglich machen, erfahren Sie im Handbuch *Verwenden von Flash* im Kapitel 18, „Barrierefreie Inhalte erstellen“.

Um eine ActionScript 3.0-Komponente einem Bildschirmleseprogramm zugänglich zu machen, müssen Sie auch ihre Accessibility-Klasse importieren und die `enableAccessibility()`-Methode dieser Klasse aufrufen. Folgende ActionScript 3.0-Komponenten können für Bildschirmleseprogramme zugänglich gemacht werden:

| Komponente  | Accessibility-Klasse |
|-------------|----------------------|
| Button      | ButtonAccImpl        |
| CheckBox    | CheckBoxAccImpl      |
| ComboBox    | ComboBoxAccImpl      |
| List        | ListAccImpl          |
| RadioButton | RadioButtonAccImpl   |
| TileList    | TileListAccImpl      |

Die Komponentenzugänglichkeitsklassen befinden sich im Paket `fl.accessibility`. Um ein Kontrollkästchen für ein Bildschirmleseprogramm zugänglich zu machen, würden Sie die folgenden Anweisungen in Ihre Anwendung einfügen:

```
import fl.accessibility.CheckBoxAccImpl;

CheckBoxAccImpl.enableAccessibility();
```

Die Zugänglichkeit wird pro Komponente nur einmal aktiviert, auch wenn Sie mehrere Instanzen davon erstellen.

**Hinweis:** Durch die Implementierung einer solchen Barrierefreiheit nimmt die Dateigröße in geringem Maße zu, da die erforderlichen Klassen beim Kompilieren mit eingebunden werden.

Die meisten Komponenten sind auch über die Tastatur bedienbar. Weitere Informationen zum Verwenden barrierefreier Komponenten und zum Navigieren mithilfe der Tastatur finden Sie in den Abschnitten zur Benutzerinteraktion unter „[Verwenden der UI-Komponenten](#)“ auf Seite 47 und in den Abschnitten zu den Accessibility-Klassen im *ActionScript 3.0 Referenzhandbuch für Flash Professional*.

# Kapitel 4: Verwenden der UI-Komponenten

In diesem Kapitel wird erläutert, wie Sie die in Flash integrierten ActionScript 3.0-Komponenten für Benutzeroberflächen (UI f. engl. „User Interface“) einsetzen.

## Arbeiten mit der Button-Komponente

Die Button-Komponente ist eine rechteckige Schaltfläche mit veränderlicher Größe, die der Benutzer mit der Maus oder mit der Leertaste aktiviert, um eine Aktion in der Anwendung auszuführen. Sie können Schaltflächen eigene Symbole zuweisen. Außerdem können Sie das Verhalten von Schaltflächen ändern und herkömmliche Schaltflächen in Schaltflächen mit Umschaltfunktion umwandeln. Eine Schaltfläche mit Umschaltfunktion (Toggle-Schaltfläche) bleibt nach dem Klicken so lange aktiviert, bis der Benutzer erneut darauf klickt.

Schaltflächen sind grundlegende Bestandteile vieler Formulare und Webanwendungen. Mit Schaltflächen können Sie dem Benutzer die Möglichkeit geben, Ereignisse einzuleiten. Die meisten Formulare enthalten beispielsweise eine Schaltfläche „Senden“, mit der die Übertragung der eingegebenen Daten an den Server eingeleitet wird. In Präsentationen ermöglichen Schaltflächen wie „Zurück“ und „Weiter“ das Navigieren zwischen den einzelnen Seiten.

## Benutzerinteraktion mit der Button-Komponente

Eine Schaltfläche in einer Anwendung können Sie aktivieren oder deaktivieren. Im deaktivierten Zustand reagiert die Schaltfläche nicht auf Maus- und Tastatureingaben. Aktivierte Schaltflächen erhalten den Eingabefokus, wenn darauf geklickt wird oder wenn sie mit der Tabulatortaste angesteuert werden. Button-Instanzen mit Eingabefokus können mit den folgenden Tasten gesteuert werden:

| Schlüssel    | Beschreibung                                                                                                                          |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Umschalt+Tab | Verschiebt den Fokus auf das vorherige Objekt.                                                                                        |
| Leertaste    | Drückt die Schaltfläche bzw. gibt diese frei und löst das Ereignis <code>click</code> aus.                                            |
| Tab          | Verschiebt den Fokus auf das nächste Objekt.                                                                                          |
| Eingabetaste | Der Fokus wird auf das nächste Objekt verschoben, falls eine Schaltfläche als Standardschaltfläche des FocusManager festgelegt wurde. |

Weitere Informationen zum Steuern des Fokus finden Sie in den Abschnitten zur IFocusManager-Schnittstelle und zur FocusManager-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#) und unter „[Arbeiten mit dem FocusManager](#)“ auf Seite 29.

Die Änderungen an den Parametern der einzelnen Button-Instanzen, die beim Authoring im Eigenschaften- oder Komponenten-Inspektor vorgenommen werden, werden jeweils in einer Live-Vorschau angezeigt.

**Hinweis:** Wenn das Symbol größer als die Schaltfläche ist, ragt es über deren Begrenzungen hinaus.

Um eine Schaltfläche in einer Anwendung zur Standardschaltfläche zu machen (d. h. zu der Schaltfläche, für die beim Drücken der Eingabetaste das Ereignis „click“ ausgelöst wird), verwenden Sie `FocusManager.defaultButton`. Im folgenden Codebeispiel wird als Standardschaltfläche eine Button-Instanz mit dem Namen `submitButton` festgelegt.

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

Die einer Anwendung hinzugefügten Button-Komponenten können Sie anhand der folgenden ActionScript-Codezeilen für Bildschirmleseprogramme zugänglich machen:

```
import fl.accessibility.ButtonAccImpl;

ButtonAccImpl.enableAccessibility();
```

Die Zugänglichkeit wird pro Komponente nur einmal aktiviert, auch wenn Sie mehrere Instanzen davon erstellen.

## Parameter der Button-Komponente

Sie können die folgenden Authoring-Parameter im Eigenschafteninspektor („Fenster“ > „Eigenschaften“ > „Eigenschaften“) oder im Komponenteninspektor („Fenster“ > „Komponenteninspektor“) für jede Button-Instanz festlegen: `emphasized`, `label`, `labelPlacement`, `selected` und `toggle`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Wenn Sie diesen Parametern einen Wert zuweisen, legen Sie den ursprünglichen Status der Eigenschaft in der Anwendung fest. Durch Festlegen der Eigenschaft in ActionScript wird der Wert, den Sie für den Parameter angeben, außer Kraft gesetzt. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur Button-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Erstellen einer Anwendung mit der Button-Komponente

Im Folgenden wird erläutert, wie Sie beim Authoring eine Button-Komponente in eine Anwendung einfügen. In diesem Beispiel ändert die Button-Komponente den Zustand einer ColorPicker-Komponente, wenn Sie darauf klicken.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine Button-Komponente vom Bedienfeld „Komponenten“ auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
  - Geben Sie den Instanznamen **aButton** ein.
  - Geben Sie **Show** für den Parameter „label“ ein.
- 3 Platzieren Sie eine ColorPicker-Komponente auf der Bühne und geben Sie ihr den Instanznamen **aCp**.
- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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;
 }
}
```

Die zweite Codezeile registriert die Funktion `clickHandler()` als Ereignisprozedurfunktion für das `MouseEvent.CLICK`-Ereignis. Das Ereignis tritt auf, wenn ein Benutzer auf die Schaltflächen klickt und somit die Funktion `clickHandler()` dazu veranlasst, je nach Schaltflächenwert, eine der folgenden Aktionen auszuführen:

- „Show“ blendet die ColorPicker-Komponente ein und ändert die Beschriftung der Schaltfläche in „Disable“.
- „Disable“ deaktiviert die ColorPicker-Komponente und ändert die Beschriftung der Schaltfläche in „Enable“.
- „Enable“ aktiviert die ColorPicker-Komponente und ändert die Beschriftung der Schaltfläche in „Hide“.
- „Hide“ blendet die ColorPicker-Komponente aus und ändert die Beschriftung der Schaltfläche in „Show“.

5 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.

## Erstellen einer Anwendung mit der Button-Komponente

Im Folgenden wird eine Schaltfläche mit Umschaltfunktion erstellt, und der Ereignistyp wird im Bedienfeld „Ausgabe“ angezeigt, wenn der Anwender auf die Schaltfläche klickt. Im Beispiel wird die Button-Instanz erstellt, indem der Konstruktor der Klasse aufgerufen wird, und sie wird mit der `addChild()`-Methode zur Bühne hinzugefügt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die Button-Komponente vom Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ des aktuellen Dokuments.

Hierdurch wird die Komponente zur Bibliothek hinzugefügt, sie wird jedoch nicht in der Anwendung angezeigt.

- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein, um eine Button-Instanz zu erstellen:

```
import fl.controls.Button;

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

Die Methode `move()` platziert die Schaltfläche auf der Bühne auf Position 50 (x-Koordinate), 50 (y-Koordinate).

- 4 Fügen Sie jetzt den folgenden ActionScript-Code hinzu, um einen Ereignis-Listener und eine Ereignisprozedur zu erstellen:

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

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

- 5 Wählen Sie „Steuerung“ > „Film testen“.

Wenn Sie auf die Schaltfläche klicken, zeigt Flash im Bedienfeld „Ausgabe“ die Meldung „Ereignistyp: click“ an.

## Arbeiten mit der CheckBox-Komponente

Eine CheckBox-Komponente (Kontrollkästchen) ist ein quadratisches Feld, das ausgewählt bzw. dessen Auswahl aufgehoben werden kann. Ausgewählte Kontrollkästchen werden durch ein Häkchen im Feld gekennzeichnet. Sie können eine Beschriftung zu einem Kontrollkästchen definieren und sie links, rechts, oben oder unten vom Kontrollkästchen platzieren.

Sie können Kontrollkästchen immer dann verwenden, wenn mehrere boolesche Werte (`true` oder `false`) abgefragt werden sollen, die sich nicht gegenseitig ausschließen. Ein Beispiel hierfür wäre ein Formular für den Autokauf, in dem anhand von Kontrollkästchen die gewünschte Sonderausstattung abgefragt wird.

### Benutzerinteraktion mit der CheckBox-Komponente

Ein Kontrollkästchen in einer Anwendung können Sie aktivieren oder deaktivieren. Wenn der Anwender auf die Beschriftung eines aktivierten Kontrollkästchens klickt, erhält dieses den Eingabefokus und wird im gedrückten Zustand angezeigt. Wenn der Mauszeiger bei gedrückter Maustaste den Begrenzungsbereich eines Kontrollkästchens oder des zugehörigen Beschriftungstexts verlässt, wird die Komponente wieder im ursprünglichen Zustand angezeigt und behält den Eingabefokus. Der Status eines Kontrollkästchens ändert sich erst, wenn die Maustaste über der Komponente losgelassen wird. Darüber hinaus kann ein Kontrollkästchen zwei Deaktiviertheitsstati annehmen: „ausgewählt“ und „Auswahl aufgehoben“, die `selectedDisabledSkin` bzw. `disabledSkin` verwenden und keine Maus- oder Tastaturinteraktion zulassen.

Deaktivierte Kontrollkästchen werden unabhängig von der Interaktion des Benutzers im deaktivierten Zustand angezeigt. Im deaktivierten Zustand reagiert das Kontrollkästchen nicht auf Maus- und Tastatureingaben.

Eine CheckBox-Instanz erhält den Eingabefokus, wenn der Benutzer darauf klickt oder die Instanz mit der Tabulatortaste ansteuert. Eine CheckBox-Instanz mit Eingabefokus lässt sich mit den folgenden Tasten steuern:

| Schlüssel    | Beschreibung                                                                                       |
|--------------|----------------------------------------------------------------------------------------------------|
| Umschalt+Tab | Verschiebt den Fokus auf das vorherige Element.                                                    |
| Leertaste    | Wählt die Komponente aus bzw. hebt ihre Auswahl auf und löst das Ereignis <code>change</code> aus. |
| Tab          | Verschiebt den Fokus auf das nächste Element.                                                      |

Weitere Informationen zum Steuern des Fokus finden Sie unter „Arbeiten mit dem FocusManager“ auf Seite 29 und im Abschnitt zur FocusManager-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

Die Änderungen an den Parametern der einzelnen CheckBox-Instanzen, die beim Authoring im Eigenschaften- oder im Komponenten-Inspektor vorgenommen werden, werden jeweils in einer Live-Vorschau angezeigt.

Die einer Anwendung hinzugefügten CheckBox-Komponenten können Sie Bildschirmleseprogrammen anhand der folgenden ActionScript-Codezeilen zugänglich machen:

```
import fl.accessibility.CheckBoxAccImpl;

CheckBoxAccImpl.enableAccessibility();
```

Diese Anweisung ist unabhängig von der Anzahl der Instanzen nur einmal pro Komponente erforderlich.

## Parameter der CheckBox-Komponente

Die folgenden Authoring-Parameter können für jede Instanz einer CheckBox-Komponente im Eigenschaften- oder Komponenteninspektor festgelegt werden: `label`, `labelPlacement` und `selected`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur CheckBox-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Erstellen einer Anwendung mit der CheckBox-Komponente

Im Folgenden wird erläutert, wie Sie beim Authoring eine CheckBox-Komponente in eine Anwendung einfügen. Als Beispiel wird der Auszug eines Darlehensantrags verwendet. Im Formular wird gefragt, ob der Antragsteller ein Eigenheim besitzt. Um mit „Ja“ zu antworten, kann der Benutzer ein Kontrollkästchen auswählen. Falls „Ja“, präsentiert das Formular zwei Optionsfelder, anhand derer der Benutzer den relativen Wert des Hauses angeben kann.

### Erstellen einer Anwendung mit der CheckBox-Komponente

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine CheckBox-Komponente vom Bedienfeld „Komponenten“ auf die Bühne.
- 3 Führen Sie im Eigenschafteninspektor die folgenden Schritte durch:
  - Geben Sie als Instanznamen **homeCh** ein.
  - Geben Sie als Breite (B) den Wert **140** ein.
  - Geben Sie für den Parameter „label“ **Own your home?** ein.
- 4 Ziehen Sie zwei RadioButton-Komponenten vom Bedienfeld „Komponenten“ auf die Bühne und platzieren Sie sie unter und rechts neben der CheckBox-Komponente. Geben Sie im Eigenschafteninspektor die folgenden Informationen ein:
  - Geben Sie als Instanznamen **underRb** und **overRb** ein.
  - Geben Sie für beide Optionsfelder als Breite (B) den Wert **120** ein.

**Verwenden der UI-Komponenten**

- Geben Sie **Under \$500,000?** für den Parameter „label“ der Instanz `underRb` ein.
- Geben Sie **Over \$500,000?** den Parameter „label“ der Instanz `overRb` ein.
- Geben Sie für beide Optionsfelder den Wert **valueGrp** für den Parameter „groupName“ ein.

5 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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;
}
```

Dieser Code erstellt eine Ereignisprozedur für ein `CLICK`-Ereignis, das die Optionsfelder `underRb` und `overRb` aktiviert, wenn das Kontrollkästchen `homeCh` ausgewählt ist, und deaktiviert sie, wenn `homeCh` nicht ausgewählt ist. Weitere Informationen finden Sie im Abschnitt zur `MouseEvent`-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

6 Wählen Sie „Steuerung“ > „Film testen“.

Im folgenden Beispiel wird die gleiche Anwendung wie im vorigen Beispiel erstellt, nur werden das Kontrollkästchen und die Optionsfelder mithilfe von ActionScript-Code erstellt.

**Erstellen eines Kontrollkästchens mithilfe von ActionScript-Code**

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die `CheckBox`-Komponente und die `RadioButton`-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ des aktuellen Dokuments. Falls das Bedienfeld „Bibliothek“ nicht angezeigt wird, öffnen Sie es, indem Sie `Strg+L` drücken oder „Fenster“ > „Bibliothek“ wählen.

Dadurch werden die Komponenten für die Anwendung bereitgestellt ohne sie auf der Bühne zu platzieren.

- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein, um Komponenteninstanzen zu erstellen und zu positionieren:



```
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?";
```

Dieser Code verwendet die Konstruktoren `CheckBox()` und `RadioButton()`, um die Komponenten zu erstellen, und die Methode `addChild()`, um sie auf der Bühne zu platzieren. Er verwendet die Methode `move()`, um die Komponenten auf der Bühne zu platzieren.

- 4 Fügen Sie jetzt den folgenden ActionScript-Code hinzu, um einen Ereignis-Listener und eine Ereignisprozedur zu erstellen:

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

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

Dieser Code erstellt eine Ereignisprozedur für das `CLICK`-Ereignis, das die Optionsfelder `underRb` und `overRb` aktiviert, wenn das Kontrollkästchen `homeCh` ausgewählt ist, und deaktiviert sie, wenn `homeCh` nicht ausgewählt ist. Weitere Informationen finden Sie im Abschnitt zur `MouseEvent`-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

- 5 Wählen Sie „Steuerung“ > „Film testen“.

## Arbeiten mit der ColorPicker-Komponente

Die `ColorPicker`-Komponente (Farbwähler) erlaubt dem Benutzer die Auswahl einer Farbe aus einer Farbfeldliste. Der Standardmodus des Farbwählers zeigt ein einzelnes Farbfeld in einer quadratischen Schaltfläche an. Wenn der Benutzer auf die Schaltfläche klickt, erscheint die Liste der verfügbaren Farben in einem Farbfeldbedienfeld, zusammen mit einem Textfeld, das den Hexadezimalwert der aktuellen Farbauswahl anzeigt.

Die im Farbwähler angezeigten Farben legen Sie fest, indem Sie die Eigenschaft `colors` der `ColorPicker`-Komponente auf die entsprechenden Farbwerte setzen.

## Benutzerinteraktion mit der ColorPicker-Komponente

Mit einem Farbwähler kann der Benutzer eine Farbe auswählen und sie einem Objekt in der Anwendung zuweisen. Angenommen, dem Benutzer soll die Möglichkeit zur Anpassung der Elemente einer Anwendung gegeben werden, etwa der Hintergrundfarbe oder der Farbe von Text. Dazu binden Sie einen Farbwähler in die Anwendung ein und wenden die vom Benutzer ausgewählte Farbe an.

Der Benutzer wählt eine Farbe aus, indem er in das entsprechende Farbfeld im Bedienfeld klickt, oder den Hexadezimalwert dieser Farbe in das Textfeld eingibt. Nachdem der Benutzer eine Farbe ausgewählt hat, können Sie die Eigenschaft `selectedColor` der ColorPicker-Komponente dazu verwenden, diese Farbe dem Text oder einem anderen Objekt in der Anwendung zuzuweisen.

Eine ColorPicker-Instanz erhält den Eingabefokus, wenn der Benutzer den Mauszeiger darauf bewegt oder die Instanz mit der Tabulatortaste ansteuert. Ist das Farbfeldbedienfeld des Farbwählers geöffnet, können Sie es anhand der folgenden Tasten steuern:

| Schlüssel         | Beschreibung                                                                        |
|-------------------|-------------------------------------------------------------------------------------|
| Startseite        | Verschiebt die Auswahl auf die erste Farbe im Farbfeldbedienfeld.                   |
| Nach-oben-Taste   | Verschiebt die Auswahl innerhalb des Farbfeldbedienfelds um eine Zeile nach oben.   |
| Nach-unten-Taste  | Verschiebt die Auswahl innerhalb des Farbfeldbedienfelds um eine Zeile nach unten.  |
| Nach-rechts-Taste | Verschiebt die Auswahl innerhalb des Farbfeldbedienfelds um eine Farbe nach rechts. |
| Nach-links-Taste  | Verschiebt die Auswahl innerhalb des Farbfeldbedienfelds um eine Farbe nach links.  |
| Ende              | Verschiebt die Auswahl auf die letzte Farbe im Farbfeldbedienfeld.                  |

## Parameter der ColorPicker-Komponente

Die folgenden Authoring-Parameter können für jede Instanz einer ColorPicker-Komponente im Eigenschaften- oder Komponenten-Inspektor festgelegt werden: `selectedColor` und `showTextField`. Jeder dieser Parameter besitzt eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur ColorPicker-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Erstellen einer Anwendung mit der ColorPicker-Komponente

Im folgenden Beispiel wird beim Authoring eine ColorPicker-Komponente in eine Anwendung eingefügt. Dabei ruft die Funktion `changeHandler()` jedes Mal, wenn die Farbe im Farbwähler geändert wird, die Funktion `drawBox()` auf, um ein neues Feld mit der neuen Farbe aufzubauen.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine ColorPicker-Komponente vom Bedienfeld „Komponenten“ auf die Mitte der Bühne und nennen Sie die Instanz **aCp**.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“.
- 5 Klicken Sie auf den Farbwähler und wählen Sie eine Farbe aus, mit der das Feld eingefärbt werden soll.

### Erstellen eines Farbwählers mithilfe von ActionScript-Code

Dieses Beispiel verwendet die Konstruktoren `ColorPicker()` und `addChild()`, um auf der Bühne einen Farbwähler zu erstellen. Die Eigenschaft `colors` wird auf die Farbwerte für Rot (0xFF0000), Grün (0x00FF00) und Blau (0x0000FF) gesetzt, um die Farben zu definieren, die im Farbwähler angezeigt werden. Es wird zudem ein Textbereich erstellt, und jedes Mal, wenn im Farbwähler eine andere Farbe ausgewählt wird, ändert der Text im Textbereich seine Farbe.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die `ColorPicker`-Komponente vom Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 3 Ziehen Sie die `TextArea`-Komponente vom Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

**Verwenden der UI-Komponenten**

```
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 Wählen Sie „Steuerung“ > „Film testen“.

## Arbeiten mit der ComboBox-Komponente

Eine ComboBox-Komponente (Kombinationsfeld) ermöglicht dem Anwender die Auswahl einer einzelnen Option aus einer Dropdownliste. Ein Kombinationsfeld kann statisch oder bearbeitbar sein. Ein bearbeitbares Kombinationsfeld ermöglicht die Eingabe von Text direkt in das Textfeld am Anfang der Liste. Wenn die Dropdownliste über den unteren Dokumentrand hinausreicht, wird sie nach oben statt nach unten geöffnet. Die ComboBox-Komponente besteht aus drei Teilkomponenten: BaseButton, TextInput und List.

Bei einem bearbeitbaren Kombinationsfeld umfasst der Kollisionsbereich nur die Schaltfläche; das Textfeld gehört nicht zum Kollisionsbereich. Bei einem statischen Kombinationsfeld umfasst der Kollisionsbereich die Schaltfläche und das Textfeld. Der Kollisionsbereich reagiert durch Öffnen oder Schließen der Dropdownliste.

Nimmt der Anwender eine Auswahl in der Liste vor, sei es mit der Maus oder der Tastatur, wird die Beschriftung der Auswahl in das Textfeld am Anfang des Kombinationsfelds kopiert.

## Benutzerinteraktion mit der ComboBox-Komponente

Sie können die ComboBox-Komponente in Formularen und Anwendungen verwenden, in denen ein Element aus einer Liste auszuwählen ist. Ein Beispiel wäre ein Formular zur Eingabe der Anschrift, in dem der Kunde die Region seines Wohnorts über eine Dropdownliste auswählt. Mit bearbeitbaren Kombinationsfeldern lassen sich auch komplexere Dateneingaben erfassen. Ein Beispiel hierfür wäre ein Formular zur Anforderung einer Wegbeschreibung, bei dem Ausgangs- und Zielort in ein bearbeitbares Kombinationsfeld eingegeben werden. Mithilfe der Dropdownliste könnten sich dabei bei früheren Anfragen eingetragene Angaben abrufen lassen.

Ist das Kombinationsfeld bearbeitbar, d. h., die Eigenschaft `editable` ist `true`, kann mit den folgenden Tasten der Fokus aus dem Texteingabefeld genommen werden, sodass der vorherige Wert erhalten bleibt. Eine Ausnahme stellt die Eingabetaste dar, mit der zuerst der neue Wert übernommen wird, sofern der Benutzer Text eingeben hat.

| Schlüssel        | Beschreibung                                                                                                                                                                                                                           |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Umschalt+Tab     | Verschiebt den Fokus auf das vorherige Element. Ist ein neues Element ausgewählt, wird ein <code>change</code> -Ereignis ausgelöst.                                                                                                    |
| Tab              | Verschiebt den Fokus auf das nächste Element. Ist ein neues Element ausgewählt, wird ein <code>change</code> -Ereignis ausgelöst.                                                                                                      |
| Nach-unten-Taste | Verschiebt die Auswahl um ein Element nach unten.                                                                                                                                                                                      |
| Ende             | Verschiebt die Auswahl an den Anfang der Liste.                                                                                                                                                                                        |
| Esc              | Schließt die Dropdownliste und gibt den Fokus zurück an das Kombinationsfeld.                                                                                                                                                          |
| Eingabetaste     | Schließt die Dropdownliste und gibt den Fokus zurück an das Kombinationsfeld. Wenn das Kombinationsfeld bearbeitbar ist und der Benutzer Text eingegeben hat, wird der Wert mit der Eingabetaste auf den eingegebenen Text festgelegt. |
| Startseite       | Verschiebt die Auswahl an den Anfang der Liste.                                                                                                                                                                                        |
| Bild-auf         | Verschiebt die Auswahl um eine Seite nach oben.                                                                                                                                                                                        |
| Bild-ab          | Verschiebt die Auswahl um eine Seite nach unten.                                                                                                                                                                                       |

Die einer Anwendung hinzugefügten ComboBox-Komponenten können Sie Bildschirmleseprogrammen anhand der folgenden ActionScript-Codezeilen zugänglich machen:

```
import fl.accessibility.ComboBoxAccImpl;

ComboBoxAccImpl.enableAccessibility();
```

Diese Anweisung ist unabhängig von der Anzahl der Instanzen nur einmal pro Komponente erforderlich.

## Parameter der ComboBox-Komponente

Die folgenden Parameter können für jede Instanz einer ComboBox-Komponente im Eigenschaften- oder Komponenten-Inspektor festgelegt werden: `dataProvider`, `editable`, `prompt` und `rowCount`. Jeder dieser Parameter besitzt eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur ComboBox-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#). Informationen über die Verwendung des Parameters „`dataProvider`“ finden Sie im Abschnitt [„Verwenden des Parameters „dataProvider““](#) auf Seite 31.

## Erstellen einer Anwendung mit der ComboBox-Komponente

Im Folgenden wird beschrieben, wie Sie beim Authoring eine ComboBox-Komponente in eine Anwendung einfügen. Das Kombinationsfeld ist bearbeitbar und wenn Sie **Add** in das Textfeld eingeben, wird ein Element in die Dropdownliste eingefügt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die ComboBox-Komponente auf die Bühne und geben Sie ihr den Instanznamen **aCb**. Setzen Sie im Komponenten-Inspektor auf der Registerkarte **Parameter** den Parameter **editable** auf **true**.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“.

## Erstellen eines Kombinationsfelds mithilfe von ActionScript-Code

Im folgenden Beispiel wird mithilfe von ActionScript-Code ein Kombinationsfeld erstellt und mit einer Liste von Universitäten in der Region San Francisco, Kalifornien, gefüllt. Die Eigenschaft **width** (Breite) wird eingestellt, um genügend Platz für den Aufforderungstextes bereitzustellen und die Eigenschaft **dropdownWidth** wird so eingestellt, dass die Dropdownliste etwas größer als der längste Universitätsname ist.

In dem Beispiel wird die Liste der Universitäten in Form einer Array-Instanz erstellt, wobei die Uninamen in der Eigenschaft **label** und die URLs der verschiedenen Universitätswebsites in der Eigenschaft **data** gespeichert werden. Der Array wird dem Kombinationsfeld zugewiesen, indem die Eigenschaft **dataProvider** gesetzt wird.

Wenn der Anwender eine Universität aus der Liste auswählt, wird ein **Event.CHANGE**-Ereignis ausgelöst sowie der Aufruf der Funktion **changeHandler()**, wodurch die Eigenschaft **data** in eine URL-Anforderung geladen wird, um auf die Website der Uni zuzugreifen.

Beachten Sie, dass die letzte Zeile die Eigenschaft **selectedIndex** der ComboBox-Instanz auf „-1“ setzt, sodass die Aufforderung wieder angezeigt wird, wenn sich die Liste schließt. Andernfalls würde der Aufforderungstext durch den Namen der ausgewählten Universität ersetzt werden.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).

- 2 Ziehen Sie die ComboBox-Komponente vom Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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.csuhayward.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 Wählen Sie „Steuerung“ > „Film testen“.

Sie können dieses Beispiel in die Flash-Authoring-Umgebung implementieren und es dort ausführen. Allerdings werden Sie eine Warnmeldung erhalten, wenn Sie versuchen, die Universitätswebsites zu öffnen, indem Sie auf die Listeneinträge des Kombinationsfelds klicken. Ein voll funktionsfähiges Kombinationsfeld wurde Ihnen im Internet bereitgestellt und zwar unter folgender URL-Adresse:

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

## Arbeiten mit der DataGrid-Komponente

Eine DataGrid-Komponente (Datenraster) erlaubt Ihnen die Anzeige von Daten in einem Raster aus Zeilen und Spalten, wobei die Daten aus einem Array bezogen werden oder aus einer externen XML-Datei, die von einem Parser analysiert und für das DataProvider-Objekt in ein Array eingelesen wird. Die DataGrid-Komponente umfasst eine vertikale und horizontale Bildlaufleiste, unterstützt die Ereignisverarbeitung (einschließlich bearbeitbarer Zellen) und die Möglichkeit zur Sortierung der Daten.

Eigenschaften wie Schriftart, Farbe und Spaltenränder in einem Raster können angepasst und in der Größe geändert werden. Sie können einen selbst definierten Movieclip als Zellrenderer für eine Spalte des Rasters verwenden. (Ein Zellrenderer sorgt für die korrekte Anzeige des Zellinhalts.) Bildlaufleisten können angezeigt oder ausgeblendet werden und mithilfe der DataGrid-Methoden lässt sich eine Anzeige im Stil eines Seitenlayouts definieren. Weitere Informationen zur Anpassung finden Sie im Abschnitt zur DataGridColumn-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

### Verwandte Themen

[Erstellen, Füllen und Größenänderung von DataGrid-Komponenten](#)

[Anpassen und sortieren von DataGrid-Komponenten](#)

[Filtern und formatieren von Daten in DataGrid-Komponenten](#)

## Benutzerinteraktion mit der DataGrid-Komponente

Die Interaktion mit einer DataGrid-Komponente kann über die Maus oder die Tastatur erfolgen.

Wenn sowohl die Eigenschaft `sortableColumns` als auch die Spalteneigenschaft `sortable` den Wert `true` haben, lassen sich die Daten per Klick auf die Spaltenüberschrift nach den Werten dieser Spalte sortieren. Sie können die Sortierung einer bestimmten Spalte deaktivieren, indem Sie die Spalteneigenschaft `sortable` auf `false` setzen.

Wenn die Eigenschaft `resizableColumns` den Wert `true` aufweist, können Sie die Spaltenbreite ändern, indem Sie die Spaltentrennlinien in der Überschriftenzeile ziehen.

Das Klicken in einer bearbeitbaren Zelle verschiebt den Fokus auf diese Zelle. Das Klicken auf eine nicht bearbeitbare Zelle hat keine Auswirkungen auf den Fokus. Eine einzelne Zelle ist dann bearbeitbar, wenn die Eigenschaften `DataGrid.editable` und `DataGridColumn.editable` der Zelle `true` sind.

Weitere Informationen finden Sie in den Abschnitten zur DataGrid- und DataGridColumn-Klasse im [ActionScript 3.0 Referenzhandbuch](#).

Wenn eine DataGrid-Instanz durch Klicken oder durch Drücken der Tabulatortaste den Fokus erhält, können Sie sie mit folgenden Tasten steuern:

| Schlüssel                            | Beschreibung                                                                                                                                                                                                                                                                                                                 |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nach-unten-Taste                     | Wenn eine Zelle bearbeitet wird, verschiebt sich die Einfügemarke an das Textende der Zelle. Ist die Zelle nicht bearbeitbar, erfolgt die Auswahl mit der Nach-unten-Taste wie mit der List-Komponente.                                                                                                                      |
| Nach-oben-Taste                      | Wenn eine Zelle bearbeitet wird, wird die Einfügemarke an den Textanfang der Zelle verschoben. Ist die Zelle nicht bearbeitbar, erfolgt die Auswahl mit der Nach-oben-Taste wie mit der List-Komponente.                                                                                                                     |
| Umschalt+Nach-oben-/Nach-unten-Taste | Ist das Datenraster nicht bearbeitbar und <code>allowMultipleSelection</code> ist <code>true</code> , werden mehrere benachbarte Zeilen ausgewählt. Die umgekehrte Richtungstaste deaktiviert die ausgewählten Zeilen, bis die Anfangszeile überschritten wird, ab der die Zeilen in umgekehrter Richtung ausgewählt werden. |



| Schlüssel                               | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Umschalt+Klick                          | Wenn <code>allowMultipleSelection</code> den Wert <code>true</code> hat, werden alle Zeilen zwischen der ausgewählten Zeile und der Position der Einfügemarke (hervorgehobene Zelle) ausgewählt.                                                                                                                                                                                                                                                                                                                             |
| Strg+Klick                              | Wenn <code>allowMultipleSelection</code> den Wert <code>true</code> hat, werden weitere Zeilen ausgewählt, die nicht benachbart sein müssen.                                                                                                                                                                                                                                                                                                                                                                                 |
| Nach-rechts-Taste                       | Wenn eine Zelle bearbeitet wird, wird die Einfügemarke ein Zeichen nach rechts verschoben. Ist die Zelle nicht bearbeitbar, hat die Nach-rechts-Taste keine Wirkung.                                                                                                                                                                                                                                                                                                                                                         |
| Nach-links-Taste                        | Wenn eine Zelle bearbeitet wird, wird die Einfügemarke ein Zeichen nach links verschoben. Ist die Zelle nicht bearbeitbar, hat die Nach-links-Taste keine Wirkung.                                                                                                                                                                                                                                                                                                                                                           |
| Startseite                              | Wählt die erste Zeile im Datenraster aus.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Ende                                    | Wählt die letzte Zeile im Datenraster aus.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Bild-auf                                | Wählt die erste Zeile innerhalb einer Seite des Datenrasters aus. Eine Seite umfasst die Anzahl an Zeilen, die im Datenraster Platz haben ohne weiterzublättern.                                                                                                                                                                                                                                                                                                                                                             |
| Bild-ab                                 | Wählt die letzte Zeile innerhalb einer Seite des Datenrasters aus. Eine Seite umfasst die Anzahl an Zeilen, die im Datenraster Platz haben ohne weiterzublättern.                                                                                                                                                                                                                                                                                                                                                            |
| Eingabetaste/Umschalt+Eingabeta-<br>ste | Ist eine Zelle bearbeitbar, wird die Änderung übernommen und die Einfügemarke an die Zelle in derselben Spalte der nächsten Zeile verschoben (je nach Verwendung der Umschalttaste nach oben oder unten).                                                                                                                                                                                                                                                                                                                    |
| Umschalt+Tab/Tab                        | Ist das Datenraster bearbeitbar, wird der Fokus auf das vorige/nächste Element verschoben, bis das Ende der Spalte erreicht ist, und dann auf die vorige/nächste Zeile, bis die erste bzw. letzte Zelle erreicht ist. Ist die erste Zelle ausgewählt, verschiebt Umschalt+Tab den Fokus auf das vorige Steuerelement. Ist die letzte Zelle ausgewählt, verschiebt Tab den Fokus auf das nächste Steuerelement.<br><br>Ist das Datenraster nicht bearbeitbar, wird der Fokus auf das vorige/nächste Steuerelement verschoben. |

Die DataGrid-Komponente lässt sich als Grundlage für zahlreiche Typen datenbetriebener Anwendungen einsetzen. Sie können auf einfache Weise eine formatierte Tabellenansicht der Daten anzeigen oder mit den Möglichkeiten des Zellrenderers komplexere und bearbeitbare Teilbereiche von Benutzeroberflächen erstellen. Die DataGrid-Komponente kann beispielsweise für folgende Einsatzzwecke verwendet werden:

- Webmail-Client
- Suchergebnisseiten
- Tabellenkalkulationsanwendungen, z.B. Darlehensberechnungen und Steuerformularanwendungen

Für die Erstellung von Anwendungen mit der DataGrid-Komponente ist es hilfreich, das Design der List-Komponente zu kennen und zu verstehen, da die DataGrid-Klasse eine Erweiterung der SelectableList-Klasse ist. Weitere Informationen zur SelectableList-Klasse und zur List-Komponente finden Sie in den Abschnitten zur SelectableList- und List-Klasse im [ActionScript 3.0 Referenzhandbuch](#).

Die einer Anwendung hinzugefügten DataGrid-Komponenten können Sie Bildschirmleseprogrammen anhand der folgenden ActionScript-Codezeilen zugänglich machen:

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

Diese Anweisung ist unabhängig von der Anzahl der Instanzen nur einmal pro Komponente erforderlich. Weitere Informationen finden Sie im Handbuch *Verwenden von Flash* im Kapitel 18, „Barrierefreie Inhalte erstellen“.

## Parameter der DataGrid-Komponente

Die folgenden Authoring-Parameter können für jede Instanz einer DataGrid-Komponente im Eigenschaften- oder Komponenten-Inspektor festgelegt werden: `allowMultipleSelection`, `editable`, `headerHeight`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `resizableColumns`, `rowHeight`, `showHeaders`, `verticalLineScrollSize`, `verticalPageScrollSize` und `verticalScrollPolicy`. Jeder dieser Parameter besitzt eine entsprechende ActionScript-Eigenschaft desselben Namens. Weitere Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur DataGrid-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Erstellen einer Anwendung mit der DataGrid-Komponente

Um eine Anwendung mit der DataGrid-Komponenten zu erstellen, müssen Sie zuerst bestimmen, woher die Daten stammen. Typischerweise stammen die Daten aus einem Array, das Sie in das Raster einlesen können, indem Sie die Eigenschaft `dataProvider` setzen. Sie können die Daten mithilfe der Methoden der Klassen „DataGrid“ und „DataGridColumn“ in das Raster einfügen.

### Verwenden eines lokalen Datenproviders mit der DataGrid-Komponente

Im folgenden Beispiel wird ein Datenraster erstellt, um die Mitgliederliste eines Softballteams anzuzeigen. Die Mitgliederliste wird in einem Array (`aRoster`) angezeigt, das der Eigenschaft `dataProvider` der DataGrid-Komponente zugewiesen wird.

- 1 Wählen Sie in Flash „Datei“ > „Neu“ und dann „Flash-Datei“ (ActionScript 3.0).
- 2 Ziehen Sie die DataGrid-Komponente vom Bedienfeld „Komponenten“ auf die Bühne.
- 3 Geben Sie im Eigenschafteninspektor den Instanznamen **aDg** ein.
- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

**Verwenden der UI-Komponenten**

```

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);
};

```

Die Funktion `bldRosterGrid()` legt die Größe des Datenrasters fest sowie die Reihenfolge der Spalten und deren Größen.

**5 Wählen Sie „Steuerung“ > „Film testen“.****Definieren der Spalten und deren Sortierung für eine DataGrid-Komponente in einer Anwendung**

Beachten Sie, dass der Benutzer den Inhalt eines Datenrasters nach den Werten einer Spalte sortieren kann, indem er auf die Überschrift der betreffenden Spalte klickt.

Im folgenden Beispiel werden einer DataGrid-Komponente mit der `addColumn()`-Methode `DataGridColumn`-Instanzen hinzugefügt. Die einzelnen Spalten repräsentieren die Namen der Spieler und deren Punkte (Scores). In dem Beispiel wird zudem die Eigenschaft `sortOptions` so eingestellt, dass sie die Sortieroptionen für jede Spalte angibt: `Array.CASEINSENSITIVE` für die Spalte „Name“ und `Array.NUMERIC` für die Spalte „Score“. Die Größe des Datenrasters wird angepasst, indem die Länge (`length`) auf die Anzahl der Zeilen gesetzt wird und die Breite (`width`) auf 200.

- 1 Wählen Sie in Flash „Datei“ > „Neu“ und dann „Flash-Datei“ (ActionScript 3.0).
- 2 Ziehen Sie die DataGrid-Komponente vom Bedienfeld „Komponenten“ auf die Bühne.
- 3 Geben Sie im Eigenschafteninspektor den Instanznamen **aDg** ein.
- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

**Verwenden der UI-Komponenten**

```

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 Wählen Sie „Steuerung“ > „Film testen“.

**Erstellen einer DataGrid-Komponenteninstanz mithilfe von ActionScript-Code**

Im folgenden Beispiel wird mithilfe von ActionScript ein Datenraster erstellt und mit einem Array an Spielernamen und Punkten gefüllt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die DataGrid-Komponente vom Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ des aktuellen Dokuments.

Hierdurch wird die Komponente der Bibliothek hinzugefügt, sie wird jedoch nicht in der Anwendung angezeigt.

- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```

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);

```

Dieser Code erstellt die DataGrid-Instanz und legt dann Größe und Position des Rasters fest.

- 4 Erstellen Sie ein Array, fügen Sie dem Array Daten hinzu, und legen Sie das Array als Datenprovider für das Datenraster fest:

```

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 Wählen Sie „Steuerung“ > „Film testen“.

### Laden einer XML-Datei in ein Datenraster

Im folgenden Beispiel werden mithilfe der `DataGridColumn`-Klasse die Spalten des Datenrasters erstellt. Das Datenraster wird ausgefüllt, indem als `value`-Parameter des Konstruktors `DataProvider()` ein XML-Objekt übergeben wird.

- 1 Erstellen Sie mithilfe eines Texteditors eine XML-Datei mit den folgenden Daten und speichern Sie sie unter `team.xml` in demselben Ordner, in dem später die FLA-Datei gespeichert wird.

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

- 2 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 3 Doppelklicken Sie im Bedienfeld „Komponenten“ auf die `DataGrid`-Komponente, um sie der Bühne hinzuzufügen.
- 4 Geben Sie im Eigenschafteninspektor den Instanznamen `aDg` ein.
- 5 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“.

## Arbeiten mit der Label-Komponente

Die Label-Komponente (eine Bezeichnung) zeigt eine einzelne Textzeile an, normalerweise, um ein anderes Element oder eine Aktivität auf einer Webseite zu kennzeichnen. Sie können festlegen, dass eine Bezeichnung mit HTML formatiert wird, um die Textformatierungs-Tags von HTML nutzen zu können. Sie können auch die Ausrichtung und Größe der Bezeichnung steuern. Label-Komponenten haben keine Ränder, können keinen Fokus erhalten und übermitteln keine Ereignisse.

Die Änderungen an den Parametern der einzelnen Label-Instanzen, die beim Authoring im Eigenschaften- oder Komponenten-Inspektor vorgenommen werden, werden jeweils in einer Live-Vorschau angezeigt. Wegen des fehlenden Rahmens müssen Sie beim Authoring einen Textinhalt mit dem Parameter „text“ zuweisen, um eine Bezeichnung in der Live-Vorschau sichtbar zu machen.

### Benutzerinteraktion mit der Label-Komponente

Verwenden Sie eine Label-Komponente, um eine Beschriftung für eine andere Komponente in einem Formular, wie z. B. „Name:“, zu erstellen, und zwar links von einem Texteingabefeld, in das der Benutzername eingetragen werden kann. Die Verwendung einer Label-Komponente hat gegenüber einem einfachen Textfeld den Vorteil, dass Stile eingesetzt werden können und somit ein einheitliches Erscheinungsbild entsteht.

Wenn Sie eine Label-Komponente drehen wollen, müssen Sie die Schriften einbetten, da sie sonst beim Testen des Films nicht angezeigt werden.

### Parameter der Label-Komponente

Die folgenden Authoring-Parameter können für jede Instanz einer Label-Komponente im Eigenschaften- oder Komponenteninspektor festgelegt werden: `autoSize`, `condenseWhite`, `selectable`, `text` und `wordWrap`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur Label-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

### Erstellen einer Anwendung mit der Label-Komponente

Im Folgenden wird erläutert, wie Sie einer Anwendung beim Authoring eine Label-Komponente hinzufügen. In dem Beispiel wird in der Label-Komponente einfach der Text „Expiration Date“ angezeigt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine Label-Komponente vom Bedienfeld „Komponenten“ auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
  - Geben Sie als Instanznamen **aLabel** ein.
  - Geben Sie als Breite (B) den Wert **80** ein.
  - Geben Sie **100** als X-Wert ein.
  - Geben Sie **100** als Y-Wert ein.
  - Geben Sie **Expiration Date** für den Parameter `text` ein.
- 3 Ziehen Sie eine TextArea-Komponente auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
  - Geben Sie als Instanznamen **aTa** ein.
  - Geben Sie als Höhe (H) den Wert **22** ein.

**Verwenden der UI-Komponenten**

- Geben Sie **200** als X-Wert ein.
- Geben Sie **100** als Y-Wert ein.

- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“.

**Erstellen einer Label-Komponenteninstanz mithilfe von ActionScript-Code**

Im folgenden Beispiel wird mithilfe von ActionScript ein Label-Parameter erstellt. Die Beschriftung soll die Funktion einer ColorPicker-Komponente identifizieren und mithilfe der Eigenschaft `htmlText` erhält der Beschriftungstext die Möglichkeit zur Formatierung.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die Label-Komponente vom Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ des aktuellen Dokuments.
- 3 Ziehen Sie die ColorPicker-Komponente vom Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ des aktuellen Dokuments.
- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

**Verwenden der UI-Komponenten**

```

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

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

addChild(aLabel);
addChild(aCp);

aLabel.htmlText = 'Fill:';
aLabel.x = 200;
aLabel.y = 150;
aLabel.width = 25;
aLabel.height = 22;

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

```

5 Wählen Sie „Steuerung“ > „Film testen“.

## Arbeiten mit der List-Komponente

Die List-Komponente ist ein Listenfeld mit Bildlaufleiste zur Auswahl eines oder mehrerer Elemente. In Listen können auch Grafiken und andere Komponenten angezeigt werden. Die in der Liste angezeigten Werte werden im Dialogfeld „Werte“ hinzugefügt, das Sie durch Klicken auf das Parameterfeld „label“ oder „data“ aufrufen können. Sie können der Liste auch mit den Methoden `List.addItem()` und `List.addItemAt()` Elemente hinzufügen.

Die List-Komponente verwendet einen von Null ausgehenden Index, wobei das Element mit dem Indexwert 0 an oberster Stelle der Liste angezeigt wird. Wenn Sie Elemente mit den Methoden und Eigenschaften der List-Klasse hinzufügen, entfernen oder ersetzen, müssen Sie in bestimmten Fällen die Indizes der Elemente angeben.

### Benutzerinteraktion mit der List-Komponente

Sie können beim Erstellen der Liste festlegen, ob der Benutzer nur ein einzelnes Element oder auch mehrere Elemente auf einmal auswählen kann. Nehmen wir als Beispiel einen Online-Shop. Das Sortiment des Online-Shops umfasst 30 Artikel, die in einer Liste mit Bildlaufleiste durch Klicken ausgewählt werden können.

Die Zeilen einer Liste können auch benutzerdefinierte Movieclips enthalten, damit mehr Informationen angezeigt werden können. Ein Beispiel wäre eine E-Mail-Anwendung, in der die verschiedenen Postfächer als List-Komponenten umgesetzt sind, wobei jedes Listenelement einer Nachricht entspricht und grafische Symbole den Benutzer über Priorität und Status der einzelnen Nachrichten informieren.

Die Liste erhält den Fokus, wenn Sie darauf klicken oder sie mit der Tabulatortaste ansteuern. Sie kann dann mit den folgenden Tasten gesteuert werden:

Schlüssel	Beschreibung
Alphanumerische Tasten	Steuert das nächste Element an, dessen Beschriftung mit <code>Key.getAsci()</code> beginnt.
Strg	Toggle-Taste, mit der die Auswahl mehrerer nicht aufeinander folgender Elemente vorgenommen oder aufgehoben werden kann.
Nach-unten-Taste	Verschiebt die Auswahl um ein Element nach unten.
Startseite	Verschiebt die Auswahl an den Anfang der Liste.



Schlüssel	Beschreibung
Bild-ab	Verschiebt die Auswahl um eine Seite nach unten.
Bild-auf	Verschiebt die Auswahl um eine Seite nach oben.
Umschalttaste	Ermöglicht die Auswahl von mehreren aufeinander folgenden Elementen.
Nach-oben-Taste	Verschiebt die Auswahl um ein Element nach oben.

**Hinweis:** Beachten Sie, dass die Bildlaufgröße in Pixel und nicht in Zeilen angegeben wird.

**Hinweis:** Die Seitengröße beim Blättern in der Dropdownliste mit der Bild-auf-Taste und Bild-ab-Taste entspricht der Anzahl der Elemente, die in den angezeigten Ausschnitt passen, minus eins. Beispiel: Bei einer zehnzeiligen Dropdownliste werden beim Blättern mit der Bild-ab-Taste die Elemente 0 bis 9, dann die Elemente 9 bis 18, dann die Elemente 18 bis 27 angezeigt usw. Die jeweils letzte angezeigte Zeile wird also immer zur ersten angezeigten Zeile der nächsten Seite.

Weitere Informationen zum Steuern des Fokus finden Sie in den Abschnitten zur IFocusManager-Schnittstelle und zur FocusManager-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#), und unter „Arbeiten mit dem FocusManager“ auf Seite 29.

Die Änderungen an den Parametern der einzelnen List-Instanzen, die beim Authoring im Eigenschaften- oder Komponenten-Inspektor vorgenommen werden, werden jeweils in einer Live-Vorschau angezeigt.

Die einer Anwendung hinzugefügten List-Komponenten können Sie Bildschirmleseprogrammen anhand der folgenden ActionScript-Codezeilen zugänglich machen:

```
import fl.accessibility.ListAccImpl;

ListAccImpl.enableAccessibility();
```

Diese Anweisung ist unabhängig von der Anzahl der Instanzen nur einmal pro Komponente erforderlich. Weitere Informationen finden Sie im Handbuch *Verwenden von Flash* im Kapitel 18, „Barrierefreie Inhalte erstellen“.

## Parameter der List-Komponente

Die folgenden Parameter können für jede Instanz einer List-Komponente im Eigenschaften- oder Komponenten-Inspektor festgelegt werden: `allowMultipleSelection`, `dataProvider`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `multipleSelection`, `verticalLineScrollSize`, `verticalPageScrollSize` und `verticalScrollPolicy`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur List-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#). Informationen über die Verwendung des Parameters „dataProvider“ finden Sie im Abschnitt „Verwenden des Parameters „dataProvider““ auf Seite 31.

## Erstellen einer Anwendung mit der List-Komponente

Im Folgenden wird erläutert, wie Sie einer Anwendung beim Authoring eine List-Komponente hinzufügen.

### Einfügen einer einfachen List-Komponente in eine Anwendung

In diesem Beispiel besteht die Liste aus Beschriftungen, die Automodelle identifizieren, und Datenfeldern, die Preise enthalten.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine List-Komponente vom Bedienfeld „Komponenten“ auf die Bühne.

- 3 Führen Sie im Eigenschafteninspektor die folgenden Schritte durch:
  - Geben Sie den Instanznamen **aList** ein.
  - Geben Sie als Breite (B) den Wert **200** an.
- 4 Verwenden Sie das Textwerkzeug, um unter der Instanz `aList` ein Textfeld zu erstellen und weisen Sie ihm den Instanznamen **aTf** zu.
- 5 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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;
}
```

In diesem Code wird die Methode `addItem()` verwendet, um `aList` mit drei Elementen zu füllen, wobei jedem Element ein `label`-Wert zugewiesen wird, der in der Liste erscheint, sowie ein `data`-Wert. Wenn der Benutzer ein Element in der Liste auswählt, ruft der Ereignis-Listener die Funktion `showData()` auf, die den `data`-Wert für das ausgewählte Element anzeigt.

- 6 Wählen Sie „Steuerung“ > „Film testen“, um die Anwendung zu kompilieren und auszuführen.

### Füllen einer List-Instanz mit einem Datenprovider

In diesem Beispiel wird eine Liste mit Automodellen und dem jeweiligen Preis erstellt. Für das Ausfüllen der Liste wird jedoch anstelle der Methode `addItem()` ein Datenprovider verwendet.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine List-Komponente vom Bedienfeld „Komponenten“ auf die Bühne.
- 3 Führen Sie im Eigenschafteninspektor die folgenden Schritte durch:
  - Geben Sie den Instanznamen **aList** ein.
  - Geben Sie als Breite (B) den Wert **200** an.
- 4 Verwenden Sie das Textwerkzeug, um unter der Instanz `aList` ein Textfeld zu erstellen und weisen Sie ihm den Instanznamen **aTf** zu.
- 5 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

**Verwenden der UI-Komponenten**

```
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 Wählen Sie „Steuerung“ > „Film testen“, um die Liste mit den Elementen anzuzeigen.

**Verwenden einer List-Komponente zum Steuern einer MovieClip-Instanz**

Im folgenden Beispiel wird eine Liste mit Farbnamen erstellt. Wenn der Benutzer eine Farbe auswählt, wird diese einem Movieclip zugewiesen.

- 1 Erstellen Sie ein Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine List-Komponente vom Bedienfeld „Komponenten“ auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
  - Geben Sie als Instanznamen **aList** ein.
  - Geben Sie als Höhe (H) den Wert **60** ein.
  - Geben Sie **100** als X-Wert ein.
  - Geben Sie **150** als Y-Wert ein.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

**Verwenden der UI-Komponenten**

```

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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.
- 5 Klicken Sie auf verschiedene Farben in der Liste, um zu sehen, wie sie im Movieclip angezeigt werden.

**Erstellen einer List-Komponenteninstanz mithilfe von ActionScript-Code:**

In diesem Beispiel wird eine einfache Liste mit ActionScript erstellt und mit der Methode `addItem()` gefüllt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die List-Komponente vom Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```

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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.

## Arbeiten mit der NumericStepper-Komponente

Die NumericStepper-Komponente (Zähler) ermöglicht es dem Benutzer, eine geordnete Zahlenmenge schrittweise durchzugehen. Die Komponente besteht aus einer Zahl in einem Textfeld, das neben kleinen aufwärts- und abwärts gerichteten Pfeilschaltflächen angezeigt wird. Wenn der Benutzer auf die Schaltflächen klickt, wird die Zahl schrittweise in der mit dem Parameter `stepSize` festgelegten Einheit erhöht oder verringert, bis der Benutzer die Schaltflächen loslässt oder der höchste bzw. niedrigste Wert erreicht wurde. Der Text im Textfeld der NumericStepper-Komponente kann ebenfalls bearbeitet werden.

Eine Live-Vorschau der einzelnen NumericStepper-Instanzen spiegelt die Einstellung des Parameters „value“ im Eigenschaften- oder Komponenten-Inspektor wieder. In der Live-Vorschau können die Pfeilschaltflächen des Zählers jedoch nicht mit der Maus oder der Tastatur bedient werden.

### Benutzerinteraktion mit der NumericStepper-Komponente

Sie können den Zähler überall verwenden, wo der Benutzer einen Zahlenwert einstellen soll. Sie können eine NumericStepper-Komponente etwa in einem Formular verwenden, in dem der Monat, der Tag und das Jahr des Gültigkeitsdatums einer Kreditkarte eingestellt werden kann. Sie können eine NumericStepper-Komponente auch verwenden, damit der Benutzer eine Schriftgröße vergrößern oder verkleinern kann.

Die NumericStepper-Komponente verarbeitet nur numerische Daten. Außerdem müssen Sie beim Authoring die Größe des Zählers anpassen, wenn Sie mehr als zwei Ziffern anzeigen möchten (z. B. die Zahlen 5246 oder 1,34).

Einen Zähler in einer Anwendung können Sie aktivieren oder deaktivieren. Im deaktivierten Zustand reagiert der Zähler nicht auf Maus- und Tastatureingaben. Ein aktivierter Zähler erhält den Fokus, wenn Sie darauf klicken oder ihn mit der Tabulatortaste ansteuern; der interne Fokus wird auf das Textfeld gelegt. Wenn eine NumericStepper-Instanz den Fokus hat, kann sie mit den folgenden Tasten gesteuert werden:

Schlüssel	Beschreibung
Nach-unten-Taste	Der Wert ändert sich um eine Einheit.
Nach-links-Taste	Verschiebt die Einfügemarke im Textfeld nach links.
Nach-rechts-Taste	Verschiebt die Einfügemarke im Textfeld nach rechts.
Umschalt+Tab	Verschiebt den Fokus auf das vorherige Objekt.
Tab	Verschiebt den Fokus auf das nächste Objekt.
Nach-oben-Taste	Der Wert ändert sich um eine Einheit.

Weitere Informationen zum Steuern des Fokus finden Sie im Abschnitt zur FocusManager-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#) und unter „Arbeiten mit dem FocusManager“ auf Seite 29.

### Parameter der Komponente NumericStepper

Die folgenden Parameter können für jede Instanz einer NumericStepper-Komponente im Eigenschaften- oder Komponenten-Inspektor festgelegt werden: `maximum`, `minimum`, `stepSize` und `value`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur NumericStepper-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Erstellen einer Anwendung mit der NumericStepper-Komponente

Im folgenden Verfahren wird erklärt, wie Sie beim Authoring die NumericStepper-Komponente in eine Anwendung einfügen. Im Beispiel werden eine NumericStepper-Komponente und eine Label-Komponente auf der Bühne platziert, und es wird ein Listener für ein `Event.CHANGE`-Ereignis für die NumericStepper-Instanz erstellt. Wenn sich der Wert in der NumericStepper-Instanz ändert, wird der neue Wert in der Eigenschaft `text` der Label-Instanz angezeigt.

- 1 Ziehen Sie die NumericStepper-Komponente vom Bedienfeld „Komponenten“ auf die Bühne.
- 2 Geben Sie im Eigenschaftensinspektor den Instanznamen `aNs` ein.
- 3 Ziehen Sie eine Label-Komponente vom Bedienfeld „Komponenten“ auf die Bühne.
- 4 Geben Sie im Eigenschaftensinspektor den Instanznamen `aLabel` ein.
- 5 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
import flash.events.Event;

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

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

In diesem Beispiel wird der Beschriftungstext (die Eigenschaft `text` der label-Instanz) auf den Wert des Zählers (der Eigenschaft „value“ der NumericStepper-Instanz) gesetzt. Durch die Funktion `changeHandler()` wird die Eigenschaft `text` der Beschriftung immer dann geändert, wenn der Wert im Zähler geändert wird.

- 6 Wählen Sie „Steuerung“ > „Film testen“.

### Erstellen eines Zählers mithilfe von ActionScript-Code

In diesem Beispiel werden mithilfe von ActionScript-Code drei Zähler erstellt, und zwar je einer für die Eingabe von Monat, Tag und Jahr des Geburtstags des Benutzers. Zudem werden Beschriftungen für Aufforderung und Bezeichner eines jeden Zählers hinzugefügt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine Label-Komponente in das Bedienfeld „Bibliothek“.
- 3 Ziehen Sie eine NumericStepper-Komponente in das Bedienfeld „Bibliothek“.
- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.

## Arbeiten mit der ProgressBar-Komponente

Die ProgressBar-Komponente (Fortschrittsleiste oder Verlaufsbalken) zeigt den Fortschritt beim Laden von Inhalt an. Bei umfangreichen Inhalten, die die Ausführung einer Anwendung verzögern, ist dies wichtig für den Benutzer, da er dadurch die Bestätigung bekommt, dass tatsächlich etwas passiert. Dies ist z.B. hilfreich, um den Fortschritt beim Laden von Bildern und Teilen einer Anwendung anzuzeigen. Der Fortschritt beim Laden kann bestimmt oder unbestimmt sein. Eine *bestimmte* Fortschrittsleiste ist eine lineare Darstellung des Fortschritts einer Aufgabe im Zeitverlauf; sie wird verwendet, wenn bekannt ist, wie groß der geladene Inhalt ist. Eine *unbestimmte* Fortschrittsleiste wird verwendet, wenn die Größe des geladenen Inhalts nicht bekannt ist. Sie können auch eine Label-Komponente hinzufügen, die den Ladefortschritt als Prozentsatz anzeigt.

Die ProgressBar-Komponente verwendet die Skalierung im 9-teiligen Segmentraster (Scale-9) und verfügt über Skins für die Leiste und für den Balken sowie über eine unbestimmte Skin.

### Benutzerinteraktion mit der ProgressBar-Komponente

Es gibt drei Modi für den Einsatz der ProgressBar-Komponente. Am häufigsten werden der Ereignismodus („event“) und der Abfragemodus („polled“) verwendet. Diese Modi bestimmen die Art des Ladevorgangs: Es werden entweder `progress-` und `complete-`Ereignisse ausgegeben (im Ereignis- und Abfragemodus) oder es werden die `bytesLoaded-` und `bytesTotal-`Eigenschaften angezeigt (im Abfragemodus). Außerdem können Sie die ProgressBar-Komponente im manuellen Modus einsetzen, indem Sie die Eigenschaften `maximum`, `minimum` und `value` zusammen mit Aufrufen an die Methode `ProgressBar.setProgress()` einstellen. Mithilfe der Eigenschaft „indeterminate“ können Sie festlegen, ob die Fortschrittsleiste eine gestreifte Füllung hat und eine Quelle unbekannter Größe (`true`) oder eine durchgängige Füllung und eine Quelle bekannter Größe (`false`).

Sie legen den Modus der Fortschrittsleiste fest, indem Sie ihre `mode`-Eigenschaft angeben, und zwar entweder mithilfe des `mode`-Parameters im Eigenschafteninspektor oder im Komponenten-Inspektor oder mithilfe von ActionScript.

Wenn die Fortschrittsleiste einen Verarbeitungsstatus anzeigen soll, etwa bei der Analyse von 100.000 Elementen, ist Folgendes zu beachten: Bei einer einzelnen Frame-Schleife gibt es keine sichtbaren Aktualisierungen in der Fortschrittsleiste, da der Bildschirm dabei nicht neu aufgebaut wird.

### Parameter der ProgressBar-Komponente

Die folgenden Parameter können für jede Instanz einer ProgressBar-Komponente im Eigenschaften- oder Komponenten-Inspektor festgelegt werden: `direction`, `mode` und `source`. Jeder dieser Parameter besitzt eine entsprechende ActionScript-Eigenschaft desselben Namens.

Mithilfe von ActionScript-Anweisungen können Sie diese und weitere Optionen für die ProgressBar-Komponente über deren Eigenschaften, Methoden und Ereignisse steuern. Weitere Informationen finden Sie im Abschnitt zur ProgressBar-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).



## Erstellen einer Anwendung mit der ProgressBar-Komponente

Im Folgenden wird gezeigt, wie Sie beim Authoring eine ProgressBar-Komponente in eine Anwendung einfügen. In diesem Beispiel wird die Fortschrittsleiste im Ereignismodus eingesetzt. Im Ereignismodus werden vom ladenden Inhalt die Ereignisse `progress` und `complete` ausgegeben. Diese verarbeitet die Fortschrittsleiste für die Anzeige des Fortschritts. Wenn das Ereignis `progress` stattfindet, wird eine Beschriftung so aktualisiert, dass sie den Prozentsatz des bereits geladenen Inhalts anzeigt. Wenn das Ereignis `complete` stattfindet, wird in diesem Beispiel „Loading complete“ angezeigt sowie der Wert der Eigenschaft `bytesTotal`, nämlich die Größe der Datei.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die ProgressBar-Komponente vom Bedienfeld „Komponenten“ auf die Bühne.
  - Geben Sie im Eigenschafteninspektor den Instanznamen **aPb** ein.
  - Geben Sie auf der Registerkarte „Parameter“ **200** als X-Wert ein.
  - Geben Sie **260** als Y-Wert ein.
  - Wählen Sie `event` für den Parameter `mode`.
- 3 Ziehen Sie die Button-Komponente vom Bedienfeld „Komponenten“ auf die Bühne.
  - Geben Sie im Eigenschafteninspektor den Instanznamen **loadButton** ein.
  - Geben Sie **220** als X-Wert ein.
  - Geben Sie **290** als Y-Wert ein.
  - Geben Sie **Show** für den Parameter „label“ ein.
- 4 Ziehen Sie die Label-Komponente auf die Bühne und geben Sie ihr den Instanznamen **progLabel**.
  - Geben Sie als Breite (B) den Wert **150** ein.
  - Geben Sie **200** als X-Wert ein.
  - Geben Sie **230** als Y-Wert ein.
  - Löschen Sie auf der Registerkarte „Parameter“ den Wert für den Parameter `text`.
- 5 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein, um eine .mp3-Audiodatei zu laden:

```
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 Wählen Sie „Steuerung“ > „Film testen“.

### Erstellen einer Anwendung mit der ProgressBar-Komponente im Abfragemodus

Im folgenden Beispiel befindet sich die ProgressBar-Komponente im Abfragemodus. Im Abfragemodus wird der Fortschritt folgendermaßen bestimmt: Es wird auf `progress`-Ereignisse zum ladenden Inhalt gewartet und mithilfe der Eigenschaften `bytesLoaded` und `bytesTotal` des ladenden Inhalts der Fortschritt berechnet. Dieses Beispiel lädt ein Sound-Objekt, wartet auf seine `progress`-Ereignisse und berechnet den geladenen Prozentsatz unter Verwendung der Eigenschaften `bytesLoaded` und `bytesTotal`. Der bereits geladene Inhalt wird als Prozentwert in einer Bezeichnung und im Bedienfeld „Ausgabe“ angezeigt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine ProgressBar-Komponente vom Bedienfeld „Komponenten“ auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
  - Geben Sie als Instanznamen **aPb** ein.
  - Geben Sie **185** als X-Wert ein.
  - Geben Sie **225** als Y-Wert ein.
- 3 Ziehen Sie eine Label-Komponente auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
  - Geben Sie als Instanznamen **progLabel** ein.
  - Geben Sie **180** als X-Wert ein.

- Geben Sie **180** als Y-Wert ein.
- Löschen Sie auf der Registerkarte „Parameter“ den Wert für den Parameter „text“.

- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie Bild 1 in der Zeitleiste aus und geben Sie den folgenden ActionScript-Code ein, mit dem ein Soundobjekt (`aSound`) erstellt und die Methode `loadSound()` aufgerufen wird, um einen Sound in das Soundobjekt zu laden:

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

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

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

aSound.load(request);

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

- 5 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.

### Erstellen einer Anwendung mit der ProgressBar-Komponente im manuellen Modus

Im folgenden Beispiel befindet sich die ProgressBar-Komponente im manuellen Modus. Im manuellen Modus müssen Sie den Fortschritt manuell einstellen, indem Sie die Methode `setProgress()` aufrufen und ihr den aktuellen und den maximalen Wert übergeben. Aufgrund dieser Werte berechnet die Methode, wie weit der Vorgang fortgeschritten ist. Die Eigenschaft `source` stellen Sie im manuellen Modus nicht ein. Im Beispiel wird eine NumericStepper-Komponente mit einem Maximalwert von 250 verwendet, um die Fortschrittsleiste hochzuzählen. Wenn sich der Wert in der NumericStepper-Komponente ändert und ein `CHANGE`-Ereignis auslöst, ruft die Ereignisprozedur (`nsChangeHandler`) die Methode `setProgress()` auf, um die Fortschrittsleiste nach vorne zu bewegen. Zudem wird angezeigt, zu wie viel Prozent der Vorgang in Relation zum Maximalwert abgeschlossen ist.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine ProgressBar-Komponente vom Bedienfeld „Komponenten“ auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
  - Geben Sie als Instanznamen **aPb** ein.
  - Geben Sie **180** als X-Wert ein.
  - Geben Sie **175** als Y-Wert ein.
- 3 Ziehen Sie eine NumericStepper-Komponente auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
  - Geben Sie als Instanznamen **aNs** ein.
  - Geben Sie **220** als X-Wert ein.
  - Geben Sie **215** als Y-Wert ein.

**Verwenden der UI-Komponenten**

- Geben Sie unter „Parameter“ **250** als Maximalwert (maximum), **0** als Minimalwert (minimum), **1** als Schritintervall (stepSize) und **0** für den Parameter value an.
- 4 Ziehen Sie eine Label-Komponente auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
    - Geben Sie als Instanznamen **progLabel** ein.
    - Geben Sie als Breite (B) den Wert **150** ein.
    - Geben Sie **180** als X-Wert ein.
    - Geben Sie **120** als Y-Wert ein.
    - Löschen Sie auf der Registerkarte „Parameter“ den Wert „Label“ für den Parameter „text“.
  - 5 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.
- 7 Klicken Sie auf den nach oben gerichteten Pfeil des Zählers, um die Fortschrittsleiste vorrücken zu lassen.

**Erstellen einer Fortschrittsleiste mithilfe von ActionScript-Code**

In diesem Beispiel wird eine Fortschrittsleiste mithilfe von ActionScript-Code erstellt. Außerdem wird die Funktion des vorigen Beispiels wiederholt, in dem eine Fortschrittsleiste im manuellen Modus erstellt wurde.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine ProgressBar-Komponente in das Bedienfeld „Bibliothek“.
- 3 Ziehen Sie eine NumericStepper-Komponente in das Bedienfeld „Bibliothek“.
- 4 Ziehen Sie eine Label-Komponente in das Bedienfeld „Bibliothek“.
- 5 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.
- 7 Klicken Sie auf den nach oben gerichteten Pfeil des Zählers, um die Fortschrittsleiste vorrücken zu lassen.

## Arbeiten mit der RadioButton-Komponente

Mit der RadioButton-Komponente (Optionsfeld) können Sie erzwingen, dass Benutzer aus einer Reihe von Optionen nur eine einzige auswählen können. Diese Komponente muss in einer Gruppe von mindestens zwei RadioButton-Instanzen verwendet werden. Es kann immer nur ein Mitglied der Gruppe ausgewählt sein. Wenn der Benutzer ein Optionsfeld in einer Gruppe auswählt, wird die Auswahl des derzeit ausgewählten Optionsfelds aufgehoben. Mit dem Parameter `groupName` geben Sie an, zu welcher Gruppe ein Optionsfeld gehört.

Optionsfelder sind grundlegende Bestandteile vieler Formularanwendungen im Web. Sie können Optionsfelder immer dann verwenden, wenn ein Benutzer aus einer Gruppe von Optionen eine einzige auswählen soll. So würden Sie Optionsfelder beispielsweise einsetzen, um auf einem Formular zu fragen, mit welcher Kreditkarte ein Kunde zahlen möchte.

## Benutzerinteraktion mit der RadioButton-Komponente

Ein Optionsfeld kann aktiviert oder deaktiviert sein. Ein deaktiviertes Optionsfeld kann keine Maus- oder Tastatureingaben entgegennehmen. Wenn der Benutzer per Mausklick oder mit der Tabulatortaste zu einer Gruppe von RadioButton-Komponenten navigiert, wird nur dem ausgewählten Optionsfeld ein Fokus zugewiesen. Der Benutzer kann die Auswahl dann mit den folgenden Tasten steuern:

Schlüssel	Beschreibung
Pfeil nach oben/Pfeil nach links	Die Auswahl wird auf das vorherige Optionsfeld innerhalb der Optionsfeldgruppe verschoben.
Pfeil nach unten/Pfeil nach rechts	Die Auswahl wird auf das nächste Optionsfeld innerhalb der Optionsfeldgruppe verschoben.
Tab	Der Fokus wird von der Optionsfeldgruppe auf die nächste Komponente verschoben.

Weitere Informationen zum Steuern des Fokus finden Sie in den Abschnitten zur IFocusManager-Schnittstelle und zur FocusManager-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#) und unter „[Arbeiten mit dem FocusManager](#)“ auf Seite 29.

Eine Live-Vorschau der einzelnen RadioButton-Instanzen auf der Bühne spiegelt die Änderungen wieder, die im Eigenschaften- oder Komponenten-Inspektor beim Authoring vorgenommen wurden. Der Umstand, dass immer nur ein Optionsfeld ausgewählt sein kann, ist in der Live-Vorschau jedoch nicht sichtbar. Wenn Sie den Parameter `selected` für zwei Optionsfelder in der gleichen Gruppe auf „true“ setzen, werden beide als ausgewählt angezeigt, obwohl zur Laufzeit nur die zuletzt erstellte Instanz als ausgewählt angezeigt wird. Weitere Informationen finden Sie im Abschnitt „[Parameter der RadioButton-Komponente](#)“ auf Seite 82.

Die einer Anwendung hinzugefügten RadioButton-Komponenten können Sie einem Bildschirmleseprogramm anhand der folgenden Codezeilen zugänglich machen:

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

Diese Anweisung ist unabhängig von der Anzahl der Instanzen nur einmal pro Komponente erforderlich. Weitere Informationen finden Sie im Handbuch „Verwenden von Flash“ im Kapitel 18, „Barrierefreie Inhalte erstellen“.

## Parameter der RadioButton-Komponente

Die folgenden Authoring-Parameter können für jede Instanz einer RadioButton-Komponente im Eigenschaften- oder Komponenteninspektor festgelegt werden: `groupName`, `label`, `labelPlacement`, `selected` und `value`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur RadioButton-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

Sie können ActionScript-Code erstellen, um zusätzliche Optionen für RadioButton-Instanzen mit Methoden, Eigenschaften und Ereignissen der RadioButton-Klasse festzulegen.

## Erstellen einer Anwendung mit der RadioButton-Komponente

Im folgenden Verfahren wird beschrieben, wie Sie beim Authoring RadioButton-Komponenten in eine Anwendung einfügen. In diesem Beispiel werden die Optionsfelder eingesetzt, um eine Ja-oder-Nein-Frage zu präsentieren. Die Daten der Optionsfelder werden jeweils in einem Textbereich angezeigt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie zwei RadioButton-Komponenten vom Bedienfeld „Komponenten“ auf die Bühne.

- 3 Wählen Sie das erste Optionsfeld aus. Geben Sie im Eigenschafteninspektor den Instanznamen **yesRb** und den Gruppennamen **rbGroup** ein.
- 4 Wählen Sie das zweite Optionsfeld aus. Geben Sie im Eigenschafteninspektor den Instanznamen **noRb** und den Gruppennamen **rbGroup** ein.
- 5 Ziehen Sie eine TextArea-Komponente vom Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz **aTa**.
- 6 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.

### Erstellen eines Optionsfelds mithilfe von ActionScript-Code

In diesem Beispiel werden mithilfe von ActionScript-Code drei Optionsfelder für die Farben Rot, Blau und Grün erstellt, und es wird ein graues Feld gezeichnet. In der `value`-Eigenschaft der einzelnen Optionsfelder wird die mit der Option verbundene Farbe als Hexadezimalwert angegeben. Wenn der Benutzer auf eines der Optionsfelder klickt, ruft die Funktion `clickHandler()` die Methode `drawBox()` auf und übergibt die Farbe aus der `value`-Eigenschaft des Optionsfelds an das Feld.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine RadioButton-Komponente in das Bedienfeld „Bibliothek“.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.

Weitere Informationen finden Sie im Abschnitt zur RadioButton-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Arbeiten mit der ScrollPane-Komponente

Mit der ScrollPane-Komponente (Bildlauffenster) können Sie Inhalt anzeigen, der zu groß ist für den Bereich, in den er geladen wird. Wenn beispielsweise in einer Anwendung nur ein kleiner Bereich für ein großes Bild zur Verfügung steht, können Sie es in ein Bildlauffenster laden. Bildlauffenster können Movieclips sowie JPEG-, PNG-, GIF- und SWF-Dateien akzeptieren.



Komponenten wie „ScrollPane“ und „UILoader“ besitzen `complete`-Ereignisse, durch die Sie bestimmen können, wann der Inhalt vollständig geladen ist. Wenn Sie Eigenschaften des Inhalts einer ScrollPane- oder UILoader-Komponente festlegen wollen, warten Sie auf das `complete`-Ereignis und setzen dann die Eigenschaft in der Ereignisprozedur. Im folgenden Codebeispiel wird ein Listener für das `Event.COMPLETE`-Ereignis sowie eine Ereignisprozedur, welche die `alpha`-Eigenschaft des ScrollPane-Inhalts auf `.5` festlegt, erstellt:

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

Wenn Sie beim Laden von Inhalt in das Bildlauffenster eine Position angeben, müssen Sie die X- und Y-Koordination als 0,0 angeben. Im folgenden Beispielcode wird der Inhalt korrekt in die ScrollPane-Komponente geladen, da das Feld bei Position 0,0 gezeichnet wird.

```
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
```

Weitere Informationen finden Sie im Abschnitt zur ScrollPane-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Benutzerinteraktion mit der ScrollPane-Komponente

Ein Bildlauffenster kann aktiviert oder deaktiviert sein. Ein deaktiviertes Bildlauffenster kann keine Maus- oder Tastatureingaben entgegennehmen. Ein Benutzer kann ein Bildlauffenster mit den folgenden Tasten steuern, wenn es den Fokus hat:

Schlüssel	Beschreibung
Nach-unten-Taste	Der Inhalt wird vertikal um eine Zeile nach oben verschoben.
Nach-oben-Taste	Der Inhalt wird vertikal um eine Zeile nach unten verschoben.
Ende	Der Inhalt wird an den unteren Rand des Bildlauffensters verschoben.
Nach-links-Taste	Der Inhalt wird horizontal um eine Zeile nach rechts verschoben.
Nach-rechts-Taste	Der Inhalt wird horizontal um eine Zeile nach links verschoben.
Startseite	Der Inhalt wird an den Anfang des des Bildlauffensters verschoben.
Ende	Der Inhalt wird an den unteren Rand des Bildlauffensters verschoben.
Bild-ab	Der Inhalt wird vertikal um eine Bildlaufseite nach oben verschoben.
Bild-auf	Der Inhalt wird vertikal um eine Bildlaufseite nach unten verschoben.

Ein Benutzer kann die Maus verwenden, um mit der ScrollPane-Komponente zu interagieren, indem er auf den Inhalt oder auf die vertikalen und horizontalen Bildlaufleisten klickt. Der Benutzer kann Inhalte mit Maus ziehen, wenn die Eigenschaft `scrollDrag` auf `true` gesetzt ist. Das Einblenden eines handförmigen Zeigers über dem Inhalt zeigt an, dass der Benutzer den Inhalt ziehen kann. Im Gegensatz zu anderen Steuerelementen, treten Aktionen dann auf, wenn die Maustaste gedrückt ist, und werden fortgesetzt, wenn sie wieder losgelassen wird. Wenn der Inhalt gültige Tabstopps enthält, müssen Sie `scrollDrag` auf „false“ setzen. Andernfalls lösen alle Mausklicks auf den Inhalt einen Bildlauf aus.

## Parameter der ScrollPane-Komponente

Die folgenden Parameter können für jede ScrollPane-Instanz im Eigenschaft- oder Komponenten-Inspektor festgelegt werden: `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `scrollDrag`, `source`, `verticalLineScrollSize`, `verticalPageScrollSize` und `verticalScrollPolicy`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur ScrollPane-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

Mithilfe von ActionScript-Anweisungen können Sie diese und weitere Optionen für eine ScrollPane-Komponente mit den Eigenschaften, Methoden und Ereignissen steuern.

## Erstellen einer Anwendung mit der ScrollPane-Komponente

In der folgenden Anleitung wird erläutert, wie eine ScrollPane-Komponente einer Anwendung beim Authoring hinzugefügt wird. In diesem Beispiel lädt ein Bildlauffenster ein Bild aus einem mit der Eigenschaft `source` festgelegten Verzeichnis.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die ScrollPane-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz **aSp**.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.

### Erstellen einer ScrollPane-Komponenteninstanz mithilfe von ActionScript-Code

Im Beispiel wird ein Bildlauffenster erstellt, die Größe dieses Fensters wird festgelegt, und es wird mithilfe der Eigenschaft `source` ein Bild in das Bildlauffenster geladen. Es werden auch zwei Listener erstellt. Der erste wartet auf ein `scroll`-Ereignis und zeigt die Position des Bilds an, wenn der Benutzer einen vertikalen oder horizontalen Bildlauf ausführt. Der zweite wartet auf ein `complete`-Ereignis und zeigt im Bedienfeld „Ausgabe“ eine Meldung zum abgeschlossenen Ladevorgang an.

In diesem Beispiel wird mithilfe von ActionScript-Code ein Bildlauffenster erstellt. In diesem Bildlauffenster wird ein Movieclip (ein rotes Feld) platziert, das 150 Pixel breit und 300 Pixel hoch ist.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).

- 2 Ziehen Sie eine ScrollPane-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 3 Ziehen Sie die DataGrid-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.

## Arbeiten mit der Slider-Komponente

Die Slider-Komponente (Schieberegler) ermöglicht die Auswahl eines Werts durch Verschieben der grafischen Repräsentation eines Reglers zwischen den beiden Enden eines Wertebereichs. Mithilfe eines Schiebereglers kann der Benutzer einen Wert, wie etwa eine Zahl oder einen Prozentwert, auswählen. Es ist auch möglich, mithilfe von ActionScript-Code den Wert des Reglers zu setzen und dadurch das Verhalten eines zweiten Objekts zu beeinflussen. So könnten Sie beispielsweise einen Schieberegler mit einem Bild verbinden und aufgrund der relativen Position bzw. des Werts des Reglers dieses Bild vergrößern oder verkleinern.

Der aktuelle Wert des Schiebereglers wird von der relativen Position des Reglers zwischen den Endpunkten der Leiste bestimmt, die den Schiebereglerwerten „minimum“ und „maximum“ entsprechen.

Der Schieberegler erlaubt eine kontinuierliche Bandbreite an Werten zwischen dem niedrigsten und dem höchsten Wert, doch ist es auch möglich, durch den Parameter `snapInterval` ein Einrastintervall einzustellen, in dem der Regler zwischen dem Minimum und dem Maximum weitergerückt wird. Unabhängig von den zugewiesenen Werten des Reglers können an einem Schieberegler die Marken einer Skala angezeigt werden.

Der Schieberegler ist standardmäßig horizontal ausgerichtet; wenn Sie den Parameter `direction` jedoch auf „vertical“ stellen, wird der Regler senkrecht angezeigt. Die Leiste eines Schiebereglers reicht vom linken Ende zum rechten, wobei die Intervallmarken über der Leiste angeordnet sind.

## Benutzerinteraktion mit der Slider-Komponente

Wenn eine Slider-Instanz den Fokus hat, kann sie mit den folgenden Tasten gesteuert werden:

**Verwenden der UI-Komponenten**

Schlüssel	Beschreibung
Nach-rechts-Taste	Erhöht den verknüpften Wert (bei einem horizontalen Schieberegler).
Nach-oben-Taste	Erhöht den verknüpften Wert (bei einem vertikalen Schieberegler).
Nach-links-Taste	Reduziert den verknüpften Wert (bei einem horizontalen Schieberegler).
Nach-unten-Taste	Reduziert den verknüpften Wert (bei einem vertikalen Schieberegler).
Umschalt+Tab	Verschiebt den Fokus auf das vorherige Objekt.
Tab	Verschiebt den Fokus auf das nächste Objekt.

Weitere Informationen zum Steuern des Fokus finden Sie in den Abschnitten zur IFocusManager-Schnittstelle und zur FocusManager-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#) und unter „Arbeiten mit dem FocusManager“ auf Seite 29.

Die Änderungen an den Parametern der einzelnen Slider-Instanzen, die beim Authoring im Eigenschaften- oder im Komponenten-Inspektor vorgenommen werden, werden jeweils in einer Live-Vorschau angezeigt.

## Parameter der Slider-Komponente

Die folgenden Authoring-Parameter können für jede Instanz der Slider-Komponente im Eigenschaften- oder Komponenten-Inspektor festgelegt werden: `direction`, `liveDragging`, `maximum`, `minimum`, `snapInterval`, `tickInterval` und `value`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur Slider-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Erstellen einer Anwendung mit der Slider-Komponente

Im folgenden Beispiel wird eine Slider-Instanz erstellt, damit die Besucher einer hypothetischen Veranstaltung ihren Zufriedenheitsgrad angeben können. Der Benutzer verschiebt den Regler nach rechts oder links, um mehr bzw. weniger Zufriedenheit auszudrücken.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine Label-Komponente aus dem Bedienfeld „Komponenten“ auf die Mitte der Bühne.
  - Weisen Sie den Instanznamen **valueLabel** zu.
  - Weisen Sie den Wert **0 percent** dem Parameter `text` zu.
- 3 Ziehen Sie eine Slider-Komponente aus dem Bedienfeld „Komponenten“ und platzieren Sie sie unter `value_lbl`.
  - Geben Sie den Instanznamen **aSlider** ein.
  - Weisen Sie eine Breite (B) von **200** zu.
  - Weisen Sie eine Höhe (H) von **10** zu.
  - Weisen Sie einen Wert von **100** dem Parameter `maximum` zu.
  - Weisen Sie einen Wert von **10** sowohl dem Parameter `snapInterval` als auch dem Parameter `tickInterval` zu.
- 4 Ziehen Sie eine weitere Label-Instanz aus dem Bedienfeld „Bibliothek“ und platzieren Sie sie unterhalb von `aSlider`.
  - Geben Sie den Instanznamen **promptLabel** ein.
  - Weisen Sie eine Breite (B) von 250 zu.
  - Weisen Sie eine Höhe (H) von 22 zu.

- Geben Sie **Please indicate your level of satisfaction** für den Parameter `text` ein.

- 5 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“.

Während der Benutzer den Regler von einem Intervall zum nächsten zieht, aktualisiert ein Listener für das `SliderEvent.CHANGE`-Ereignis die `text`-Eigenschaft der Beschriftung `valueLabel`, sodass der angezeigte Prozentwert der Reglerposition entspricht.

## Erstellen einer Anwendung mit der Slider-Komponente mithilfe von ActionScript-Code

Im folgenden Beispiel wird mithilfe von ActionScript-Code ein Schieberegler erstellt. Es wird das Bild einer Blume geladen und mit einem Schieberegler kann der Benutzer das Bild verblassen lassen oder in frischen, leuchtenden Farben darstellen. Dazu wird die `alpha`-Eigenschaft so geändert, dass sie dem Wert des Schiebereglers entspricht.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die Label-Komponente und die Slider-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ des aktuellen Dokuments.  
Hierdurch werden die Komponenten zur Bibliothek hinzugefügt, sie werden jedoch nicht in der Anwendung angezeigt.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein, um Komponenteninstanzen zu erstellen und zu positionieren:

```
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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.
- 5 Bewegen Sie den Schieberegler nach links, um das Bild blasser werden zu lassen, und nach rechts, um es farbkräftig darzustellen.

## Arbeiten mit der TextArea-Komponente

Die TextArea-Komponente (Textfeld) erweitert das native ActionScript-TextField-Objekt. Die TextArea-Komponente können Sie einsetzen, um Text anzuzeigen, aber auch um Text zu bearbeiten und, wenn die Eigenschaft `editable` auf `true` gesetzt ist, um eine Texteingabe aufzunehmen. Die Komponente kann mehrere Textzeilen anzeigen oder empfangen, lange Textzeilen werden umbrochen, wenn die Eigenschaft `wordWrap` auf `true` gesetzt ist. Anhand der Eigenschaft `restrict` können Sie die Anzahl der Zeichen einschränken, die ein Benutzer eingeben darf, und mit `maxChars` können Sie angeben, wieviel Zeichen ein Benutzer maximal eingeben darf. Ragt der Text über die horizontalen oder vertikalen Grenzen des Textbereichs hinaus, erscheint automatisch eine horizontale bzw. vertikale Bildlaufleiste, es sei denn, die damit verbundenen Eigenschaften `horizontalScrollPolicy` und `verticalScrollPolicy` wurden auf `off` gesetzt.

Sie können die TextArea-Komponente überall dort einsetzen, wo ein Textfeld mit mehreren Zeilen benötigt wird. Sie können die TextArea-Komponente beispielsweise für ein Kommentarfeld in einem Formular verwenden. Sie können einen Listener einrichten, mit dem überprüft wird, ob das Feld leer ist, wenn der Benutzer zum Verlassen des Felds die Tabulatortaste drückt. In diesem Fall kann der Listener beispielsweise die Fehlermeldung anzeigen, dass ein Kommentar in dem Feld eingegeben werden muss.

Wenn Sie ein Textfeld mit nur einer Zeile benötigen, verwenden Sie die TextInput-Komponente.

Um die Formatierung innerhalb der TextArea-Instanz zu ändern, können Sie mithilfe der Methode `setStyle()` die Eigenschaft `textFormat` ändern. Sie können eine TextArea-Komponente auch mit HTML-Tags formatieren. Dazu verwenden Sie in ActionScript die Eigenschaft `htmlText` und wenn Sie die Eigenschaft `displayAsPassword` auf `true` setzen, werden anstelle des Texts ausschließlich Sternchen angezeigt. Wenn Sie die Eigenschaft `condenseWhite` auf `true` setzen, entfernt Flash alle zusätzlichen Leerräume aufgrund von Leerzeichen, Zeilenumbrüchen, usw. Auf den Text, der sich bereits im Steuerelement befindet, hat dies keinen Einfluss.

## Benutzerinteraktion mit der TextArea-Komponente

TextArea-Komponenten können in einer Anwendung aktiviert und deaktiviert werden. In deaktiviertem Zustand werden Eingaben über die Maus oder die Tastatur nicht angenommen. In aktiviertem Zustand gelten für dieses Objekt dieselben Fokus-, Auswahl- und Navigationsregeln wie für das ActionScript-Objekt „TextField“. Wenn eine TextArea-Instanz den Fokus hat, können Sie diese mithilfe der folgenden Tasten steuern:

Schlüssel	Beschreibung
Pfeiltasten	Versetzen die Einfügemarke innerhalb des Texts um je ein Zeichen nach oben, unten, links und rechts; vorausgesetzt, der Text ist bearbeitbar.
Bild-ab	Versetzt die Einfügemarke an das Ende des Texts; vorausgesetzt, der Text ist bearbeitbar.
Bild-auf	Versetzt die Einfügemarke an den Textanfang, wenn der Text bearbeitbar ist.
Umschalt+Tab	Bewegt den Fokus zum vorigen Objekt in der Tabulatorschleife.
Tab	Bewegt den Fokus zum nächsten Objekt in der Tabulatorschleife.

Weitere Informationen zum Steuern des Fokus finden Sie im Abschnitt zur FocusManager-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#) und unter „Arbeiten mit dem FocusManager“ auf Seite 29.

## Parameter der TextArea-Komponente

Die folgenden Authoring-Parameter können für jede Instanz der TextArea-Komponente im Eigenschaften- oder Komponenten-Inspektor festgelegt werden: `condenseWhite`, `editable`, `horizontalScrollPolicy`, `maxChars`, `restrict`, `text`, `verticalScrollPolicy` und `wordwrap`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur TextArea-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

Die Änderungen an den Parametern der einzelnen TextArea-Instanzen, die beim Authoring im Eigenschaften- oder Komponenteninspektor vorgenommen werden, werden jeweils in einer Live-Vorschau angezeigt. Wenn eine Bildlaufleiste benötigt wird, wird diese in der Vorschau zwar angezeigt, funktioniert jedoch nicht. Außerdem kann in der Live-Vorschau weder Text ausgewählt noch Text in der Komponenteninstanz auf der Bühne eingegeben werden.

Mithilfe von ActionScript können Sie diese und weitere Optionen für die TextArea-Komponente über ihre Eigenschaften, Methoden und Ereignisse steuern. Weitere Informationen finden Sie im Abschnitt zur TextArea-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## So erstellen Sie eine Anwendung mit der TextArea-Komponente

In der folgenden Anleitung wird erläutert, wie beim Authoring eine TextArea-Komponente in eine Anwendung eingefügt wird. Im Beispiel wird die Ereignisprozedur `focusOut` für die TextArea-Instanz eingerichtet. Die Prozedur überprüft, ob der Benutzer etwas in das Textfeld eingegeben hat, bevor der Fokus auf einen anderen Teil der Benutzeroberfläche gelegt wird.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine TextArea-Komponente vom Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz **aTa**. Übernehmen Sie für alle Parameter die Standardeinstellungen.
- 3 Ziehen Sie eine zweite TextArea-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne, platzieren Sie sie unter der ersten und nennen Sie die Instanz **bTa**. Übernehmen Sie für alle Parameter die Standardeinstellungen.
- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
import flash.events.FocusEvent;

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

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

In diesem Beispiel werden die im Textbereich `aTa` zulässigen Zeichen auf Kleinbuchstaben, das Komma, das Hochkomma (Apostroph) und Leerzeichen beschränkt. Zudem werden zum Textbereich `aTa` Ereignisprozeduren für die Ereignisse `change`, `KEY_FOCUS_CHANGE` und `MOUSE_FOCUS_CHANGE` eingerichtet. Die Funktion `changeHandler()` bewirkt, dass der Text, der in das Textfeld `aTa` eingegeben wird, automatisch im Textfeld `bTa` angezeigt wird, indem `aTa.text` bei jedem `change`-Ereignis `bTa.text` zugewiesen wird. Die Funktion `k_m_fHandler()` für die Ereignisse `KEY_FOCUS_CHANGE` und `MOUSE_FOCUS_CHANGE` verhindert, dass der Benutzer die Tabulatortaste drücken kann, um den Fokus auf das nächste Steuerelement zu setzen, ohne vorher Text einzugeben. Dies wird erreicht, indem das Standardverhalten verhindert wird.

- 5 Wählen Sie „Steuerung“ > „Film testen“.

Wenn Sie die Tabulatortaste drücken, um den Fokus in den zweiten Textbereich zu setzen, ohne Text einzugeben, wird eine Fehlermeldung angezeigt und der Fokus kehrt zurück auf den ersten Textbereich. Während Sie in den ersten Textbereich Text eingeben, können Sie sehen, dass er im zweiten Textbereich dupliziert wird.

## Erstellen einer TextArea-Instanz mithilfe von ActionScript-Code

Im folgenden Beispiel wird mithilfe von ActionScript-Code eine TextArea-Komponente erstellt. Die Eigenschaft `condenseWhite` wird auf `true` gesetzt, um Leerräume zu vermeiden, und der Eigenschaft `htmlText` wird Text zugewiesen, um die HTML-Formatierungsattribute nutzen zu können.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine TextArea-Komponente in das Bedienfeld „Bibliothek“.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:



```
import fl.controls.TextArea;

var aTa:TextArea = new TextArea();

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

In diesem Beispiel wird die Eigenschaft `htmlText` verwendet, um einen Textblock mit HTML-Tags für Fettsatz und Unterstreichung auszuzeichnen und im Textbereich `a_ta` anzuzeigen. In diesem Beispiel wird zudem die Eigenschaft `condenseWhite` auf `true` gesetzt, um Leerräume innerhalb des Textblocks auf ein einzelnes Leerzeichen zu reduzieren. Die Methode `setSize()` stellt die Höhe und Breite des Textbereichs ein und die Methode `move()` seine Position. Die Methode `addChild()` schließlich, platziert die `TextArea`-Instanz auf der Bühne.

4 Wählen Sie „Steuerung“ > „Film testen“.

## Arbeiten mit der TextInput-Komponente

Die `TextInput`-Komponente (Texteingabefeld) ist eine einzeilige Textkomponente, die das native `ActionScript`-Objekt „`TextField`“ erweitert. Wenn Sie ein Textfeld mit mehreren Zeilen benötigen, verwenden Sie die `TextArea`-Komponente. Sie können die `TextInput`-Komponente beispielsweise für ein Kennwortfeld in einem Formular verwenden. Sie können auch einen Listener einrichten, mit dem überprüft wird, ob das Feld genügend Zeichen enthält, wenn der Benutzer zum Verlassen des Felds die Tabulatortaste drückt. In diesem Fall kann der Listener beispielsweise eine Fehlermeldung anzeigen, die darauf hinweist, dass die richtige Anzahl an Zeichen in dem Feld eingegeben werden muss.

Um die Formatierung innerhalb der `TextInput`-Instanz zu ändern, können Sie mithilfe der Methode `setStyle()` die Eigenschaft `textFormat` ändern. Die `TextInput`-Komponente kann auch im HTML-Format oder als Kennwortfeld formatiert werden, in dem der Text unkenntlich dargestellt wird.

## Benutzerinteraktion mit der TextInput-Komponente

`TextInput`-Komponenten können in einer Anwendung aktiviert und deaktiviert werden. In deaktiviertem Zustand werden Eingaben über die Maus oder die Tastatur nicht empfangen. In aktiviertem Zustand gelten für dieses Objekt dieselben Fokus-, Auswahl- und Navigationsregeln wie für das `ActionScript`-Objekt „`TextField`“. Wenn eine `TextInput`-Instanz den Fokus hat, können Sie diese auch mithilfe der folgenden Tasten steuern:

Schlüssel	Beschreibung
Pfeiltasten	Verschieben die Einfügemarke um je ein Zeichen nach links oder rechts.
Umschalt+Tab	Verschiebt den Fokus auf das vorherige Objekt.
Tab	Verschiebt den Fokus auf das nächste Objekt.

Weitere Informationen zum Steuern des Fokus finden Sie im Abschnitt zur `FocusManager`-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#) und unter „Arbeiten mit dem `FocusManager`“ auf Seite 29.

Die Änderungen an den Parametern der einzelnen TextInput-Instanzen, die beim Authoring im Eigenschaften- oder Komponenten-Inspektor vorgenommen werden, werden jeweils in einer Live-Vorschau angezeigt. Außerdem kann in der Live-Vorschau weder Text ausgewählt noch Text in der Komponenteninstanz auf der Bühne eingegeben werden.

Wenn Sie einer Anwendung die TextInput-Komponente hinzufügen, können Sie diese im Bedienfeld „Eingabehilfen“ für Bildschirmleseprogramme zugänglich machen.

## Parameter der TextInput-Komponente

Die folgenden Authoring-Parameter können für jede Instanz der TextInput-Komponente im Eigenschaften- oder Komponenten-Inspektor festgelegt werden: `editable`, `displayAsPassword`, `maxChars`, `restrict` und `text`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur TextInput-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

Mithilfe von ActionScript können Sie diese und weitere Optionen für die TextInput-Komponente über ihre Eigenschaften, Methoden und Ereignisse steuern. Weitere Informationen finden Sie im Abschnitt zur TextInput-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Erstellen einer Anwendung mit der TextInput-Komponente

In der folgenden Anleitung wird erläutert, wie eine TextInput-Komponente in eine Anwendung eingefügt wird. In dem Beispiel werden zwei TextInput-Felder erstellt: eines zur Eingabe und eines zur Bestätigung eines Kennworts. Es wird ein Ereignis-Listener verwendet, der sicher stellt, dass mindestens acht Zeichen eingegeben wurden, und dass der Text für die zwei Felder übereinstimmt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine Label-Komponente vom Bedienfeld „Komponenten“ auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
  - Geben Sie den Instanznamen **pwdLabel** ein.
  - Geben Sie als Breite (B) den Wert **100** ein.
  - Geben Sie für X den Wert **50** ein.
  - Geben Sie für Y den Wert **150** ein.
  - Geben Sie auf der Registerkarte „Parameter“ für den Parameter „text“ den Wert **Password:** ein.
- 3 Ziehen Sie eine zweite Label-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und geben Sie die folgenden Werte ein:
  - Geben Sie den Instanznamen **confirmLabel** ein.
  - Geben Sie als Breite (B) den Wert **100** ein.
  - Geben Sie für X den Wert **50** ein.
  - Geben Sie für Y den Wert **200** ein.
  - Geben Sie auf der Registerkarte „Parameter“ für den Parameter „text“ den Wert **Confirm Password:** ein.
- 4 Ziehen Sie eine TextInput-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und geben Sie die folgenden Werte ein:
  - Geben Sie den Instanznamen **pwdTi** ein.
  - Geben Sie als Breite (B) den Wert **150** ein.

- Geben Sie für X den Wert **190** ein.
  - Geben Sie für Y den Wert **150** ein.
  - Doppelklicken Sie im Bereich `Parameter` auf den Wert für den Parameter **`displayAsPassword`** und wählen Sie „true“ aus. Dadurch werden anstelle des eingegebenen Texts ausschließlich Sternchen angezeigt.
- 5 Ziehen Sie eine zweite `TextInput`-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und geben Sie die folgenden Werte ein:
- Geben Sie den Instanznamen **`confirmTi`** ein.
  - Geben Sie als Breite (B) den Wert **150** ein.
  - Geben Sie für X den Wert **190** ein.
  - Geben Sie für Y den Wert **200** ein.
  - Doppelklicken Sie auf der Registerkarte „Parameter“ auf den Wert für den Parameter `displayAsPassword` und wählen Sie **`true`** aus. Dadurch werden anstelle des eingegebenen Texts ausschließlich Sternchen angezeigt.
- 6 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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);
```

Mit diesem Code wird eine Ereignisprozedur `enter` für die `TextInput`-Instanz `confirmTi` erstellt. Wenn zwei Kennwörter nicht übereinstimmen oder der Benutzer weniger als acht Zeichen eingibt, wird im Beispiel die Meldung „Password is incorrect. Please reenter it.“ angezeigt. Wenn die Kennwörter mindestens acht Zeichen enthalten und übereinstimmen, wird im Beispiel der eingegebene Wert im Bedienfeld „Ausgabe“ angezeigt.

- 7 Wählen Sie „Steuerung“ > „Film testen“.

### Erstellen einer `TextInput`-Instanz mithilfe von ActionScript-Code

Im folgenden Beispiel wird mithilfe von ActionScript-Code eine `TextInput`-Komponente erstellt. Es wird zudem eine Beschriftung erstellt, die verwendet wird, um den Benutzer zur Eingabe des Namens aufzufordern. Im Beispiel wird die Eigenschaft `restrict` dieser Komponente festgelegt, sodass nur Großbuchstaben und Kleinbuchstaben, ein Punkt und ein Leerzeichen eingegeben werden können. Es wird auch ein `TextFormat`-Objekt erstellt, das zur Textformatierung sowohl in der Label- als auch in der `TextInput`-Komponente verwendet wird.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine `TextInput`-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 3 Ziehen Sie eine Label-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

**Verwenden der UI-Komponenten**

```
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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.

## Arbeiten mit der TileList-Komponente

Die TileList-Komponente (gekachelte Liste) ist eine Liste aus Zeilen und Spalten, deren Inhalt von einem Datenprovider bezogen wird. Dabei verweist ein Element (*item*) auf eine Dateneinheit, die in einer Zelle der gekachelten Liste gespeichert ist. Ein Element, das vom Datenprovider stammt, hat in der Regel eine `label`- und eine `source`-Eigenschaft. Die Eigenschaft `label` identifiziert den Inhalt, der in einer Zelle angezeigt werden soll, und `source` liefert den zugehörigen Wert.

Sie können eine Array-Instanz erstellen oder von einem Server abrufen. Die TileList-Komponente verfügt über Methoden, die als Proxy für ihren Datenprovider dienen, z. B. die Methoden `addItem()` und `removeItem()`. Wenn der Liste kein externer Datenprovider zugeordnet wird, erstellen diese Methoden automatisch eine Datenprovider-Instanz, die durch `List.dataProvider` zur Verfügung gestellt wird.

### Benutzerinteraktion mit der TileList-Komponente

Eine gekachelte Liste rendert jede Zelle anhand eines Sprites, das die ICellRenderer-Benutzeroberfläche implementiert. Diesen Renderer können Sie mit der TileList-Eigenschaft `cellRenderer` angeben. Der Standard-CellRenderer der TileList-Komponente ist „ImageCell“, mit dem ein Bild (Klasse, Bitmap, Instanz oder URL) und eine optionale Beschriftung angezeigt wird. Die Beschriftung ist eine einzelne Zeile, die immer am unteren Rand der Zelle angezeigt wird. In einer gekachelten Liste ist ein Bildlauf nur in einer Richtung möglich.

Wenn eine TileList-Instanz den Fokus hat, können Sie diese auch mithilfe der folgenden Tasten steuern:

Schlüssel	Beschreibung
Nach-oben- oder Nach-unten-Taste	Navigation nach unten oder nach oben innerhalb einer Spalte. Hat die Eigenschaft <code>allowMultipleSelection</code> den Wert <code>true</code> , können Sie diese Tasten zusammen mit der Umschalttaste verwenden, um mehrere Zellen auszuwählen.
Nach-links- oder Nach-rechts-Taste	Navigation nach links oder rechts innerhalb einer Spalte. Hat die Eigenschaft <code>allowMultipleSelection</code> den Wert <code>true</code> , können Sie diese Tasten zusammen mit der Umschalttaste verwenden, um mehrere Zellen auszuwählen.
Startseite	Wählt die erste Zeile in einer gekachelten Liste aus. Wenn der Eigenschaft <code>allowMultipleSelection</code> der Wert <code>true</code> zugewiesen wurde, können Sie durch Gedrückthalten der Umschalttaste und gleichzeitiges Drücken der Taste „Pos1“ alle Zellen von der aktuellen Auswahl bis zur ersten Zelle auswählen.
Ende	Wählt die letzte Zeile in einer gekachelten Liste aus. Wenn der Eigenschaft <code>allowMultipleSelection</code> der Wert <code>true</code> zugewiesen wurde, können Sie durch Gedrückthalten der Umschalttaste und gleichzeitiges Drücken der Taste „Ende“ alle Zellen von der aktuellen Auswahl bis zur letzten Zelle auswählen.
Strg	Wenn der Eigenschaft <code>allowMultipleSelection</code> der Wert <code>true</code> zugewiesen wurde, können Sie mehrere Zellen in beliebiger Reihenfolge auswählen.

Die einer Anwendung hinzugefügten `TileList`-Komponenten können Sie Bildschirmleseprogrammen anhand der folgenden `ActionScript`-Codezeilen zugänglich machen:

```
import fl.accessibility.TileListAccImpl;

TileListAccImpl.enableAccessibility();
```

Diese Anweisung ist unabhängig von der Anzahl der Instanzen nur einmal pro Komponente erforderlich. Weitere Informationen finden Sie im Handbuch *Verwenden von Flash* im Kapitel 18, „Barrierefreie Inhalte erstellen“.

## Parameter der `TileList`-Komponente

Die folgenden `Authoring`-Parameter können für jede Instanz der `TileList`-Komponente im `Eigenschaften-` oder `Komponenten-Inspektor` festgelegt werden: `allowMultipleSelection`, `columnCount`, `columnWidth`, `dataProvider`, `direction`, `horizontalScrollLineSize`, `horizontalScrollPageSize`, `labels`, `rowCount`, `rowHeight`, `ScrollPolicy`, `verticalScrollLineSize` und `verticalScrollPageSize`. Jeder dieser Parameter besitzt eine entsprechende `ActionScript`-Eigenschaft desselben Namens. Informationen über die Verwendung des Parameters „`dataProvider`“ finden Sie im Abschnitt „[Verwenden des Parameters „dataProvider“](#)“ auf Seite 31.

Sie können zusätzliche Optionen für `Loader`-Instanzen mit den Methoden, `Eigenschaften` und `Ereignissen` von `ActionScript` festlegen. Weitere Informationen finden Sie in der `TileList`-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Erstellen einer Anwendung mit der `TileList`-Komponente

Dieses Beispiel verwendet `Movieclips`, um eine gekachelte Liste mit einem Array von Farben zu füllen.

- 1 Erstellen Sie ein neues `Flash`-Dokument (`ActionScript 3.0`).
- 2 Ziehen Sie eine `TileList`-Komponente auf die Bühne, und geben Sie ihr den Instanznamen `aTL`.
- 3 Öffnen Sie das Bedienfeld „`Aktionen`“, wählen Sie in der Hauptzeitleiste `Bild 1` aus und geben Sie den folgenden `ActionScript`-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“, um die Anwendung zu testen.

### Erstellen einer TileList-Komponenteninstanz mithilfe von ActionScript-Code

In diesem Beispiel wird eine TileList-Instanz dynamisch erstellt und es werden ihr Instanzen der ColorPicker-, ComboBox-, NumericStepper- und CheckBox-Komponenten hinzugefügt. Es wird ein Array erstellt, das Beschriftungen und die Namen der anzuzeigenden Komponenten enthält. Dieses Array (dp) wird der TileList-Eigenschaft dataProvider der zugewiesen. Mit den Eigenschaften columnWidth und rowHeight und der Methode setSize() wird das Layout der gekachelten Liste definiert, mit der Methode move() wird sie auf der Bühne positioniert und mit dem Format contentPadding wird der Leerraum zwischen den Rändern der TileList-Instanz und ihrem Inhalt festgelegt. Zudem wird mithilfe der Methode sortItemsOn() der Inhalt nach seinen Beschriftungen sortiert.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die folgenden Komponenten aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“: „ColorPicker“, „ComboBox“, „NumericStepper“, „CheckBox“ und „TileList“.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

**Verwenden der UI-Komponenten**

```
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 Wählen Sie „Steuerung“ > „Film testen“, um die Anwendung zu testen.

## Arbeiten mit der UILoader-Komponente

Die UILoader-Komponente (UI-Ladeobjekt) ist ein Container, der SWF-, JPEG-, progressive JPEG-, PNG- und GIF-Dateien anzeigen kann. Sie können ein UI-Ladeobjekt immer dann verwenden, wenn Sie Inhalt von einem entfernten Speicherort abrufen und in eine Flash-Anwendung übertragen müssen. Beispielsweise können Sie mit einem UI-Ladeobjekt ein Firmenlogo (JPEG-Datei) in ein Formular einfügen. Sie könnten eine UILoader-Komponente etwa in einer Anwendung verwenden, in der Fotos angezeigt werden. Sie verwenden die Methode `load()`, um den Inhalt zu laden, die Eigenschaft `percentLoaded`, um festzustellen, welcher Anteil des Inhalts jeweils geladen ist, und das Ereignis `complete`, um festzustellen, wann der Ladevorgang abgeschlossen ist.

Um die Größe des Inhalts anzupassen, können Sie entweder den Inhalt skalieren oder die Größe der UILoader-Instanz ändern. Standardmäßig wird der Inhalt so skaliert, dass er in das UILoader-Objekt passt. Sie können den Inhalt auch zur Laufzeit laden und den Ladevorgang überwachen (obwohl der Inhalt nach dem Laden zwischengespeichert wird, sodass der Fortschritt schnell auf 100% springt). Wenn Sie beim Laden von Inhalt in das UILoader-Objekt eine Position angeben, müssen Sie die X- und Y-Koordination als 0,0 angeben.

## Benutzerinteraktion mit der UILoader-Komponente

Eine UILoader-Komponente kann nicht den Fokus erhalten. Jedoch kann der Inhalt, der in der UILoader-Komponente geladen ist, den Fokus annehmen und mit ihm interagieren. Weitere Informationen zum Steuern des Fokus finden Sie im Abschnitt zur `FocusManager`-Klasse im *ActionScript 3.0 Referenzhandbuch für Flash Professional* und unter „Arbeiten mit dem `FocusManager`“ auf Seite 29.

## Parameter der UILoader-Komponente

Die folgenden Authoring-Parameter können für jede Instanz einer UILoader-Komponente im Eigenschaften- oder Komponenteninspektor festgelegt werden: `autoLoad`, `maintainAspectRatio`, `source` und `scaleContent`. Jeder dieser Parameter besitzt eine entsprechende ActionScript-Eigenschaft desselben Namens.

Eine Live-Vorschau der einzelnen UILoader-Instanzen spiegelt die Änderungen wieder, die im Eigenschaften- oder Komponenten-Inspektor beim Authoring vorgenommen wurden.

Sie können zusätzliche Optionen für UILoader-Instanzen mit den Methoden, Eigenschaften und Ereignissen von ActionScript festlegen. Weitere Informationen finden Sie im Abschnitt zur UILoader-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Erstellen einer Anwendung mit der UILoader-Komponente

In der folgenden Anleitung wird erläutert, wie beim Authoring eine UILoader-Komponente in eine Anwendung eingefügt wird. In dem Beispiel lädt das UI-Ladeobjekt ein GIF-Bild eines Logos.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die UILoader-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
- 3 Geben Sie im Eigenschafteninspektor den Instanznamen **aUI** ein.
- 4 Wählen Sie den Loader auf der Bühne und im Komponenteninspektor aus, und geben Sie <http://www.helpexamples.com/images/logo.gif> für den Parameter `source` ein.

## Erstellen einer UILoader-Komponenteninstanz mithilfe von ActionScript-Code

In diesem Beispiel wird mithilfe von ActionScript-Code eine UILoader-Komponente erstellt und das JPEG-Bild einer Blume geladen. Wenn das Ereignis `complete` eintritt, wird die Anzahl der geladenen Bytes im Bedienfeld „Ausgabe“ angezeigt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die UILoader-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“.



## Arbeiten mit der UIScrollBar-Komponente

Mithilfe der UIScrollBar-Komponente können Sie einem Textfeld eine Bildlaufleiste hinzufügen. Dies ist beim Authoring oder mithilfe von ActionScript zur Laufzeit möglich. Um die UIScrollBar-Komponente zu verwenden, erstellen Sie auf der Bühne ein Textfeld und ziehen die UIScrollBar-Komponente aus dem Bedienfeld „Komponenten“ in einen beliebigen Quadranten des Textfeld-Begrenzungsrahmens.

Wenn die gesamte Bildlaufleiste kleiner ist als die beiden Bildlaufpfeile zusammen, wird sie nicht korrekt angezeigt. Einer der Pfeile wird dann von dem anderen verdeckt. Flash bietet dafür keine Fehlerprüfung an. In diesem Fall empfiehlt es sich, die Bildlaufleiste mithilfe von ActionScript auszublenden. Wenn die Bildlaufleiste so klein ist, dass das Bildlauffeld keinen Platz hat, blendet Flash das Bildlauffeld aus.

Die UIScrollBar-Komponente funktioniert wie eine übliche Bildlaufleiste. Sie enthält Bildlaufpfeile an den beiden Enden und dazwischen die Spur der Bildlaufleiste sowie das Bildlauffeld. Sie kann an einem beliebigen Rand eines Textfelds angefügt und vertikal oder horizontal verwendet werden.

Informationen zum TextField finden Sie im Abschnitt zur TextField-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Benutzerinteraktion mit der UIScrollBar-Komponente

Im Gegensatz zu vielen anderen Komponenten kann die UIScrollBar-Komponente auf fortgesetzte Mauseingaben reagieren. Der Benutzer kann also die Maustaste gedrückt halten anstatt wiederholt zu klicken.

Für die UIScrollBar-Komponente gibt es keine Interaktion über die Tastatur.

## Parameter der UIScrollBar-Komponente

Die folgenden Authoring-Parameter können für jede Instanz einer UIScrollBar-Komponente im Eigenschaften- oder Komponenteninspektor festgelegt werden: `direction` und `scrollTargetName`. Jeder dieser Parameter besitzt eine entsprechende ActionScript-Eigenschaft desselben Namens.

Sie können zusätzliche Optionen für UIScrollBar-Instanzen mit den Klassenmethoden, Eigenschaften und Ereignissen von ActionScript festlegen. Weitere Informationen finden Sie im Abschnitt zur UIScrollBar-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Erstellen einer Anwendung mit der UIScrollBar-Komponente

Im Folgenden wird beschrieben, wie Sie beim Authoring eine UIScrollBar-Komponente in eine Anwendung einfügen.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Erstellen Sie ein dynamisches Textfeld, das groß genug für ein oder zwei Textzeilen ist, und geben Sie der Instanz im Eigenschafteninspektor den Namen **myText**.
- 3 Wählen Sie im Eigenschafteninspektor als Zeilentyp für das Texteingabefeld „Mehrzeilig“ oder „Mehrzeilig, kein Umbruch“, wenn Sie eine horizontale Bildlaufleiste verwenden möchten.
- 4 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie Bild 1 in der Zeitleiste aus und geben Sie den folgenden ActionScript-Code ein, um die Eigenschaft `text` so zu füllen, dass ein Benutzer einen Bildlauf ausführen muss, um den gesamten Text zu sehen:

```
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."
```

**Hinweis:** Achten Sie darauf, dass das Textfeld auf der Bühne klein genug ist, damit der Text nicht vollständig sichtbar ist. Andernfalls wird die Bildlaufleiste nicht angezeigt oder erscheint lediglich als zwei Zeilen ohne Grip (welchen Sie ziehen, um einen Bildlauf für den Inhalt auszuführen).

- 5 Vergewissern Sie sich, dass die Ausrichtung an Objekten aktiviert ist („Ansicht“ > „Ausrichten“ > „An Objekten ausrichten“).
- 6 Ziehen Sie eine UIScrollBar-Instanz aus dem Bedienfeld „Komponenten“ auf das Texteingabefeld, nahe der Seite, an der die Bildlaufleiste eingefügt werden soll. Die Komponente muss das Textfeld beim Loslassen der Maustaste überlappen, damit sie korrekt an das Feld gebunden wird. Geben Sie den Instanznamen **mySb** ein.

Die Eigenschaft `scrollTargetName` der Komponente wird im Eigenschaften- und Komponenten-Inspektor automatisch mit dem Instanznamen des Textfelds versehen. Falls dies auf der Registerkarte „Parameter“ nicht zu sehen ist, haben Sie die UIScrollBar-Instanz evtl. nicht genug überlappt.

- 7 Wählen Sie „Steuerung“ > „Film testen“.

### Erstellen einer UIScrollBar-Komponenteninstanz mithilfe von ActionScript-Code

Sie können eine UIScrollBar-Instanz mithilfe von ActionScript erstellen und zur Laufzeit mit einem Textfeld verknüpfen. Im folgenden Beispiel wird eine horizontal ausgerichtete UIScrollBar-Instanz erstellt und unten an ein Textfeld mit dem Instanznamen **myTxt** angehängt. Das Textfeld wird mit Text aus einer URL-Adresse gefüllt. Im Beispiel wird zudem die Größe der Bildlaufleiste so eingestellt, dass sie der Größe des Textfelds entspricht.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die ScrollBar-Komponente in das Bedienfeld „Bibliothek“.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden ActionScript-Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“.

# Kapitel 5: Anpassen der UI-Komponenten

## Überblick über die Anpassung von UI-Komponenten

Sie können die Darstellung der Komponenten in Ihren Anwendungen anpassen, indem Sie eines oder beide der folgenden Elemente ändern:

**Stile** Jede Komponente verfügt über eine Reihe von Stilen, die Sie einstellen können, um anzugeben, welche Werte Flash für das Erscheinungsbild der Komponente verwendet. Stile definieren meist Skins und Symbole, die für die verschiedenen Zustände der Komponente verwendet werden sollen. Außerdem geben sie die Textformatierung und Auffüllungswerte an.

**Skins** Eine *Skin* besteht aus mehreren Symbolen, die das grafische Erscheinungsbild einer Komponente in einem bestimmten Zustand ausmachen. Während ein Stil die zu verwendende Skin angibt, handelt es sich bei einer Skin um ein grafisches Element, das Flash verwendet, um die Komponente zu zeichnen. Beim *Skinning* wird die Darstellung einer Komponente geändert, indem ihre Grafiken modifiziert oder ersetzt werden.

**Hinweis:** Das Standardaussehen der ActionScript 3.0-Komponenten ist mit einem Thema (Aeon Halo) vergleichbar, diese Skins sind jedoch in die Komponenten integriert. Die ActionScript 3.0-Komponenten unterstützen nicht die von den ActionScript 2.0-Komponenten unterstützten externen Themendateien.

## Festlegen von Stilen

Die Stile einer Komponente definieren im Allgemeinen Werte für die Skins, Symbole, Textformatierung und Auffüllung, die Flash zum Zeichnen der Komponente in verschiedenen Zuständen verwendet. Beispielsweise zeichnet Flash eine Schaltfläche im Down-Status (also wenn ein Benutzer auf die Schaltfläche klickt) mit einer anderen Skin als im Up- oder Normal-Status. Eine andere Skin wird auch für den deaktivierten Zustand verwendet, der auftritt, wenn die `enabled`-Eigenschaft auf `false` gesetzt wird.

Stile für Komponenten können auf verschiedenen Ebenen festgelegt werden: Dokument, Klasse und Instanz. Außerdem können einige Stileigenschaften von einer übergeordneten Komponente übernommen werden. So übernimmt die List-Komponente beispielsweise ScrollBar-Stile von BaseScrollPane.

Sie können Stile festlegen, um eine Komponente wie folgt anzupassen:

- Festlegen von Stilen für eine Komponenteninstanz. Sie können die Farb- und Texteneigenschaften für eine einzelne Komponenteninstanz ändern. Dies ist in einigen Situationen hilfreich, kann jedoch auch zeitintensiv sein, wenn Sie einzelne Eigenschaften für alle Komponenten in einem Dokument festlegen müssen.
- Festlegen von Stilen für alle Komponenten eines bestimmten Typs in einem Dokument. Wenn alle Komponenten eines bestimmten Typs, wie beispielsweise Kontrollkästchen und Schaltflächen, im Dokument ein einheitliches Erscheinungsbild aufweisen sollen, können Sie Stile auf der Komponentenebene festlegen.

Die für Container festgelegten Stileigenschaften werden von den enthaltenen Komponenten übernommen.

Bei der Live-Vorschau von Komponenten auf der Bühne werden Änderungen an Stileigenschaften in Flash nicht angezeigt.

## Hintergrundinformationen zu Stileinstellungen

Bei der Verwendung von Stilen sollten Sie die folgenden wichtigen Aspekte berücksichtigen:

**Vererbung** Eine untergeordnete Komponente übernimmt standardmäßig einen Stil von der übergeordneten Komponente. Die Vererbung von Stilen kann nicht in ActionScript festgelegt werden.

**Vorrang** Wenn ein Stil einer Komponente auf mehrere Arten festgelegt wurde, verwendet Flash den ersten Stil in der Reihenfolge des Vorrangs. Flash sucht in der folgenden Reihenfolge nach Stilen, bis ein Wert gefunden wird:

- 1 Flash sucht nach einer Stileigenschaft in der Komponenteninstanz.
- 2 Wenn es sich um einen Vererbungsstil handelt, durchsucht Flash die übergeordnete Hierarchie nach einem übernommenen Wert.
- 3 Flash sucht in der Komponente nach dem Stil.
- 4 Flash sucht nach einer globalen StyleManager-Einstellung.
- 5 Wenn die Eigenschaft immer noch nicht definiert ist, lautet der Eigenschaftswert `undefined`.

## Zugriff auf die Standardstile einer Komponente

Zum Zugriff auf die Standardstile einer Komponente können Sie die statische `getStyleDefinition()`-Methode für die Komponentenkategorie verwenden. Mit dem folgenden Code werden zum Beispiel die Standardstile für die ComboBox-Komponente abgerufen und die Standardwerte für die Eigenschaften `buttonWidth` und `downArrowDownSkin` angezeigt:

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

## Festlegen und Abrufen von Stilen für eine Komponenteninstanz

Jede UI-Komponenteninstanz kann die Methoden `setStyle()` und `getStyle()` direkt aufrufen, um einen Stil festzulegen oder abzurufen. Die folgende Syntax legt einen Stil und einen Wert für eine Komponenteninstanz fest:

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

Die folgende Syntax ruft einen Stil einer Komponenteninstanz ab:

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

Beachten Sie, dass die `getStyle()`-Methode das `type`-Objekt zurückgibt, da es mehrere Stile mit unterschiedlichen Datentypen zurückgeben kann. Mit dem folgenden Code wird beispielsweise der Schriftstil für die TextArea-Instanz (`aTa`) festgelegt und dann mit der `getStyle()`-Methode abgerufen. In diesem Beispiel wird der zurückgegebene Wert in ein `TextFormat`-Objekt umgewandelt, das einer `TextFormat`-Variable zugewiesen wird. Ohne diese Umwandlung würde bei der Kompilierung ein Fehler ausgegeben, da versucht würde, eine `Object`-Variable einer `TextFormat`-Variablen zuzuweisen.

```
import flash.text.TextFormat;

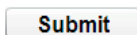
var tf:TextFormat = new TextFormat();
tf.font = "Georgia";
aTa.setStyle("textFormat", tf);
aTa.text = "Hello World!";
var aStyle:TextFormat = aTa.getStyle("textFormat") as TextFormat;
trace(aStyle.font);
```

## Festlegen von Texteigenschaften mithilfe von TextFormat

Mit dem `TextFormat`-Objekt können Sie den Text für eine Komponenteninstanz formatieren. Mit den Eigenschaften des `TextFormat`-Objekts können Sie verschiedene Textmerkmale festlegen, wie beispielsweise `bold`, `bullet`, `color`, `font`, `italic`, `size` und so weiter. Sie können diese Eigenschaften im `TextFormat`-Objekt festlegen und dann die `setStyle()`-Methode aufrufen, um die Eigenschaften auf eine Komponenteninstanz anzuwenden. Mit dem folgenden Code werden beispielsweise die Eigenschaften `font`, `size` und `bold` eines `TextFormat`-Objekts festgelegt und auf eine `Button`-Instanz angewendet:

```
/* 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);
```

Die folgende Abbildung zeigt die Auswirkungen dieser Einstellungen auf eine `Submit`-Schaltfläche:



Stileigenschaften, die mit `setStyle()` für eine Komponenteninstanz festgelegt werden, haben die höchste Priorität und setzen alle anderen Stileinstellungen außer Kraft. Je mehr Eigenschaften Sie mithilfe von `setStyle()` für eine einzelne Komponenteninstanz festlegen, desto langsamer wird die Komponente zur Laufzeit dargestellt.

## Festlegen eines Stils für alle Instanzen einer Komponente

Mit der statischen `setComponentStyle()`-Methode der `StyleManager`-Klasse können Sie einen Stil für alle Instanzen einer Komponenteklasse festlegen. So können Sie zum Beispiel die Textfarbe für alle Schaltflächen auf Rot einstellen, indem Sie zuerst eine `Button`-Komponente auf die Bühne ziehen und dann im Bedienfeld „Aktionen“ in Bild 1 der Zeitleiste den folgenden ActionScript-Code hinzufügen:

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

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

Alle Schaltflächen, die Sie der Bühne anschließend hinzufügen, haben rote Bezeichnungen.

## Festlegen eines Stils für alle Komponenten

Mit der statischen `setStyle()`-Methode der `StyleManager`-Klasse können Sie einen Stil für alle Komponenten festlegen.

- 1 Ziehen Sie eine `List`-Komponente auf die Bühne und geben Sie ihr den Instanznamen **aList**.
- 2 Ziehen Sie eine `Button`-Komponente auf die Bühne und geben Sie ihr den Instanznamen **aButton**.
- 3 Drücken Sie **F9** oder wählen Sie „Fenster“ > „Aktionen“, um das Bedienfeld „Aktionen“ zu öffnen, falls es nicht bereits angezeigt wird. Geben Sie den folgenden Code in Bild 1 der Zeitleiste ein, um die Textfarbe für alle Komponenten auf Rot einzustellen:

```
import fl.managers.StyleManager;

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

- 4 Fügen Sie im Bedienfeld „Aktionen“ den folgenden Code ein, um die Liste mit Text zu versehen:

```
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 Wählen Sie „Steuerung“ > „Film testen“ oder drücken Sie Strg+Eingabe, um den Code zu kompilieren und den Inhalt zu testen. Der Text der Schaltflächenbezeichnung und in der Liste sollte rot sein.

## Überblick über Skins

Die Darstellung einer Komponente wird über verschiedene grafische Elemente definiert, wie beispielsweise eine Kontur, eine Füllfarbe, Symbole und sogar andere Komponenten. Eine ComboBox-Komponente enthält beispielsweise eine List-Komponente, die wiederum eine ScrollBar-Komponente enthält. Die grafischen Elemente bilden gemeinsam die Darstellung der ComboBox. Das Erscheinungsbild einer Komponente basiert jedoch auf ihrem aktuellen Zustand. So sieht ein Kontrollkästchen ohne Bezeichnung in etwa folgendermaßen in Ihrer Anwendung aus:



*Ein Kontrollkästchen im normalen Up-Status*

Wenn Sie die Maustaste auf dem Kontrollkästchen gedrückt halten, ändert sich sein Erscheinungsbild wie folgt:



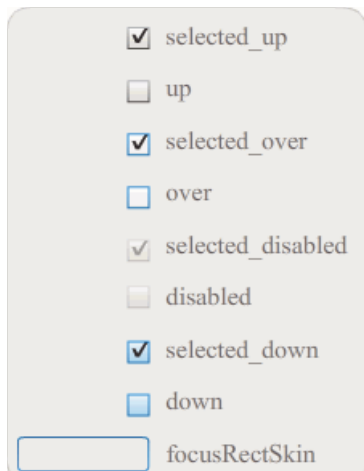
*Ein Kontrollkästchen im Down-Status*

Wenn Sie die Maustaste loslassen, wird das Kontrollkästchen wieder in seiner ursprünglichen Darstellung angezeigt, enthält jetzt jedoch ein Häkchen, um anzuzeigen, dass es ausgewählt wurde.



*Ein Kontrollkästchen im ausgewählten Status*

Die Symbole, welche die Komponente in ihren verschiedenen Zuständen darstellen, bezeichnet man zusammenfassend als *Skins*. Sie können das Erscheinungsbild einer Komponente in einigen oder allen Zuständen ändern, indem Sie ihre Skins wie jedes andere Flash-Symbol bearbeiten. Der Zugriff auf die Skins einer Komponente ist mit zwei verschiedenen Methoden möglich. Am einfachsten ist es, wenn Sie die Komponente auf die Bühne ziehen und darauf doppelklicken. Dadurch wird eine Palette mit den Skins der Komponente geöffnet, die für ein Kontrollkästchen folgendermaßen aussieht.



Die Skins eines Kontrollkästchens

Sie können auch vom Bedienfeld „Bibliothek“ aus auf die einzelnen Skins einer Komponente zugreifen. Wenn Sie eine Komponente auf die Bühne ziehen, kopieren Sie sie zusammen mit einem Ordner, der ihre Elemente und ggf. andere Komponenten enthält, auch in die Bibliothek. Wenn Sie beispielsweise eine ComboBox-Komponente auf die Bühne ziehen, enthält das Bedienfeld „Bibliothek“ auch die List-, ScrollBar- und TextInput-Komponenten, die in die ComboBox-Komponente integriert sind, sowie einen Ordner mit den Skins für jede dieser Komponenten und einen Ordner mit den gemeinsamen Elementen dieser Komponenten. Sie können die Skins für jede dieser Komponenten bearbeiten, indem Sie ihren Skin-Ordner (ComboBoxSkins, ListSkins, ScrollBarSkins oder TextInputSkins) öffnen und auf das Symbol für die gewünschte Skin doppelklicken. Wenn Sie zum Beispiel auf ComboBox\_downSkin doppelklicken, wird die Skin im Symbolbearbeitungsmodus geöffnet, wie in der folgenden Abbildung gezeigt:



ComboBox\_downSkin



## Erstellen einer neuen Skin

Wenn Sie eine Komponente im Dokument mit einem neuen Erscheinungsbild versehen möchten, können Sie die Skins der Komponente bearbeiten. Um auf die Skins einer Komponente zuzugreifen, doppelklicken Sie einfach auf der Bühne auf die Komponente, um eine Palette der Skins zu öffnen. Doppelklicken Sie dann auf die Skin, die Sie bearbeiten möchten, um sie im Symbolbearbeitungsmodus zu öffnen. Doppelklicken Sie zum Beispiel auf der Bühne auf die TextArea-Komponente, um ihre Elemente im Symbolbearbeitungsmodus zu öffnen. Stellen Sie die Vergrößerung bei Bedarf auf 400 % oder höher ein und bearbeiten Sie das Symbol, um seine Darstellung zu ändern. Die vorgenommenen Änderungen wirken sich auf alle Instanzen der Komponente im Dokument aus. Alternativ dazu können Sie im Bedienfeld „Bibliothek“ auf eine bestimmte Skin doppelklicken, um sie auf der Bühne im Symbolbearbeitungsmodus zu öffnen.

Sie können die Skins von Komponenten folgendermaßen ändern:

- Erstellen einer neuen Skin für alle Instanzen
- Erstellen von neuen Skins für einige Instanzen

### Erstellen einer Skin für alle Instanzen

Wenn Sie die Skin einer Komponente bearbeiten, wird die Darstellung der Komponente standardmäßig für alle Instanzen im Dokument geändert. Möchten Sie für eine Komponente verschiedene Erscheinungsbilder erstellen, müssen Sie die zu ändernden Skins duplizieren, unterschiedlich benennen, bearbeiten und dann die gewünschten Stile festlegen, um die Skins anzuwenden. Weitere Informationen finden Sie unter „[Erstellen von Skins für einige Instanzen](#)“ auf Seite 109.

In diesem Kapitel wird beschrieben, wie Sie eine oder mehrere Skins für die einzelnen UI-Komponenten ändern. Wenn Sie eines dieser Verfahren verwenden, um eine oder mehrere Skins einer UI-Komponente zu ändern, gelten diese Änderungen für alle Instanzen im Dokument.

### Erstellen von Skins für einige Instanzen

Sie können eine Skin für einige Instanzen einer Komponente erstellen, indem Sie die nachstehenden allgemeinen Schritte ausführen:

- Wählen Sie die Skin im Ordner „Assets“ der Komponente im Bedienfeld „Bibliothek“ aus.
- Duplizieren Sie die Skin und weisen Sie ihr einen eindeutigen Klassennamen zu.
- Bearbeiten Sie die Skin, um das gewünschte Erscheinungsbild zu erstellen.
- Rufen Sie die `setStyle()`-Methode für die Komponenteinstanz auf, um dem Skin-Stil die neue Skin zuzuweisen.

Im folgenden Beispiel wird eine neue Skin „selectedDownSkin“ für eine von zwei Button-Instanzen erstellt.

- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie zwei Button-Komponenten aus dem Bedienfeld „Komponenten“ auf die Bühne und weisen Sie ihnen die Instanznamen **aButton** und **bButton** zu.
- 3 Öffnen Sie das Bedienfeld „Bibliothek“ und darin die Ordner „Component Assets“ und „ButtonSkins“.
- 4 Klicken Sie auf die Skin „selectedDownSkin“, um sie auszuwählen.
- 5 Klicken Sie mit der rechten Maustaste und wählen Sie „Duplizieren“ aus dem Kontextmenü.
- 6 Geben Sie im Dialogfeld „Symbol duplizieren“ einen eindeutigen Namen für die neue Skin ein, zum Beispiel **Button\_mySelectedDownSkin**. Klicken Sie dann auf „OK“.

- 7 Wählen Sie im Ordner „Bibliothek“ > „Component Assets“ > „ButtonSkins“ die Skin „Button\_mySelectedDownSkin“ und klicken Sie mit der rechten Maustaste darauf, um das Kontextmenü zu öffnen. Wählen Sie „Verknüpfung“, um das Dialogfeld „Verknüpfungseigenschaften“ zu öffnen.
- 8 Klicken Sie auf das Kontrollkästchen „Export für ActionScript“. Lassen Sie das Kontrollkästchen „In erstes Bild exportieren“ ausgewählt und stellen Sie sicher, dass der Klassenname eindeutig ist. Klicken Sie auf „OK“. Wenn Sie in einer Warnung darauf hingewiesen werden, dass keine Klassendefinition gefunden wurde, klicken Sie erneut auf „OK“, um eine neue zu erstellen.
- 9 Doppelklicken Sie im Bedienfeld „Bibliothek“ auf die Skin „Button\_mySelectedDownSkin“, um sie im Symbolbearbeitungsmodus zu öffnen.
- 10 Klicken Sie auf die blaue Füllung in der Mitte der Skin, bis die Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt wird. Klicken Sie auf die Farbauswahl und wählen Sie die Farbe #00CC00 für die Füllung der Skin aus.
- 11 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 12 Klicken Sie im Eigenschafteninspektor für jede Schaltfläche auf die Registerkarte „Parameter“ und setzen Sie den toggle-Parameter auf `true`.
- 13 Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Zeitleiste den folgenden Code hinzu:

```
bButton.setStyle("selectedDownSkin", Button_mySelectedDownSkin);
bButton.setStyle("downSkin", Button_mySelectedDownSkin);
```
- 14 Wählen Sie „Steuerung“ > „Film testen“.
- 15 Klicken Sie auf jede Schaltfläche. Sie sehen, dass für die Down-Skin (ausgewählt und nicht ausgewählt) des `bButton`-Objekts das neue Skinsymbol verwendet wird.

## Anpassen der Button-Komponente

Button-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder die entsprechenden Eigenschaften der Button-Klasse, zum Beispiel `height` und `width` sowie `scaleX` und `scaleY`.

Wenn die Größe der Schaltfläche geändert wird, wird die Größe des Symbols und der Beschriftung nicht automatisch angepasst. Die Begrenzungsbox einer Schaltfläche entspricht dem Rahmen der Schaltfläche. Sie markiert zugleich den Kollisionsbereich der Instanz. Wenn Sie eine Instanz vergrößern, vergrößert sich auch der Kollisionsbereich. Wenn der Platz innerhalb der Begrenzungsbox nicht für die Beschriftung ausreicht, wird diese unvollständig angezeigt.

Wenn die Schaltfläche über ein Symbol verfügt und dieses größer als die Schaltfläche ist, reicht das Symbol über den Rand der Schaltfläche hinaus.

## Verwenden von Stilen mit der Button-Komponente

Die Stile einer Schaltfläche definieren in der Regel Werte für die Skins, Symbole, Textformatierung und Auffüllung, die verwendet werden, wenn die Komponente in verschiedenen Zuständen gezeichnet wird.

Im folgenden Verfahren werden zwei Button-Komponenten auf der Bühne platziert. Dabei wird die Eigenschaft `emphasized` für beide Schaltflächen auf `true` gesetzt, wenn der Benutzer auf eine der Schaltflächen klickt. Außerdem wird der Stil `emphasizedSkin` für die zweite Schaltfläche auf `selectedOverSkin` eingestellt, wenn der Benutzer darauf klickt, damit die beiden Schaltflächen im selben Status mit unterschiedlichen Skins angezeigt werden.

- 1 Erstellen Sie eine Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie nacheinander zwei Button-Komponenten auf die Bühne und geben Sie ihnen die Instanznamen **aBtn** und **bBtn**. Geben Sie den Schaltflächen auf der Registerkarte „Parameter“ des Eigenschafteninspektors die Bezeichnungen „Button A“ und „Button B“.

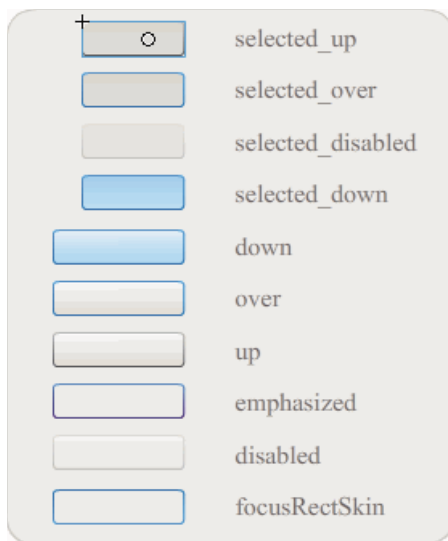
- 3 Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Zeitleiste den folgenden Code hinzu:

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

- 4 Wählen Sie „Steuerung“ > „Film testen“.
- 5 Klicken Sie auf eine der Schaltflächen, um die Auswirkung des Stils `emphasizedSkin` auf jede Schaltfläche zu sehen.

## Verwenden von Skins mit der Button-Komponente

Die Button-Komponente verwendet die folgenden Skins für ihre verschiedenen Zustände. Um eine oder mehrere Skins zu bearbeiten und so das Erscheinungsbild der Schaltfläche zu ändern, doppelklicken Sie auf der Bühne auf die Button-Instanz, um eine Palette mit ihren Skins zu öffnen, wie in der folgenden Abbildung gezeigt:

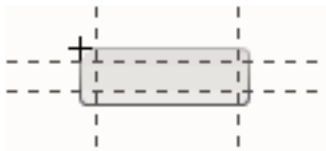


Button-Skins

Aktivierte Schaltflächen werden im Over-Status angezeigt, wenn sich der Mauszeiger darüber befindet. Beim Klicken auf die Schaltfläche erhält sie den Eingabefokus und wird im Down-Status angezeigt. Beim Loslassen der Maustaste wird die Schaltfläche wieder im Over-Status angezeigt. Wenn der Mauszeiger bei gedrückter Maustaste verschoben wird, sodass er sich nicht mehr über der Schaltfläche befindet, kehrt sie wieder in den Ausgangszustand zurück. Wenn der toggle-Parameter den Wert `true` hat, wird der gedrückte (pressed) Zustand mit der Skin „selectedDownSkin“ angezeigt, der Up-Status mit „selectedUpSkin“ und der Over-Status mit „selectedOverSkin“.

Deaktivierte Schaltflächen werden unabhängig von der Benutzerinteraktion immer im Disabled-Status angezeigt.

Um eine der Skins zu bearbeiten, doppelklicken Sie darauf, um sie im Symbolbearbeitungsmodus zu öffnen, wie in der folgenden Abbildung gezeigt:



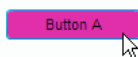
Schaltfläche im Symbolbearbeitungsmodus

Nun können Sie die Flash-Authoring-Tools verwenden, um die Skin wie gewünscht zu bearbeiten.

Mit dem folgenden Verfahren wird die Farbe der Skin „selected\_over“ der Schaltfläche geändert.

- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie eine Button-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne. Setzen Sie auf der Registerkarte „Parameter“ den toggle-Parameter auf `true`.
- 3 Doppelklicken Sie auf die Button-Instanz, um eine Palette ihrer Skins zu öffnen.
- 4 Doppelklicken Sie auf die Skin „selected\_over“, um sie im Symbolbearbeitungsmodus zu öffnen.
- 5 Stellen Sie die Vergrößerung auf 400 % ein, um das Symbol besser bearbeiten zu können.
- 6 Doppelklicken Sie auf den Hintergrund, bis seine Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt wird.
- 7 Wählen Sie in der Farbauswahl „Füllung“ die Farbe #CC0099 aus, um sie als Hintergrundfarbe für die Skin „selected\_over“ zu verwenden.
- 8 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 9 Wählen Sie „Steuerung“ > „Film testen“.
- 10 Klicken Sie auf die Schaltfläche, um sie in den ausgewählten Zustand zu versetzen.

Wenn Sie den Mauszeiger über die Schaltfläche bewegen, sollte die Skin „selected\_over“ wie in der folgenden Abbildung angezeigt werden.



Schaltfläche mit der Skin „selected\_over“ bei geänderter Farbe

## Anpassen der CheckBox-Komponente

CheckBox-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder geeignete Eigenschaften der CheckBox-Klasse. Beispielsweise können Sie die Größe eines Kontrollkästchens ändern, indem Sie die Eigenschaften `height` und `width` sowie `scaleX` und `scaleY` festlegen. Wenn die Größe eines Kontrollkästchens geändert wird, ändert sich nur die Größe der Begrenzungsbox; die Größe des Beschriftungstextes und des Kontrollkästchensymbols bleibt gleich.

Die Begrenzungsbox einer CheckBox-Instanz ist nicht sichtbar. Sie markiert zugleich den Kollisionsbereich der Instanz. Wenn Sie eine Instanz vergrößern, vergrößert sich auch der Kollisionsbereich. Wenn der Platz innerhalb der Begrenzungsbox nicht für die Beschriftung ausreicht, wird diese unvollständig angezeigt.

### Verwenden von Stilen mit der CheckBox-Komponente

Durch Zuweisen von Stileigenschaften können Sie die Gestaltung einer CheckBox-Instanz ändern. Mit dem folgenden Verfahren ändern Sie beispielsweise Größe und Farbe für die Beschriftung des Kontrollkästchens.

- 1 Ziehen Sie die CheckBox-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz **myCb**.
- 2 Klicken Sie im Eigenschafteninspektor auf die Registerkarte „Parameter“ und geben Sie den folgenden Wert für den `label`-Parameter ein: **Weniger als €500?**
- 3 Klicken Sie auf Bild 1 der Hauptzeitleiste und geben Sie im Bedienfeld „Aktionen“ den folgenden Code ein:

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

Weitere Informationen finden Sie unter „[Festlegen von Stilen](#)“ auf Seite 104. Unter „[Erstellen einer neuen Skin](#)“ auf Seite 109 und „[Verwenden von Skins mit der CheckBox-Komponente](#)“ auf Seite 113 wird beschrieben, wie Sie Stileigenschaften festlegen, um die Symbole und Skins einer Komponente zu ändern.

### Verwenden von Skins mit der CheckBox-Komponente

Die CheckBox-Komponente verfügt über die folgenden Skins, die Sie bearbeiten können, um das Erscheinungsbild zu ändern.



CheckBox-Skins

Mit diesem Beispiel werden die Kontur- und Hintergrundfarben der Komponente in den Zuständen `up` und `selectedUp` geändert. Mit ähnlichen Verfahren können auch die Skins für die anderen Zustände geändert werden.

- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie die CheckBox-Komponente auf die Bühne, wodurch sie auch, zusammen mit einem Ordner mit ihren Elementen, in der Bibliothek platziert wird.
- 3 Doppelklicken Sie auf die CheckBox-Komponente auf der Bühne, um ein Bedienfeld mit ihren Skinsymbolen zu öffnen.
- 4 Doppelklicken Sie auf das Symbol „selected\_up“, um es im Symbolbearbeitungsmodus zu öffnen.
- 5 Stellen Sie die Vergrößerung auf 800 % ein, um das Symbol besser bearbeiten zu können.
- 6 Klicken Sie auf den Rand des Kontrollkästchens, um ihn auszuwählen. Verwenden Sie die Farbauswahl „Füllung“ im Eigenschafteninspektor, um die Farbe #0033FF auszuwählen und für den Rand zu verwenden.
- 7 Doppelklicken Sie auf den Hintergrund der CheckBox-Komponente, um ihn auszuwählen. Wählen Sie in der Farbauswahl „Füllung“ die Farbe #00CCFF für den Hintergrund aus.
- 8 Wiederholen Sie die Schritte 4 bis 8 für die up-Skin des Kontrollkästchens.
- 9 Wählen Sie „Steuerung“ > „Film testen“.

## Anpassen der ColorPicker-Komponente

Die Größe einer ColorPicker-Komponente kann nur über ihre Stile geändert werden: `swatchWidth`, `swatchHeight`, `backgroundPadding`, `textFieldWidth` und `textFieldHeight`. Wenn Sie die Größe einer ColorPicker-Komponente mit dem Werkzeug „Transformieren“ oder in ActionScript mit der `setSize()`-Methode oder den Eigenschaften `width`, `height`, `scaleX` oder `scaleY` ändern, werden diese Werte bei der Erstellung der SWF-Datei ignoriert und die ColorPicker-Komponente wird in ihrer Standardgröße angezeigt. Die Größe des Palettenhintergrunds wird an die Anzahl der Spalten angepasst, die mit `setStyle()` für den `columnCount`-Stil festgelegt wurde. Die Standardanzahl beträgt 18. Sie können benutzerdefinierte Farben auf den Wert 1024 einstellen. Dadurch wird die Größe der Palette vertikal so geändert, dass sie der Anzahl der Farbfelder entspricht.

## Verwenden von Stilen mit der ColorPicker-Komponente

Sie können mehrere Stile festlegen, um die Darstellung der ColorPicker-Komponente zu ändern. Mit dem folgenden Verfahren ändern Sie beispielsweise die Anzahl der Spalten (`columnCount`) in der ColorPicker-Komponente in 12. Außerdem ändern Sie die Höhe (`swatchHeight`) und Breite (`swatchWidth`) der Farbfelder sowie die Auffüllung für das Textfeld (`textPadding`) und den Hintergrund (`backgroundPadding`).

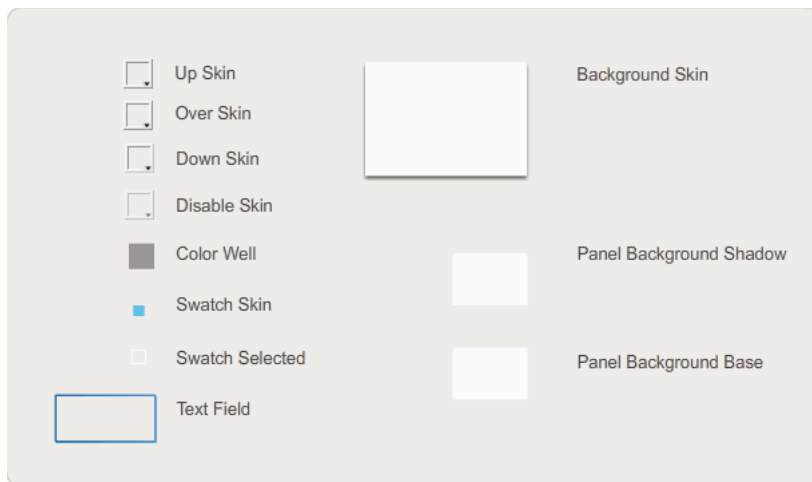
- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie die ColorPicker-Komponente auf die Bühne und geben Sie ihr den Instanznamen **aCp**.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein:

```
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 Wählen Sie „Steuerung“ > „Film testen“.
- 5 Klicken Sie auf die ColorPicker-Komponente, um sie zu öffnen und zu sehen, wie diese Einstellungen das Erscheinungsbild geändert haben.

## Verwenden von Skins mit der ColorPicker-Komponente

Die ColorPicker-Komponente verwendet die folgenden Skins, um ihre verschiedenen Zustände darzustellen.



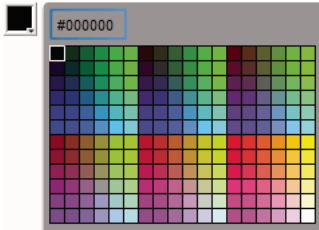
ColorPicker-Skins

Sie können die Farbe der Background-Skin ändern, um die Farbe des Palettenhintergrunds anzupassen.

- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie die ColorPicker-Komponente auf die Bühne.
- 3 Doppelklicken Sie auf die Komponente, um die Palette ihrer Skins zu öffnen.
- 4 Doppelklicken Sie auf die Background-Skin, bis sie ausgewählt ist und die Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt wird.
- 5 Wählen Sie in der Farbauswahl „Füllung“ die Farbe #999999 für die Background-Skin aus.

- 6 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 7 Wählen Sie „Steuerung“ > „Film testen“.

Wenn Sie auf die ColorPicker-Komponente klicken, sollte der Hintergrund der Palette grau angezeigt werden, wie in der folgenden Abbildung.



Farbauswahl mit dunkelgrauer Background-Skin

## Anpassen der ComboBox-Komponente

ComboBox-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder die entsprechenden Eigenschaften der ComboBox-Klasse, zum Beispiel `height` und `width`, `scaleX` und `scaleY`.

Das Kombinationsfeld wird an die angegebene Breite und Höhe angepasst. Die Größe der Liste wird an die Breite der Komponente angepasst, es sei denn, die `dropdownWidth`-Eigenschaft wurde festgelegt.

Falls der Text nicht in das Kombinationsfeld passt, wird er unvollständig angezeigt. Sie müssen die Größe der ComboBox-Komponente ändern und die `dropdownWidth`-Eigenschaft festlegen, damit der Text in das Kombinationsfeld passt.

## Verwenden von Stilen mit der ComboBox-Komponente

Durch Zuweisen von Stileigenschaften können Sie die Gestaltung einer ComboBox-Komponente ändern. Die Stile geben Werte für die Skins, den `CellRenderer`, die Auffüllung und die Schaltflächenbreite der Komponente an. Im folgenden Beispiel werden die Stile `buttonWidth` und `textPadding` festgelegt. Der Stil `buttonWidth` legt die Breite des Hit-Bereichs der Schaltfläche fest und ist wirksam, wenn die ComboBox-Komponente bearbeitbar ist und Sie mit der Schaltfläche nur die Dropdownliste öffnen können. Der `textPadding`-Stil bestimmt den Abstand zwischen dem äußeren Rand des Textfeldes und dem darin enthaltenen Text. Er eignet sich zum vertikalen Zentrieren des Textes im Textfeld, wenn Sie die Höhe des Kombinationsfeldes vergrößern. Andernfalls wird der Text möglicherweise ganz oben im Textfeld angezeigt.

- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie die ComboBox-Komponente auf die Bühne und geben Sie ihr den Instanznamen `aCb`.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein:



```
import fl.data.DataProvider;

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

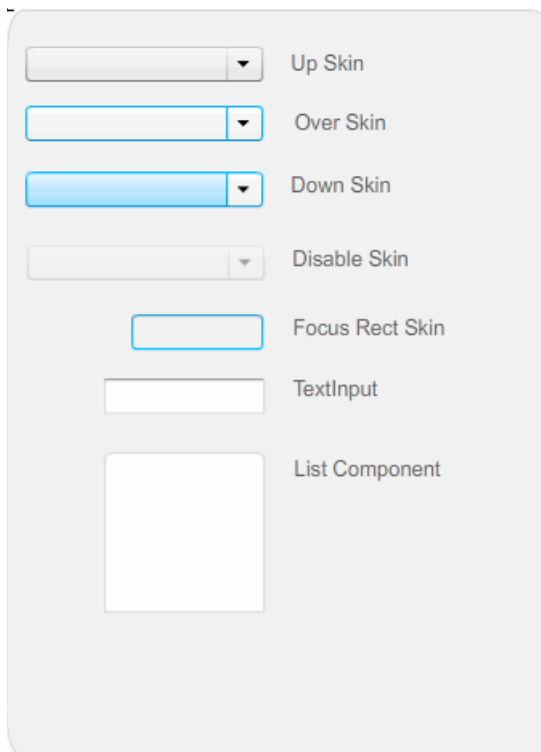
var items:Array = [
 {label:"San Francisco", data:"601 Townsend St."},
 {label:"San Jose", data:"345 Park Ave."},
 {label:"San Diego", data:"10590 West Ocean Air Drive, Suite 100"},
 {label:"Santa Rosa", data:"2235 Mercury Way, Suite 105"},
 {label:"San Luis Obispo", data:"3220 South Higuera Street,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

#### 4 Wählen Sie „Steuerung“ > „Film testen“.

Der Bereich der Schaltfläche, auf den Sie klicken können, um die Dropdownliste zu öffnen, ist nur ein schmaler Rand an der rechten Seite. Beachten Sie auch, dass der Text vertikal im Textfeld zentriert ist. Sie können das Beispiel ohne die beiden `setStyle()`-Anweisungen ausführen, um deren Auswirkungen zu verdeutlichen.

## Verwenden von Skins mit der ComboBox-Komponente

Die ComboBox-Komponente verwendet die folgenden Skins, um ihre verschiedenen Zustände darzustellen:



ComboBox-Skins

Sie können die Farbe der Up-Skin und damit die Farbe der Komponente im inaktiven Status auf der Bühne ändern.

#### 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).

- 2 Ziehen Sie die ComboBox-Komponente auf die Bühne.
- 3 Doppelklicken Sie auf die Komponente, um die Palette ihrer Skins zu öffnen.
- 4 Doppelklicken Sie auf die Up-Skin, bis sie ausgewählt und zum Bearbeiten bereit ist.
- 5 Stellen Sie die Vergrößerung auf 400 % ein.
- 6 Klicken Sie auf den mittleren Bereich der Skin, bis die Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt wird.
- 7 Wählen Sie in der Farbauswahl „Füllung“ die Farbe #33FF99 für die Up-Skin aus.
- 8 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 9 Wählen Sie „Steuerung“ > „Film testen“.

Die ComboBox-Komponente sollte wie in der folgenden Abbildung auf der Bühne angezeigt werden.



*Kombinationsfeld mit benutzerdefinierter Farbe für die Background-Skin*

## Anpassen der DataGrid-Komponente

Sie können eine DataGrid-Komponente beim Authoring oder zur Laufzeit horizontal und vertikal transformieren. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder entsprechende Eigenschaften, zum Beispiel `width`, `height`, `scaleX` und `scaleY`. Ist keine horizontale Bildlaufleiste vorhanden, werden die Spaltenbreiten proportional angepasst. Falls eine Größenanpassung der Spalte (und damit der Zellen) eintritt, wird der Text in den Zellen möglicherweise abgeschnitten.

## Verwenden von Stilen mit der DataGrid-Komponente

Sie können Stileigenschaften festlegen, um die Darstellung der DataGrid-Komponente zu ändern. Die DataGrid-Komponente übernimmt die Stile von der List-Komponente. (Siehe „[Verwenden von Stilen mit der List-Komponente](#)“ auf Seite 124.)

### Festlegen von Stilen für einzelne Spalten

Ein DataGrid-Objekt kann mehrere Spalten enthalten und Sie können für jede Spalte verschiedene `CellRenderer` festlegen. Jede Spalte einer DataGrid-Komponente wird von einem `DataGridColumn`-Objekt repräsentiert. Die `DataGridColumn`-Klasse enthält die Eigenschaft `cellRenderer`, für die Sie einen `CellRenderer` für die Spalte definieren können.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die DataGrid-Komponente in das Bedienfeld „Bibliothek“.
- 3 Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Zeitleiste den folgenden Code hinzu: Dieser Code erstellt eine DataGrid-Komponente mit einem langen Textstring in der dritten Spalte. Am Ende des Beispiels wird die `cellRenderer`-Eigenschaft der Spalte auf den Namen eines `CellRenderers` gesetzt, der eine mehrzeilige Zelle wiedergibt.

```
/* 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 Speichern Sie die FLA-Datei unter dem Namen „MultiLineGrid.fla“.
- 5 Erstellen Sie eine neue ActionScript-Datei.
- 6 Kopieren Sie den folgenden ActionScript-Code in das Skriptfenster:

```
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 Speichern Sie die ActionScript-Datei unter dem Namen „MultiLineCell.as“ in demselben Ordner, in dem Sie auch „MultiLineGrid fla“ gespeichert haben.
- 8 Kehren Sie zur Anwendung „MultiLineGrid fla“ zurück und wählen Sie „Steuerung“ > „Film testen“.

Die DataGrid-Komponente sollte nun folgendermaßen aussehen:

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

Datenraster für die Anwendung „MultiLineGrid fla“

## Festlegen von Kopfzeilenstilen

Sie können den Textstil für eine Kopfzeile mit dem Stil `headerTextFormat` festlegen. Im folgenden Beispiel wird das `TextFormat`-Objekt verwendet, um den Stil `headerTextFormat` auf Arial, kursiv, rot und die Schriftgröße 14 einzustellen.

- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie die DataGrid-Komponente auf die Bühne und geben Sie ihr den Instanznamen **aDg**.
- 3 Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein:

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

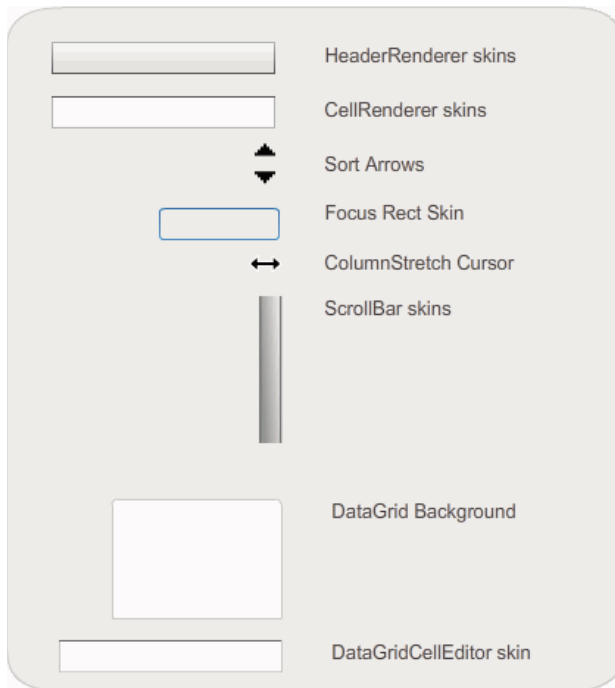
var myDP:Array = new Array();
myDP = [{FirstName:"Winston", LastName:"Elstad"},
 {FirstName:"Ric", LastName:"Dietrich"},
 {FirstName:"Ewing", LastName:"Canepa"},
 {FirstName:"Kevin", LastName:"Wade"},
 {FirstName:"Kimberly", LastName:"Dietrich"},
 {FirstName:"AJ", LastName:"Bilow"},
 {FirstName:"Chuck", 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 Wählen Sie „Steuerung“ > „Film testen“ aus, um die Anwendung auszuführen.

## Verwenden von Skins mit der DataGrid-Komponente

Die DataGrid-Komponente verwendet die folgenden Skins, um ihre verschiedenen Zustände darzustellen:



*DataGrid-Skins*

Die CellRenderer-Skin wird für die Tabellenzellen der DataGrid-Komponente verwendet, während die HeaderRenderer-Skin für die Kopfzeile verwendet wird. Im folgenden Verfahren wird die Hintergrundfarbe der Kopfzeile geändert. Mit demselben Verfahren könnten Sie aber auch die Hintergrundfarbe der DataGrid-Tabellenzellen durch Bearbeiten der CellRenderer-Skin ändern.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die DataGrid-Komponente auf die Bühne und geben Sie ihr den Instanznamen **aDg**.
- 3 Doppelklicken Sie auf die Komponente, um die Palette ihrer Skins zu öffnen.
- 4 Stellen Sie die Vergrößerung auf 400 % ein, um die Symbole besser bearbeiten zu können.
- 5 Doppelklicken Sie auf die HeaderRenderer-Skin, um eine Palette der Skins zu öffnen.
- 6 Doppelklicken Sie auf die Up\_Skin, um sie im Symbolbearbeitungsmodus zu öffnen. Klicken Sie auf den Hintergrund, bis er ausgewählt ist und die Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt wird.
- 7 Wählen Sie in der Farbauswahl „Füllung“ die Farbe #00CC00 aus, um sie als Hintergrundfarbe für die HeaderRenderer-Skin „Up\_Skin“ zu verwenden.
- 8 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 9 Fügen Sie dem Bedienfeld „Aktionen“ in Bild 1 der Zeitleiste den folgenden Code hinzu, um das Datenraster mit Daten zu versehen:

```
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 Wählen Sie „Steuerung“ > „Film testen“, um die Anwendung zu testen.

Die DataGrid-Komponente wird wie in der folgenden Abbildung angezeigt. Der Hintergrund der Kopfzeile ist grün.

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

Datenraster mit benutzerdefiniertem Hintergrund für die Kopfzeile

## Anpassen der Label-Komponente

Label-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Außerdem können Sie den Authoring-Parameter `autoSize` einstellen. Dadurch wird zwar nicht die Begrenzungsbox in der Live-Vorschau geändert, aber die Größe der Bezeichnung wird angepasst. Die Größe der Bezeichnung ändert sich in Abhängigkeit vom `wordwrap`-Parameter. Wenn der Parameter `true` lautet, wird die Bezeichnung vertikal an den Text angepasst. Beim Wert `false` wird die Größe der Bezeichnung horizontal angepasst. Zur Laufzeit verwenden Sie die Methode `setSize()`. Weitere Informationen finden Sie in den Abschnitten über die `Label.setSize()`-Methode und die `Label.autoSize`-Eigenschaft im *ActionScript 3.0 Referenzhandbuch für Flash Professional*. Lesen Sie auch das Thema „Erstellen einer Anwendung mit der Label-Komponente“ auf Seite 66.

## Verwenden von Stilen mit der Label-Komponente

Durch Zuweisen von Stileigenschaften können Sie die Gestaltung einer Label-Instanz ändern. Ein Stil kann immer nur für den gesamten Text innerhalb einer Instanz der Label-Komponente zugewiesen werden. Die Label-Komponente verfügt über den `textFormat`-Stil, der dieselben Attribute wie das `TextFormat`-Objekt hat und Ihnen ermöglicht, dieselben Eigenschaften für den Inhalt von `Label.text` festzulegen wie auch für eine reguläre Flash `TextField`-Komponente. Im folgenden Beispiel wird die Textfarbe der Bezeichnung auf Rot festgelegt.

- 1 Ziehen Sie die Label-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz `a_label`.
- 2 Klicken Sie auf die Registerkarte „Parameter“ und ersetzen Sie den Wert der Text-Eigenschaft durch den folgenden Text:

**Color me red**

- 3 Wählen Sie in der Hauptzeitleiste das erste Bild aus, öffnen Sie das Bedienfeld „Aktionen“ und geben Sie den folgenden Code ein:

```
/* 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 Wählen Sie „Steuerung“ > „Film testen“.

Weitere Informationen zu Label-Stilen finden Sie im Abschnitt zur Label-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Skins und die Label-Komponente

Die Label-Komponente enthält keine grafischen Elemente, die mit Skins versehen werden können.

## Anpassen der List-Komponente

List-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` und die entsprechenden Eigenschaften der List-Klasse, zum Beispiel `height` und `width`, `scaleX` und `scaleY`.

Wenn die Größe einer Liste geändert wird, schrumpfen die Listenzeilen horizontal, wobei der enthaltene Text gegebenenfalls abgeschnitten wird. Vertikal werden in der Liste nach Bedarf Zeilen hinzugefügt oder entfernt. Bildlaufleisten werden automatisch bedarfsgemäß positioniert.

## Verwenden von Stilen mit der List-Komponente

Sie können Stileigenschaften festlegen, um das Erscheinungsbild der List-Komponente zu ändern. Die Stile legen die Werte für die Skins und Auffüllung beim Zeichnen der Komponente fest.

Die unterschiedlichen Skin-Stile ermöglichen es Ihnen, verschiedene Klassen für die Skin anzugeben. Weitere Informationen zur Verwendung der Skin-Stile finden Sie unter „[Überblick über Skins](#)“ auf Seite 107.



Im folgenden Verfahren wird der Wert des `contentPadding`-Stils für die List-Komponente festgelegt. Beachten Sie, dass der Wert dieser Einstellung von der Größe der Liste subtrahiert wird, um die Auffüllung um den Inhalt zu erzielen. Deshalb müssen Sie die Liste möglicherweise vergrößern, damit der Text in der Liste nicht abgeschnitten wird.

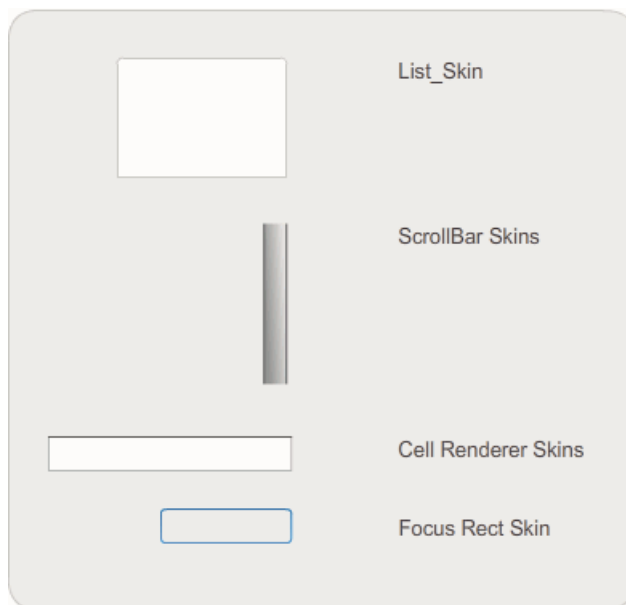
- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie die List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz **aList**.
- 3 Wählen Sie in der Hauptzeitleiste Bild 1 aus, öffnen Sie das Bedienfeld „Aktionen“ und geben Sie den folgenden Code ein, mit dem der Stil `contentPadding` festgelegt und die Liste mit Daten versehen wird:

```
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 Wählen Sie „Steuerung“ > „Film testen“.

## Verwenden von Skins mit der List-Komponente

Die List-Komponente verwendet die folgenden Skins, um ihre verschiedenen Zustände darzustellen:

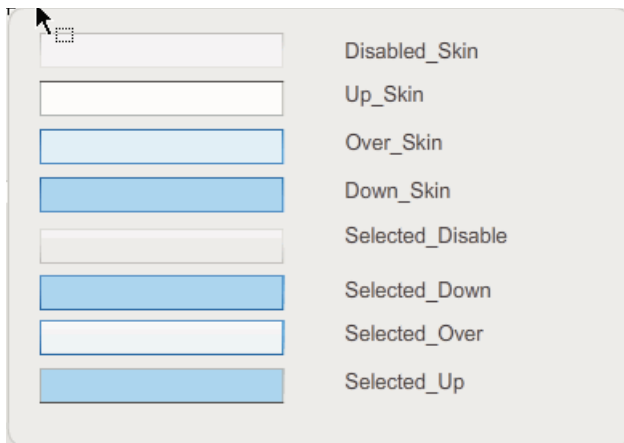


List-Skins

Weitere Informationen zum Skinning der ScrollBar-Komponente finden Sie unter „[Anpassen der UI ScrollBar-Komponente](#)“ auf Seite 141. Einzelheiten zum Skinning der Focus Rect-Skin finden Sie unter „[Anpassen der TextArea-Komponente](#)“ auf Seite 135.

**Hinweis:** Wenn Sie die ScrollBar-Skin in einer Komponente ändern, wird sie auch für alle anderen Komponenten, die die ScrollBar-Komponente verwenden, geändert.

Doppelklicken Sie auf die CellRendererer-Skin, um eine zweite Palette mit den Skins für die verschiedenen Zustände einer Listenzelle zu öffnen.



CellRenderer-Skins für die Liste

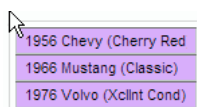
Sie können die Darstellung der Listenzellen ändern, indem Sie diese Skins bearbeiten. In den folgenden Schritten wird die Farbe der Up-Skin geändert, um das Aussehen der Liste im normalen, nicht aktivierten Zustand zu ändern.

- 1 Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
- 2 Ziehen Sie die List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz **aList**.
- 3 Doppelklicken Sie auf die Liste, um die Palette ihrer Skins zu öffnen.
- 4 Doppelklicken Sie auf die CellRenderer-Skin, um eine Palette mit den CellRenderer-Skins zu öffnen.
- 5 Doppelklicken Sie auf die Up\_Skin, um sie zum Bearbeiten zu öffnen.
- 6 Klicken Sie auf den Füllbereich der Skin, um ihn auszuwählen. Im Eigenschafteninspektor wird die Farbauswahl „Füllung“ mit der aktuellen Füllfarbe der Skin angezeigt.
- 7 Wählen Sie in der Farbauswahl „Füllung“ die Farbe #CC66FF aus, um sie als Füllfarbe für die Skin „Up\_Skin“ zu verwenden.
- 8 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 9 Fügen Sie dem Bedienfeld „Aktionen“ in Bild 1 der Zeitleiste den folgenden Code hinzu, um die Liste mit Daten zu versehen:

```
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 Wählen Sie „Steuerung“ > „Film testen“.

Die Liste sollte jetzt wie in der folgenden Abbildung angezeigt werden:



Listenzellen mit benutzerdefinierter Farbe für Up\_Skin

Der Rahmen, der sich aus dem `contentPadding`-Stil ergibt.

## Anpassen der NumericStepper-Komponente

NumericStepper-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder die entsprechenden Eigenschaften und Methoden der NumericStepper-Klasse, zum Beispiel `width`, `height`, `scaleX` und `scaleY`.

Wenn Sie die Größe der NumericStepper-Komponente ändern, wird dadurch die Breite der Pfeilschaltflächen nach oben und unten nicht geändert. Wenn der Stepper so stark vergrößert wird, dass er die Standardhöhe überschreitet, werden die Pfeilschaltflächen standardmäßig oben und unten an der Komponente fixiert. Andernfalls wird durch die Skalierung im neunteiligen Segmentraster (Scale-9) festgelegt, wie die Schaltflächen gezeichnet werden. Die Pfeilschaltflächen werden immer rechts vom Textfeld angezeigt.

### Stile und die NumericStepper-Komponente

Sie können die Stileigenschaften der NumericStepper-Komponente festlegen, um ihr Erscheinungsbild zu ändern. Die Stile legen die Werte für die Skins, Auffüllung und Textformatierung beim Zeichnen der Komponente fest. Mit dem Stil `textFormat` können Sie Größe und Darstellung des NumericStepper-Wertes ändern. Die unterschiedlichen Skin-Stile ermöglichen es Ihnen, verschiedene Klassen für die Skin anzugeben. Weitere Informationen zur Verwendung der Skin-Stile finden Sie unter „[Überblick über Skins](#)“ auf Seite 107.

In den folgenden Schritten wird der Stil `textFormat` verwendet, um das Aussehen des in der NumericStepper-Komponente angezeigten Werts zu ändern.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die NumericStepper-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz `myNs`.
- 3 Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Hauptzeitleiste den folgenden Code hinzu:

```
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);
```

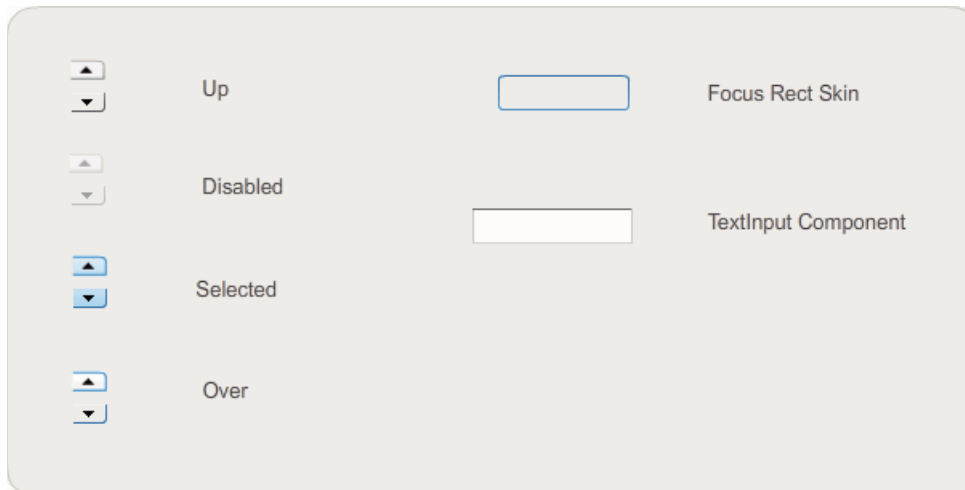
- 4 Wählen Sie „Steuerung“ > „Film testen“.

### Skins und die NumericStepper-Komponente

Die NumericStepper-Komponente verfügt über Skins, die den Up- und Down-Status sowie den ausgewählten und deaktivierten Zustand ihrer Schaltflächen darstellen.

Wenn ein Stepper aktiviert ist, werden die Schaltflächen nach oben und unten im Over-Status angezeigt, wenn der Zeiger über sie hinweg bewegt wird. Beim Klicken auf die Schaltflächen werden sie im Down-Status angezeigt. Wenn die Maustaste losgelassen wird, kehren die Schaltflächen zum Over-Status zurück. Wenn der Zeiger von den Schaltflächen weg bewegt wird, während die Maustaste noch gedrückt ist, kehren die Schaltflächen zu ihrem ursprünglichen Zustand zurück.

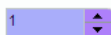
Wenn ein Stepper deaktiviert ist, wird unabhängig von der Benutzerinteraktion der deaktivierte Zustand angezeigt.  
Eine NumericStepper-Komponente verfügt über die folgenden Skins:



*NumericStepper-Skins*

- 1 Erstellen Sie eine neue FLA-Datei.
- 2 Ziehen Sie die NumericStepper-Komponente auf die Bühne.
- 3 Stellen Sie die Vergrößerung auf 400 % ein, um das Bild besser bearbeiten zu können.
- 4 Doppelklicken Sie auf den Hintergrund der TextInput-Skin im Fenster „Skins“, bis Sie zur Gruppenebene gelangen und die Hintergrundfarbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt wird.
- 5 Wählen Sie in der Farbauswahl „Füllung“ im Eigenschafteninspektor die Farbe #9999FF aus, um sie als Hintergrundfarbe für die TextInput-Skin zu verwenden.
- 6 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 7 Doppelklicken Sie erneut auf die NumericStepper-Komponente, um das Bedienfeld mit den Skins zu öffnen.
- 8 Doppelklicken Sie auf den Hintergrund der Schaltfläche mit dem Pfeil nach oben in der Up-Gruppe, bis der Hintergrund ausgewählt ist und die entsprechende Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt wird.
- 9 Wählen Sie die Farbe #9966FF aus, um sie als Hintergrundfarbe für die Schaltfläche mit dem Pfeil nach oben zu verwenden.
- 10 Wiederholen Sie die Schritte 8 und 9 für den Pfeil nach unten in der Up-Gruppe.
- 11 Wählen Sie „Steuerung“ > „Film testen“.

Die NumericStepper-Instanz sollte jetzt wie in der folgenden Abbildung aussehen:



## Anpassen der ProgressBar-Komponente

ProgressBar-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder die entsprechenden Eigenschaften der ProgressBar-Klasse, zum Beispiel `height` und `width`, `scaleX` und `scaleY`.

Eine ProgressBar-Komponente hat drei Skins: je eine für Fortschritt, Leiste und den unbestimmten Zustand. Für die Skalierung der Elemente wird Scale-9 verwendet.

### Stile und die ProgressBar-Komponente

Durch Zuweisen von Stileigenschaften können Sie die Gestaltung einer ProgressBar-Instanz ändern. Die ProgressBar-Stile legen die Werte für die Skins und Auffüllung beim Zeichnen der Komponente fest. Im folgenden Beispiel wird gezeigt, wie Sie eine ProgressBar-Instanz vergrößern und ihren `barPadding`-Stil festlegen.

- 1 Erstellen Sie eine neue FLA-Datei.
- 2 Ziehen Sie die ProgressBar-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz **myPb**.
- 3 Klicken Sie auf Bild 1 der Hauptzeitleiste und geben Sie im Bedienfeld „Aktionen“ den folgenden Code ein:

```
myPb.width = 300;
myPb.height = 30;

myPb.setStyle("barPadding", 3);
```

- 4 Wählen Sie „Steuerung“ > „Film testen“.

Weitere Informationen zum Festlegen von Skin-Stilen finden Sie unter „[Überblick über Skins](#)“ auf Seite 107.

### Skins und die ProgressBar-Komponente

Die Skins der ProgressBar-Komponente zeigen den Fortschritt, die abgeschlossene Leiste und einen unbestimmten Zustand, wie in der folgenden Abbildung dargestellt.



ProgressBar-Skins

Die Leiste wird über der Fortschritt-Skin platziert, wobei die Position mithilfe der `barPadding`-Eigenschaft bestimmt wird. Für die Skalierung der Elemente wird Scale-9 verwendet.

Die unbestimmte Leiste wird verwendet, wenn die Eigenschaft `indeterminate` der ProgressBar-Instanz auf `true` gesetzt wurde. Die Größe der Skin wird vertikal und horizontal geändert, um sie an die Größe der ProgressBar-Komponente anzupassen.

Sie können diese Skins bearbeiten, um die Darstellung der Fortschrittsleiste zu ändern. Im folgenden Verfahren wird beispielsweise die Farbe der Leiste für den unbestimmten Zustand geändert.

- 1 Erstellen Sie eine neue FLA-Datei.
- 2 Ziehen Sie eine `ProgressBar`-Komponente auf die Bühne und doppelklicken Sie darauf, um das Fenster mit den Skinsymbolen zu öffnen.
- 3 Doppelklicken Sie auf die Skin der unbestimmten Leiste.
- 4 Stellen Sie die Vergrößerung auf 400 % ein, um das Symbol besser bearbeiten zu können.
- 5 Doppelklicken Sie auf einen der diagonalen Streifen, halten Sie die Umschalttaste gedrückt und klicken Sie auf die anderen Streifen. Die aktuelle Farbe wird in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt.
- 6 Klicken Sie im Eigenschafteninspektor auf die Farbauswahl „Füllung“, um sie öffnen, und wählen Sie die Farbe #00CC00 aus, um sie für die ausgewählten diagonalen Streifen zu verwenden.
- 7 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 8 Wählen Sie „Steuerung“ > „Film testen“.

Die `ProgressBar`-Komponente sollte jetzt wie in der folgenden Abbildung aussehen.



## Anpassen der RadioButton-Komponente

`RadioButton`-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()`.

Die Begrenzungsbox der `RadioButton`-Komponente ist unsichtbar. Sie definiert zugleich den Kollisionsbereich der Komponente. Wenn Sie die Komponente vergrößern, wird auch der Kollisionsbereich vergrößert.

Wenn die Begrenzungsbox der Komponente zu klein ist, um die Komponentenbezeichnung aufzunehmen, wird die Bezeichnung abgeschnitten.

## Verwenden von Stilen mit der RadioButton-Komponente

Sie können Stileigenschaften festlegen, um das Erscheinungsbild eines Optionsfelds zu ändern. Die `RadioButton`-Stileigenschaften legen die Werte für die Skins, Symbole, Textformatierung und Auffüllung beim Zeichnen der Komponente fest. Die `RadioButton`-Stile definieren Werte für die Skins und Auffüllung des Layouts beim Zeichnen der Komponente.

Im folgenden Beispiel wird der `textFormat`-Stil einer `CheckBox`-Komponente abgerufen und auf eine `RadioButton`-Komponente angewendet, damit beide Bezeichnungen denselben Stil aufweisen.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine `CheckBox`-Komponente auf die Bühne und geben Sie ihr im Eigenschafteninspektor den Instanznamen **myCh**.

3 Ziehen Sie eine RadioButton-Komponente auf die Bühne und geben Sie ihr im Eigenschafteninspektor den Instanznamen **myRb**.

4 Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Zeitleiste den folgenden Code hinzu:

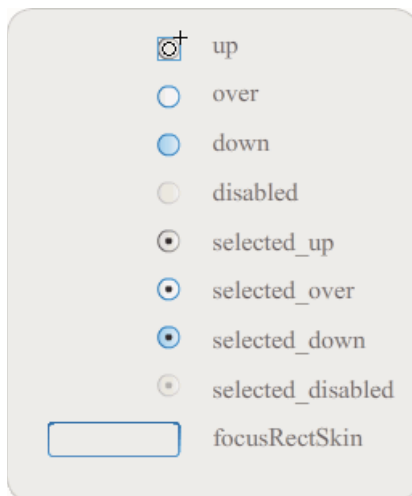
```
var tf:TextFormat = new TextFormat();
tf.color = 0x00FF00;
tf.font = "Georgia";
tf.size = 18;
myCh.setStyle("textFormat", tf);
myRb.setStyle("textFormat", myCh.getStyle("textFormat"));
```

Dieser Code legt den Stil `textFormat` für die CheckBox-Komponente fest und wendet ihn dann auf die RadioButton-Komponente an, indem die Methode `getStyle()` für die CheckBox-Komponente aufgerufen wird.

5 Wählen Sie „Steuerung“ > „Film testen“.

## Skins und die RadioButton-Komponente

Die RadioButton-Komponente verfügt über die folgenden Skins, die Sie bearbeiten können, um das Erscheinungsbild zu ändern.



*RadioButton-Skins*

Wenn ein Optionsfeld aktiviert, aber nicht ausgewählt ist, zeigt es die Over-Skin an, wenn ein Benutzer mit der Maus darauf zeigt. Wenn der Benutzer auf das Optionsfeld klickt, erhält es den Fokus und wird mit der Skin „selected\_down“ angezeigt. Wenn der Benutzer die Maustaste loslässt, zeigt das Optionsfeld die Skin „selected\_up“ an. Wenn ein Benutzer den Zeiger aus dem Hit-Bereich des Optionsfelds bewegt, während die Maustaste noch gedrückt ist, wird das Optionsfeld mit der Up-Skin angezeigt.

Deaktivierte Optionsfelder werden unabhängig von der Benutzerinteraktion immer im Disabled-Status angezeigt.

Im folgenden Beispiel wird die Skin „selected\_up“, die den ausgewählten Zustand darstellt, ersetzt.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine RadioButton-Komponente auf die Bühne und doppelklicken Sie darauf, um die Palette mit den Skins zu öffnen.
- 3 Stellen Sie die Vergrößerung auf 800 % ein, um das Symbol besser bearbeiten zu können.

- 4 Doppelklicken Sie auf die Skin „selected\_up“, um sie auszuwählen, und drücken Sie die Entf-Taste, um sie zu löschen.
- 5 Wählen Sie im Bedienfeld „Werkzeuge“ das Rechteckwerkzeug aus.
- 6 Legen Sie im Eigenschafteninspektor die Strichfarbe auf Rot (#FF0000) und die Füllfarbe auf Schwarz (#000000) fest.
- 7 Klicken Sie und ziehen Sie ab dem Fadenkreuz, das die Einfügemarke (auch *Ursprung* oder *Nullpunkt* genannt) des Symbols kennzeichnet, mit dem Zeiger ein Rechteck auf.
- 8 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 9 Wählen Sie „Steuerung“ > „Film testen“.
- 10 Klicken Sie auf die RadioButton-Komponente, um sie auszuwählen.

Die RadioButton-Komponente im ausgewählten Zustand sollte ungefähr der folgenden Abbildung entsprechen.



## Anpassen der ScrollPane-Komponente

ScrollPane-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder die entsprechenden Eigenschaften und Methoden der ScrollPane-Klasse, zum Beispiel `height`, `width`, `scaleX` und `scaleY`.

Die ScrollPane-Komponente hat die folgenden grafischen Merkmale:

- Die Einfügemarke (auch *Ursprung* oder *Nullpunkt* genannt) des Inhalts befindet sich oben links im Bereich.
- Bei deaktivierter horizontaler Bildlaufleiste wird die vertikale Bildlaufleiste von oben nach unten an der rechten Seite des Bildlauffensters angezeigt. Bei deaktivierter vertikaler Bildlaufleiste wird die horizontale Bildlaufleiste von links nach rechts am unteren Rand des Bildlauffensters angezeigt. Es lassen sich auch beide Bildlaufleisten deaktivieren.
- Wenn das Bildlauffenster zu klein ist, werden die Inhalte möglicherweise nicht korrekt angezeigt.
- Wenn die Größe des Bildlauffensters geändert wird, werden die Spur der Bildlaufleiste und das Bildlauffeld entsprechend vergrößert oder verkleinert und die Größe der Kollisionsbereiche wird entsprechend angepasst. Die Schaltflächen behalten ihre ursprüngliche Größe bei.

## Verwenden von Stilen mit der ScrollPane-Komponente

Die Stileigenschaften der ScrollPane-Komponente legen die Werte für Skins und Auffüllung des Layouts beim Zeichnen der Komponente fest. Die unterschiedlichen Skin-Stile ermöglichen es Ihnen, verschiedene Klassen für die Skins der Komponente anzugeben. Weitere Informationen zur Verwendung der Skin-Stile finden Sie unter „[Überblick über Skins](#)“ auf Seite 107.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine ScrollPane-Komponente auf die Bühne und geben Sie ihr den Instanznamen **mySp**.



- 3 Klicken Sie im Eigenschafteninspektor auf die Registerkarte „Parameter“ und geben Sie den folgenden Wert für den `source`-Parameter ein: <http://www.helpexamples.com/flash/images/image1.jpg>.
- 4 Klicken Sie auf Bild 1 der Hauptzeitleiste und geben Sie im Bedienfeld „Aktionen“ den folgenden Code ein:  

```
mySp.setStyle("contentPadding", 5);
```

Die Auffüllung wird zwischen dem Rahmen und dem Inhalt der Komponente außen an den Bildlaufleisten angewendet.
- 5 Wählen Sie „Steuerung“ > „Film testen“.

## Skins und die ScrollPane-Komponente

Die ScrollPane-Komponente verwendet einen Rahmen und Bildlaufleisten für Bildlaufelemente. Informationen zum Skinning von Bildlaufleisten finden Sie unter [„Verwenden von Skins mit der UIScrollBar-Komponente“](#) auf Seite 141.

## Anpassen der Slider-Komponente

Sie können die Slider-Komponente beim Authoring und zur Laufzeit horizontal transformieren. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder die entsprechenden Eigenschaften der Slider-Klasse, zum Beispiel `width` und `scaleX`.

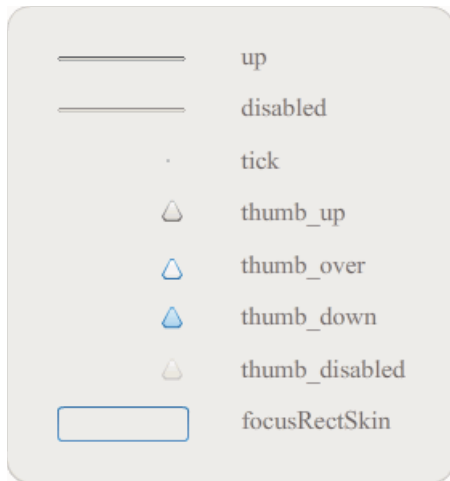
Sie können einen Schieberegler nur verlängern, jedoch nicht die Höhe vergrößern. Flash ignoriert die `height`-Eigenschaft und den `height`-Parameter der `setSize()`-Methode. Sie können jedoch einen vertikalen Schieberegler erstellen und vertikal verlängern.

## Stile und die Slider-Komponente

Die Stile der Slider-Komponente legen nur die Klassen für die Skins sowie einen Wert für `FocusRectPadding` fest, der angibt, wie viele Pixel für die Auffüllung zwischen der Begrenzungsbox der Komponente und dem äußeren Rand liegen. Weitere Informationen zum Verwenden von Skin-Stilen finden Sie unter [„Überblick über Skins“](#) auf Seite 107.

## Skins und die Slider-Komponente

Die Slider-Komponente verfügt über die folgenden Skins, die Sie bearbeiten können, um das Erscheinungsbild zu ändern.



Slider-Skins

Im folgenden Beispiel wird die Farbe der Up-Spur in Blau geändert.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die Slider-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
- 3 Doppelklicken Sie auf die Slider-Komponente, um ein Fenster mit ihren Skins zu öffnen.
- 4 Doppelklicken Sie an ihrem Registrierungspunkt auf die Up-Spur, um sie im Symbolbearbeitungsmodus zu öffnen.
- 5 Stellen Sie die Vergrößerung auf 800 % ein, um das Symbol besser bearbeiten zu können. Die Spur des Schiebereglers besteht aus drei Leisten.
- 6 Klicken Sie auf die oberste Leiste, um sie auszuwählen. Wenn sie ausgewählt ist, wird die entsprechende Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt.
- 7 Wählen Sie in der Farbauswahl „Füllung“ im Eigenschafteninspektor die Farbe #000066 aus, um sie für die obere Leiste der Schiebereglerspur zu verwenden.
- 8 Klicken Sie auf die mittlere Leiste der Schiebereglerspur, um sie auszuwählen. Wenn sie ausgewählt ist, wird die entsprechende Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt.
- 9 Wählen Sie in der Farbauswahl „Füllung“ im Eigenschafteninspektor die Farbe #0066FF aus, um sie für die mittlere Leiste der Schiebereglerspur zu verwenden.
- 10 Klicken Sie auf die untere Leiste der Schiebereglerspur, um sie auszuwählen. Wenn sie ausgewählt ist, wird die entsprechende Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt.
- 11 Wählen Sie in der Farbauswahl „Füllung“ im Eigenschafteninspektor die Farbe #00CCFF aus, um sie für die untere Leiste der Schiebereglerspur zu verwenden.
- 12 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 13 Wählen Sie „Steuerung“ > „Film testen“.

Der Schieberegler sollte jetzt wie in der folgenden Abbildung angezeigt werden.



## Anpassen der TextArea-Komponente

TextArea-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder die entsprechenden Eigenschaften der TextArea-Klasse, zum Beispiel `height` und `width`, `scaleX` und `scaleY`.

Beim Ändern der Größe einer TextArea-Komponente wird die Größe des Rahmens an die Begrenzungsbox angepasst. Falls Bildlaufleisten erforderlich sind, werden sie am unteren und rechten Rand platziert. Anschließend wird die Größe des Textbereichs innerhalb des verbleibenden Bereichs angepasst. Eine TextArea-Komponente enthält keine Elemente mit fester Größe. Wenn eine TextArea-Komponente nicht breit genug für den enthaltenen Text ist, wird der Text abgeschnitten.

### Stile und die TextArea-Komponente

Die Stile der TextArea-Komponente legen die Werte für die Skins, Auffüllung und Textformatierung beim Zeichnen der Komponente fest. Die Stile `textFormat` und `disabledTextFormat` bestimmen den Stil des Textes, der in der TextArea-Komponente angezeigt wird. Weitere Informationen zu Skin-Stileigenschaften finden Sie unter [„Verwenden von Skins mit der TextArea-Komponente“](#) auf Seite 135.

Im folgenden Beispiel wird der Stil `disabledTextFormat` so eingestellt, dass die Darstellung des Textes sich ändert, wenn die TextArea-Komponente deaktiviert ist. Dasselbe Verfahren gilt jedoch auch zum Festlegen des `textFormat`-Stils für eine aktivierte TextArea-Komponente.

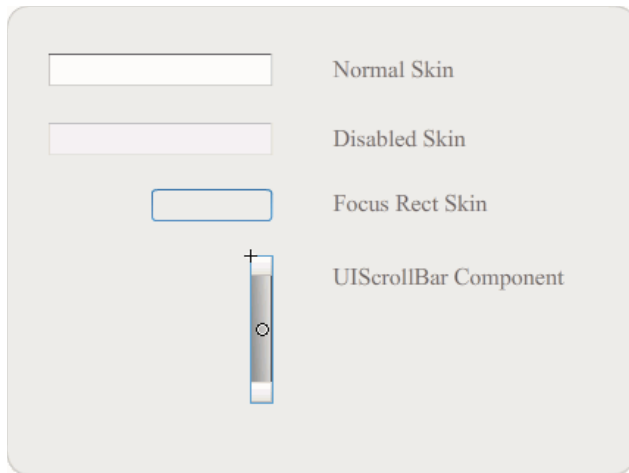
- 1 Erstellen Sie eine neue Flash-Datei.
- 2 Ziehen Sie eine TextArea-Komponente auf die Bühne und geben Sie ihr den Instanznamen **myTa**.
- 3 Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Hauptzeitleiste den folgenden Code hinzu:

```
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 Wählen Sie „Steuerung“ > „Film testen“.

### Verwenden von Skins mit der TextArea-Komponente

Die TextArea-Komponente verfügt über die folgenden Skins, die Sie bearbeiten können, um das Erscheinungsbild zu ändern.



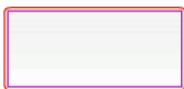
TextArea-Skins

**Hinweis:** Wenn Sie die ScrollBar-Skin in einer Komponente ändern, wird sie auch für alle anderen Komponenten, die die ScrollBar-Komponente verwenden, geändert.

Im folgenden Verfahren werden die Rahmenfarben der Focus Rect Skin, die angezeigt wird, wenn das Textfeld den Fokus erhält, und der Normal-Skin geändert.

- 1 Erstellen Sie eine neue Flash-Datei.
- 2 Ziehen Sie eine TextArea-Komponente auf die Bühne und doppelklicken Sie darauf, um das Fenster mit den Skinsymbolen zu öffnen.
- 3 Doppelklicken Sie auf Focus Rect Skin.
- 4 Klicken Sie auf den Rahmen der Focus Rect Skin, um ihn auszuwählen. Wenn er ausgewählt ist, erscheint die aktuelle Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor.
- 5 Klicken Sie auf die Farbauswahl „Füllung“ im Eigenschafteninspektor, um sie zu öffnen, und wählen Sie die Farbe #CC0000 aus, um sie für den Rahmen zu verwenden.
- 6 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 7 Doppelklicken Sie auf die TextArea-Komponente, um ein Bedienfeld mit ihren Skinsymbolen zu öffnen.
- 8 Doppelklicken Sie auf die Normal-Skin.
- 9 Wählen Sie nacheinander alle Seiten des Rahmens der Normal-Skin aus und stellen Sie die Farbe auf #990099 ein.
- 10 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 11 Wählen Sie „Steuerung“ > „Film testen“.

Wenn Sie die TextArea-Komponente auswählen, um mit der Texteingabe zu beginnen, werden ihre Ränder wie in der folgenden Abbildung angezeigt:



Der äußere Rahmen ist die Focus Rect Skin und der innere Rahmen ist der Rahmen der Normal-Skin.

Informationen zum Bearbeiten der UIScrollBar-Skin finden Sie unter „[Anpassen der UIScrollBar-Komponente](#)“ auf Seite 141.

## Anpassen der TextInput-Komponente

Sie können die Größe einer TextInput-Instanz beim Authoring und zur Laufzeit ändern. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` und die entsprechenden Eigenschaften der TextInput-Klasse, zum Beispiel `height`, `width`, `scaleX` und `scaleY`.

Beim Ändern der Größe einer TextInput-Komponente wird die Größe des Rahmens an die neue Begrenzungsbox angepasst. In TextInput-Komponenten werden keine Bildlaufleisten verwendet, da die Einfügemarke durch die Interaktion des Benutzers mit dem Text automatisch verschoben wird. Anschließend wird die Größe des Textfelds innerhalb des verbleibenden Bereichs angepasst. Eine TextInput-Komponente enthält keine Elemente mit fester Größe. Wenn die TextInput-Komponente zu klein für den anzuzeigenden Text ist, wird der Text abgeschnitten.

### Stile und die TextInput-Komponente

Die Stile der TextInput-Komponente legen die Werte für die Skins, Auffüllung und Textformatierung beim Zeichnen der Komponente fest. Die Stile `textFormat` und `disabledTextFormat` bestimmen den Stil des Textes, der in der Komponente angezeigt wird. Weitere Informationen zu Skin-Stileigenschaften finden Sie unter „[Skins und die TextInput-Komponente](#)“ auf Seite 137.

Im folgenden Beispiel wird der `textFormat`-Stil festgelegt, um Schriftart, Größe und Farbe des Textes zu definieren, der in der TextInput-Komponente angezeigt wird. Mit demselben Verfahren wird auch der `disabledTextFormat`-Stil festgelegt, der angewendet wird, wenn die Komponente deaktiviert ist.

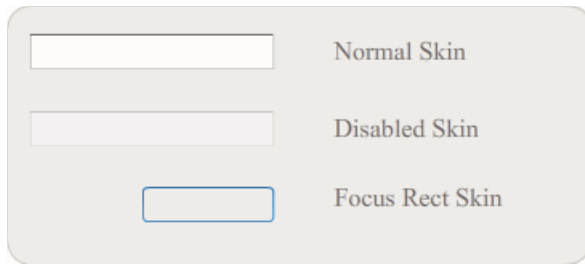
- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine TextInput-Komponente auf die Bühne und geben Sie ihr den Instanznamen **myTi**.
- 3 Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Hauptzeitleiste den folgenden Code hinzu:

```
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 Wählen Sie „Steuerung“ > „Film testen“.

### Skins und die TextInput-Komponente

Die TextInput-Komponente verfügt über die folgenden Skins, die Sie bearbeiten können, um das Erscheinungsbild zu ändern.



*TextInput-Bezeichnung*

Im folgenden Verfahren werden die Rahmen- und Hintergrundfarben einer TextInput-Komponente geändert:

- 1 Erstellen Sie eine neue Flash-Datei.
- 2 Ziehen Sie eine TextInput-Komponente auf die Bühne und doppelklicken Sie darauf, um das Fenster mit den Skins zu öffnen.
- 3 Doppelklicken Sie auf die Normal-Skin.
- 4 Stellen Sie die Vergrößerung auf 800 % ein, um das Symbol besser bearbeiten zu können.
- 5 Wählen Sie nacheinander alle Seiten des Rahmens der Normal-Skin aus und stellen Sie die Farbe auf #993399 ein.
- 6 Doppelklicken Sie auf den Hintergrund, bis seine Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt wird. Wählen Sie die Farbe #99CCCC aus, um sie für den Hintergrund zu verwenden.
- 7 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 8 Wählen Sie „Steuerung“ > „Film testen“.

Das Texteingabefeld sollte jetzt wie in der folgenden Abbildung aussehen:



## Anpassen der TileList-Komponente

TileList-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder die entsprechenden Eigenschaften, zum Beispiel `width`, `height`, `columnCount`, `rowCount`, `scaleX` und `scaleY`. Die in der TileList-Komponente enthaltene ScrollBar-Komponente wird mit dem Listenfeld skaliert.

### Stile und die TileList-Komponente

Die Stile der TileList-Komponente legen die Werte für die Skins, Auffüllung und Textformatierung beim Zeichnen der Komponente fest. Die Stile `textFormat` und `disabledTextFormat` bestimmen den Stil des Textes, der in der Komponente angezeigt wird. Weitere Informationen zu den Skin-Stilen finden Sie unter „[Verwenden von Skins mit der TileList-Komponente](#)“ auf Seite 139.

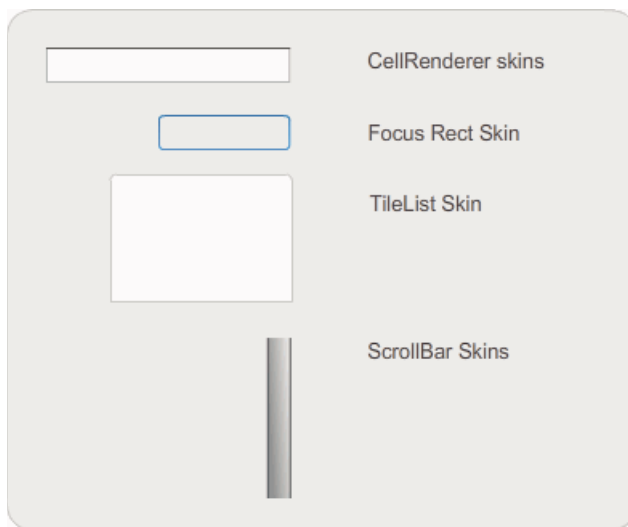
Im folgenden Beispiel wird die Methode `setRendererStyle()` mit dem Stil `textFormat` aufgerufen, um die Schriftart, die Schriftgröße, die Schriftfarbe und die Textattribute der Bezeichnungen, die in der `TileList`-Instanz angezeigt werden, festzulegen. Mit demselben Verfahren wird auch der Stil `disabledTextFormat` festgelegt, der angewendet wird, wenn die Eigenschaft `enabled` auf `false` eingestellt ist.

- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie die `TileList`-Komponente auf die Bühne und geben Sie ihr den Instanznamen `myTl`.
- 3 Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Zeitleiste den folgenden Code hinzu:

```
myTl.setSize(100, 100);
myTl.addItem({label:"#1"});
myTl.addItem({label:"#2"});
myTl.addItem({label:"#3"});
myTl.addItem({label:"#4"});
var tf:TextFormat = new TextFormat();
tf.font = "Arial";
tf.color = 0x00FF00;
tf.size = 16;
tf.italic = true;
tf.bold = true;
tf.underline = true;
tf.align = "center";
myTl.setRendererStyle("textFormat", tf);
```

## Verwenden von Skins mit der TileList-Komponente

Die `TileList`-Komponente verfügt über eine `TileList`-Skin, eine `CellRenderer`-Skin und eine `ScrollBar`-Skin. Sie können diese Skins bearbeiten, um die Darstellung der `TileList`-Komponente zu ändern.



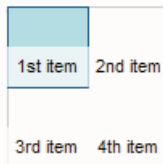
*TileList-Skins*

**Hinweis:** Wenn Sie die `ScrollBar`-Skin in einer Komponente ändern, wird sie auch für alle anderen Komponenten, die die `ScrollBar`-Komponente verwenden, geändert.

Mit dem folgenden Verfahren wird die Farbe der CellRenderer-Skin „Selected\_Up“ der TileList-Komponente geändert.

- 1 Erstellen Sie ein Flash-Dokument (ActionScript 3.0).
- 2 Ziehen Sie eine TileList-Komponente auf die Bühne und doppelklicken Sie darauf, um die Palette mit den Skins zu öffnen.
- 3 Doppelklicken Sie auf die CellRenderer-Skin, dann auf die Selected\_Up-Skin und dann auf den rechteckigen Hintergrund.
- 4 Wählen Sie in der Farbauswahl „Füllung“ im Eigenschafteninspektor die Farbe #99FFFF für die Skin „Selected\_Up“ aus.
- 5 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 6 Doppelklicken Sie auf der Registerkarte „Parameter“ des Eigenschafteninspektors auf die zweite Spalte der dataProvider-Zeile, um das Dialogfeld „Werte“ zu öffnen. Fügen Sie Elemente mit den folgenden Bezeichnungen hinzu: 1. Element, 2. Element, 3. Element, 4. Element.
- 7 Wählen Sie „Steuerung“ > „Film testen“.
- 8 Klicken Sie auf eine Zelle in der TileList-Komponente, um sie auszuwählen, und bewegen Sie dann den Mauszeiger von der ausgewählten Zelle weg.

Die ausgewählte Zelle sollte jetzt wie in der folgenden Abbildung angezeigt werden:



*TileList-Komponente mit geänderter Farbe für die Skin „Selected\_Up“*

## Anpassen der UILoader-Komponente

UILoader-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder die entsprechenden Eigenschaften, zum Beispiel `width`, `height`, `scaleX` und `scaleY`.

Das Größenverhalten der UILoader-Komponente wird von der Eigenschaft `scaleContent` gesteuert. Wenn `scaleContent` auf `true` gesetzt ist, wird der Inhalt skaliert, damit er in die Grenzen des Loaders passt (und erneut skaliert, wenn `setSize()` aufgerufen wird). Wenn die Eigenschaft `scaleContent` auf `false` gesetzt wurde, wird die Größe der Komponente auf die Größe des Inhalts fixiert. In diesem Fall bleiben `setSize()` und die Größeneigenschaften wirkungslos.

Die UILoader-Komponente verfügt über keine UI-Elemente, auf die Sie Stile oder Skins anwenden können.



## Anpassen der UIScrollBar-Komponente

UIScrollBar-Komponenten können sowohl beim Authoring als auch zur Laufzeit horizontal und vertikal transformiert werden. Eine vertikale UIScrollBar kann nicht in der Breite geändert werden, bei einer horizontalen UIScrollBar können Sie die Höhe nicht ändern. Beim Authoring wählen Sie hierzu die Komponente auf der Bühne aus und verwenden anschließend das Werkzeug „Frei transformieren“ oder die Befehle unter „Modifizieren“ > „Transformieren“. Zur Laufzeit verwenden Sie die Methode `setSize()` oder die entsprechenden Eigenschaften der UIScrollBar-Klasse, zum Beispiel `width`, `height`, `scaleX` und `scaleY`.

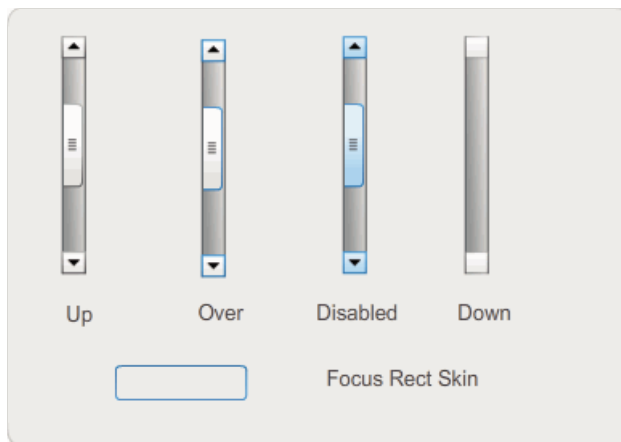
**Hinweis:** Mit der Methode `setSize()` können Sie nur die Breite einer horizontalen Bildlaufleiste oder die Höhe einer vertikalen Bildlaufleiste ändern. Beim Authoring können Sie die Höhe einer horizontalen Bildlaufleiste oder die Breite einer vertikalen Bildlaufleiste ändern, die Werte werden beim Veröffentlichen des Films jedoch zurückgesetzt. Nur die Länge einer Bildlaufleiste kann geändert werden.

## Verwenden von Stilen mit der UIScrollBar-Komponente

Die Stile der UIScrollBar-Komponente legen nur die Klassen für die Skins sowie einen Wert für `FocusRectPadding` fest, der angibt, wie viele Pixel für die Auffüllung zwischen der Begrenzungsbox der Komponente und dem äußeren Rand liegen. Weitere Informationen zum Verwenden von Skin-Stilen finden Sie unter „[Überblick über Skins](#)“ auf Seite 107.

## Verwenden von Skins mit der UIScrollBar-Komponente

Die UIScrollBar-Komponente verwendet die folgenden Skins.



UIScrollBar-Skins

Horizontale und vertikale Bildlaufleisten verwenden dieselben Skins. Wenn eine horizontale Bildlaufleiste angezeigt wird, dreht die UIScrollBar-Komponente die Skins entsprechend.

**Hinweis:** Wenn Sie die ScrollBar-Skin in einer Komponente ändern, wird sie auch für alle anderen Komponenten, die die ScrollBar-Komponente verwenden, geändert.

Im folgenden Beispiel wird die Farbe des Bildlauffelds und der Pfeilschaltflächen der UIScrollBar-Komponente geändert.

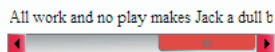
- 1 Erstellen Sie ein neues Flash-Dokument (ActionScript 3.0).

- 2 Ziehen Sie die UIScrollBar-Komponente auf die Bühne und geben Sie ihr den Instanznamen **mySb**. Setzen Sie die Richtung auf der Registerkarte „Parameter“ auf den Wert „horizontal“.
- 3 Doppelklicken Sie auf die Bildlaufleiste, um die Palette ihrer Skins zu öffnen.
- 4 Klicken Sie auf die Up-Skin, um sie auszuwählen.
- 5 Stellen Sie die Vergrößerung auf 400 % ein, um das Symbol besser bearbeiten zu können.
- 6 Doppelklicken Sie auf den Hintergrund des Pfeils nach rechts (bzw. nach oben, falls es sich um eine vertikale Bildlaufleiste handelt), bis der Hintergrund ausgewählt ist und seine Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor angezeigt wird.
- 7 Wählen Sie die Farbe #CC0033 aus, um sie für den Hintergrund der Schaltfläche zu verwenden.
- 8 Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um zum Dokumentbearbeitungsmodus zurückzukehren.
- 9 Wiederholen Sie die Schritte 6, 7 und 8 für das Bildlauffeld und den linken Pfeil (oder den nach unten gerichteten Pfeil bei einer vertikalen Bildlaufleiste).
- 10 Fügen Sie in Bild 1 der Zeitleiste im Bedienfeld „Aktionen“ den folgenden Code hinzu, um die Bildlaufleiste einem Textfeld zuzuordnen.

```
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 Wählen Sie „Steuerung“ > „Film testen“.

Der UIScrollBar-Komponente sollte jetzt wie in der folgenden Abbildung angezeigt werden.



*Horizontale Bildlaufleiste mit Bildlauffeld und linker und rechter Pfeilschaltfläche in Rot*

# Kapitel 6: Verwenden der FLVPlayback-Komponente

Mit der FLVPlayback-Komponente können Sie ganz einfach einen Video-Player in Ihre Adobe Flash CS5 Professional-Anwendung einschließen, um progressiv heruntergeladene Videodateien über HTTP oder Streaming-Videodateien von einem Adobe Macromedia Flash Media Server oder von einem Flash Video Streaming Service (FVSS) abzuspielen.

Adobe Flash Player 9 Update 3 (Version 9.0.115.0 oder höher) kann mit wesentlichen Verbesserungen beim Abspielen von Videos in Flash Player aufwarten. Dieses Update umfasst Änderungen an der FLVPlayback-Komponente, bei denen die Videohardware des Endbenutzers genutzt wird, um beim Abspielen von Video eine bessere Leistung zu erzielen. Die Verbesserungen der FLVPlayback-Komponente führen auch zu einer höheren Genauigkeit bei der Anzeige von Videodateien im Vollbildmodus.

Weiterhin wurde die Funktionalität der FLVPlayback-Komponente in Flash Player 9 Update 3 verbessert, da nun hochauflösende MPEG-4-Videoformate in H.264-Kodierung nach dem Industriestandard unterstützt werden. Zu diesen Formaten zählen MP4, M4A, MOV, MP4V, 3GP und 3G2.

**Hinweis:** Geschützte MP4-Dateien werden nicht unterstützt (beispielsweise von Apple® iTunes® heruntergeladene oder über FairPlay® digital verschlüsselte Dateien).

Die benutzerfreundliche FLVPlayback-Komponente verfügt über die folgenden Merkmale und Vorteile:

- Kann auf die Bühne gezogen und schnell und einfach implementiert werden.
- Unterstützt den Vollbildmodus.
- Bietet verschiedene vordefinierte *Skins* zum individuellen Gestalten der Abspielsteuerungen.
- Ermöglicht die Auswahl von Farb- und Alphawerten für vordefinierte *Skins*.
- Ermöglicht erfahrenen Benutzern das Erstellen eigener *Skins*.
- Bietet eine Live-Vorschau beim Authoring.
- Bietet Layout-Eigenschaften, um die Videodatei bei Größenänderungen zu zentrieren.
- Ermöglicht den Abspielbeginn, wenn ein ausreichender Anteil der progressiv heruntergeladenen Videodatei heruntergeladen wurde.
- Bietet Cue-Points, mit deren Hilfe Sie Ihr Video mit Text, Grafiken und Animationen synchronisieren können.
- Die SWF-Datei wird nicht unnötig groß.

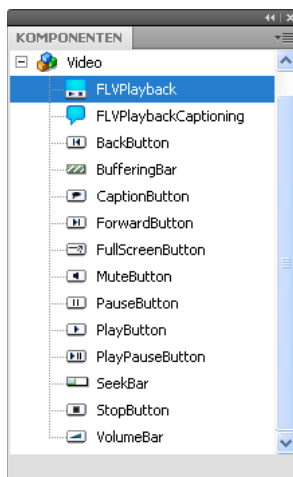
## Verwenden der FLVPlayback-Komponente

Sie verwenden die FLVPlayback-Komponente, indem Sie sie auf die Bühne ziehen und eine abzuspielende Videodatei angeben. Darüber hinaus können Sie auch noch verschiedene Parameter einstellen, die das Verhalten der Komponente regeln und die Videodatei beschreiben.

**Verwenden der FLVPlayback-Komponente**

Die FLVPlayback-Komponente enthält auch eine ActionScript-API (API = Application Programming Interface, Anwendungsprogrammierschnittstelle). Die API umfasst die folgenden Klassen. Eine ausführliche Beschreibung dieser Klassen finden Sie im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#) beschrieben werden: CuePointType, FLVPlayback, FLVPlaybackCaptioning, NCManager, NCManagerNative, VideoAlign, VideoError, VideoPlayer, VideoState und verschiedene Ereignisklassen - AutoLayoutEvent, LayoutEvent, MetadataEvent, SkinErrorEvent, SoundEvent, VideoEvent und VideoProgressEvent.

Die FLVPlayback-Komponente enthält die FLV Playback Custom UI-Komponenten. Die FLVPlayback-Komponente ist eine Kombination aus dem Anzeigebereich oder Video-Player, in dem Sie die Videodatei anzeigen, und den Steuerelementen, die Sie zum Abspielen der Datei verwenden. Die FLV Playback Custom UI-Komponenten bieten Steuerelemente und Mechanismen zum Abspielen, Beenden, Anhalten und so weiter. Zu diesen Steuerelementen gehören BackButton, BufferingBar, CaptionButton (für FLVPlaybackCaptioning), ForwardButton, FullScreenButton, MuteButton, PauseButton, PlayButton, PlayPauseButton, SeekBar, StopButton und VolumeBar. Die FLVPlayback-Komponente und die FLV Playback Custom UI-Steuerelemente werden im Bedienfeld „Komponenten“ angezeigt, wie in der folgenden Abbildung dargestellt:



FLVPlayback-Komponenten im Bedienfeld „Komponenten“

Der Vorgang, bei dem Sie der FLVPlayback-Komponente Abspielsteuerungen hinzufügen, wird *Skinning* genannt. Die FLVPlayback-Komponente verfügt über eine ursprüngliche Standardskin, SkinOverAll.swf, mit Steuerungen für die Funktionen Abspielen, Stopp, Zurück, Vor, Suchleiste, Ton aus, Lautstärke, Vollbild und Untertitel. Sie können diese Skin mit den folgenden Methoden ändern:

- Wählen Sie eine von mehreren vordefinierten Skins aus.
- Erstellen Sie eine benutzerdefinierte Skin und fügen Sie sie der Gruppe der vordefinierten Skins hinzu.
- Wählen Sie einzelne Steuerelemente aus den FLV Playback Custom UI-Komponenten aus und passen Sie sie individuell an.

Wenn Sie eine vordefinierte Skin verwenden, können Sie die Skinfarbe und die Alphawerte separat festlegen. Dies kann beim Authoring oder zur Laufzeit geschehen. Weitere Informationen finden Sie unter [„Auswählen einer vordefinierten Skin“](#) auf Seite 163.

Nachdem Sie eine andere Skin ausgewählt haben, wird die ausgewählte Skin zur neuen Standardskin.

Weitere Informationen zum Auswählen oder Erstellen einer Skin für die FLVPlayback-Komponente finden Sie unter [„Anpassen der FLVPlayback-Komponente“](#) auf Seite 163.

## Erstellen einer Anwendung mit der FLVPlayback-Komponente

Sie haben die folgenden Möglichkeiten, um eine FLVPlayback-Komponente in Ihre Anwendung einzuschließen:

- Ziehen Sie die FLVPlayback-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und geben Sie einen Wert für den Parameter `source` an.
- Erstellen Sie mit dem Videoimportassistenten eine Komponente auf der Bühne und wählen Sie eine Skin aus, um sie individuell anzupassen.
- Verwenden Sie den Konstruktor `FLVPlayback()`, um eine FLVPlayback-Instanz dynamisch auf der Bühne zu erstellen (dazu muss sich die Komponente in der Bibliothek befinden).

**Hinweis:** Wenn Sie eine FLVPlayback-Instanz mithilfe von ActionScript erstellen, müssen Sie ihr auch eine Skin zuweisen, indem Sie die Eigenschaft `skin` mit ActionScript festlegen. Wenn Sie eine Skin auf diese Weise anwenden, wird sie nicht automatisch mit der SWF-Datei veröffentlicht. Sie müssen sowohl die SWF-Datei der Anwendung als auch die SWF-Datei der Skin auf den Anwendungsserver kopieren, andernfalls ist die Skin-SWF-Datei beim Ausführen der Anwendung nicht verfügbar.

### Ziehen der FLVPlayback-Komponente aus dem Bedienfeld „Komponenten“

- 1 Klicken Sie im Bedienfeld „Komponenten“ auf das Pluszeichen (+), um den Videoeintrag zu öffnen.
- 2 Ziehen Sie die FLVPlayback-Komponente auf die Bühne.
- 3 Während die FLVPlayback-Komponente auf der Bühne ausgewählt ist, suchen Sie auf der Registerkarte „Parameter“ des Komponenten-Inspektors die Zelle „Wert“ für den Parameter `source` und geben Sie einen String ein, der Folgendes angibt:
  - Lokaler Pfad einer Videodatei
  - URL einer Videodatei
  - URL einer SMIL-Datei, in der das Abspielen einer Videodatei beschrieben wird (SMIL = Synchronized Multimedia Integration Language)Informationen zum Erstellen einer SMIL-Datei, die eine oder mehrere FLV-Dateien beschreibt, finden Sie unter [„Verwenden einer SMIL-Datei“](#) auf Seite 174.
- 4 Während die FLVPlayback-Komponente auf der Bühne ausgewählt ist, klicken Sie auf der Registerkarte „Parameter“ des Komponenten-Inspektors auf die Zelle „Wert“ für den Parameter `skin`.
- 5 Klicken Sie auf das Lupensymbol, um das Dialogfeld „Skin auswählen“ zu öffnen.
- 6 Wählen Sie eine der folgenden Optionen:
  - Wählen Sie in der Dropdownliste „Skin“ eine der vordefinierten Skins aus, um eine Gruppe von Abspielsteuerungen an die Komponente anzuhängen.
  - Wenn Sie eine benutzerdefinierte Skin erstellt haben, wählen Sie in der Dropdownliste den Eintrag „Benutzerdefinierte Skin-URL“ und geben Sie in das Feld „URL“ die URL für die SWF-Datei mit der Skin ein.
  - Wählen Sie „Aus“ und ziehen Sie individuelle FLV Playback Custom UI-Komponenten auf die Bühne, um Abspielsteuerungen hinzuzufügen.

**Hinweis:** In den ersten beiden Fällen wird eine Vorschau der Skin im Anzeigebereich über der Dropdownliste angezeigt. Mit der Farbauswahl können Sie die Farbe der Skin ändern.

Um die Farbe einer benutzerdefinierten UI-Steuerung zu ändern, müssen Sie die Komponente individuell anpassen. Weitere Informationen zur Verwendung von benutzerdefinierten UI-Steuerungen finden Sie unter [„Separates Zuweisen von Skins für FLV Playback Custom UI-Komponenten“](#) auf Seite 164.
- 7 Klicken Sie auf „OK“, um das Dialogfeld „Skin auswählen“ zu schließen.

**Verwenden der FLVPlayback-Komponente**

- 8 Wählen Sie „Steuerung“ > „Film testen“, um die SWF-Datei auszuführen und das Video zu starten.

Im folgenden Verfahren wird der Videoimportassistent verwendet, um eine FLVPlayback-Komponente hinzuzufügen.

**Verwenden des Videoimportassistenten:**

- 1 Wählen Sie „Datei“ > „Importieren“ > „Video importieren“.
- 2 Geben Sie den Speicherort der Videodatei an, indem Sie eine der folgenden Optionen auswählen:
  - Auf Ihrem Computer
  - Bereits auf einem Webserver, FVSS-Server oder Flash Media Server bereitgestellt
- 3 Geben Sie je nach Auswahl den Dateipfad oder die URL für den Speicherort der Videodatei ein und klicken Sie dann auf „Weiter“.
- 4 Wenn Sie einen Dateipfad angegeben haben, wird als Nächstes das Dialogfeld „Bereitstellung“ angezeigt, in dem Sie eine der aufgeführten Optionen wählen können, um anzugeben, wie das Video bereitgestellt werden soll:
  - Von einem standardmäßigen Webserver progressiv herunterladen
  - Vom Flash Video Streaming Service (FVSS) streamen
  - Von Flash Media Server streamen
  - Video in eine SWF-Datei einbetten und in der Zeitleiste abspielen

**Wichtig:** Wählen Sie nicht die Option zum Einbetten des Videos. Die FLVPlayback-Komponente spielt nur externes Streaming-Video ab. Mit dieser Option wird keine FLVPlayback-Komponente auf der Bühne platziert.

- 5 Klicken Sie auf „Weiter“.
- 6 Wählen Sie eine der folgenden Optionen:
  - Wählen Sie in der Dropdownliste „Skin“ eine der vordefinierten Skins aus, um eine Gruppe von Abspielsteuerungen an die Komponente anzuhängen.
  - Wenn Sie eine benutzerdefinierte Skin für die Komponente erstellt haben, wählen Sie in der Dropdownliste den Eintrag „Benutzerdefinierte Skin-URL“ und geben Sie in das Feld „URL“ die URL für die SWF-Datei mit der Skin ein.
  - Wählen Sie „Aus“ und ziehen Sie individuelle FLV Playback Custom UI-Komponenten auf die Bühne, um Abspielsteuerungen hinzuzufügen.

**Hinweis:** In den ersten beiden Fällen wird eine Vorschau der Skin im Anzeigebereich über der Dropdownliste angezeigt.

- 7 Klicken Sie auf „OK“, um das Dialogfeld „Skin auswählen“ zu schließen.
- 8 Lesen Sie im Dialogfeld „Videoimport fertig stellen“, was als Nächstes passiert, und klicken Sie auf „Fertig stellen“.
- 9 Wenn Sie Ihre FLA-Datei noch nicht gespeichert haben, wird das Dialogfeld „Speichern unter“ angezeigt.
- 10 Wählen Sie „Steuerung“ > „Film testen“, um die SWF-Datei auszuführen und das Video zu starten.

Im folgenden Verfahren wird die FLVPlayback-Komponente mithilfe von ActionScript hinzugefügt.

**Dynamisches Erstellen einer Instanz mithilfe von ActionScript:**

- 1 Ziehen Sie die FLVPlayback-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ („Fenster“ > „Bibliothek“).

**Verwenden der FLVPlayback-Komponente**

- 2 Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Zeitleiste den folgenden Code hinzu: Geben Sie dabei für *install\_drive* das Laufwerk ein, auf dem Flash installiert wurde, und ändern Sie den Pfad, um den Speicherort des Skins-Ordners in Ihrer Installation anzugeben:

Windows:

```
import fl.video.*;
var my_FLVPlybk = new FLVPlayback();
my_FLVPlybk.x = 100;
my_FLVPlybk.y = 100;
addChild(my_FLVPlybk);
my_FLVPlybk.skin = "file:///install_drive|/Program Files/Adobe/Adobe Flash
CS5/en/Configuration/FLVPlayback Skins/ActionScript 3.0/SkinOverPlaySeekMute.swf"
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/water.flv";
```

Macintosh:

```
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";
```

**Hinweis:** Wenn Sie die Eigenschaften *source* und *skin* nicht festlegen, scheint der erstellte Movieclip leer zu sein.

- 3 Wählen Sie „Steuerung“ > „Film testen“, um die SWF-Datei auszuführen und die Videodatei zu starten.

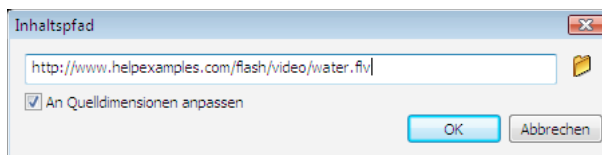
## Parameter der FLVPlayback-Komponente

Für jede Instanz der FLVPlayback-Komponente können Sie die folgenden Parameter im Komponenten-Inspektor oder im Eigenschafteninspektor festlegen: *align*, *autoPlay*, *cuePoints*, *preview*, *scaleMode*, *skin*, *skinAutoHide*, *skinBackgroundAlpha*, *skinBackgroundColor*, *source* und *volume*. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft mit demselben Namen. Wenn Sie diesen Parametern einen Wert zuweisen, legen Sie den ursprünglichen Status der Eigenschaft in der Anwendung fest. Durch Festlegen der Eigenschaft in ActionScript wird der Wert, den Sie für den Parameter angeben, außer Kraft gesetzt. Informationen zu den möglichen Werten für diese Parameter finden Sie im Abschnitt zur FLVPlayback-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

### Festlegen des *source*-Parameters für FLVPlayback

Mit dem Parameter *source* geben Sie den Namen und den Speicherort der Videodatei an, wodurch Flash mitgeteilt wird, wie die Datei abzuspielen ist.

Öffnen Sie das Dialogfeld „Inhaltspfad“, indem Sie im Komponenten-Inspektor auf die Zelle „Wert“ für den Parameter *source* doppelklicken.



Dialogfeld „Inhaltspfad“ für die FLVPlayback-Komponente

**Verwenden der FLVPlayback-Komponente**

Das Dialogfeld „Inhaltspfad“ enthält ein Kontrollkästchen, über das Sie festlegen können, ob die FLVPlayback-Instanz auf der Bühne den Abmessungen der Videoquelldatei entsprechen soll. Die Videoquelldatei enthält bevorzugte Abmessungen (Höhe und Breite) für das Abspielen. Wenn Sie diese Option aktivieren, werden die Abmessungen der FLVPlayback-Instanz an diese bevorzugten Abmessungen angepasst.

**Die Quelle**

Geben Sie die URL oder den lokalen Pfad für die Videodatei oder für eine XML-Datei an, in der festgelegt ist, wie die Videodatei abgespielt wird. Wenn Ihnen der genaue Speicherort der Videodatei nicht bekannt ist, klicken Sie auf das Ordnersymbol, um ein Dialogfeld zu öffnen, in dem Sie die Datei suchen können. Falls sich die gesuchte Videodatei an oder unter dem Speicherort der SWF-Zieldatei befindet, verwendet Flash automatisch den relativen Pfad zu diesem Speicherort, sodass die Datei von einem Webserver bereitgestellt werden kann. Andernfalls wird ein absoluter Windows- oder Macintosh-Dateipfad verwendet. Zur Eingabe des Namens einer lokalen XML-Datei geben Sie den Pfad und den Namen ein.

Wenn Sie eine HTTP-URL angeben, wird die Videodatei als progressiver Download abgespielt. Wenn Sie eine RTMP-URL angeben, wird die Videodatei von einem Flash Media Server oder FVSS gestreamt. Eine URL zu einer XML-Datei könnte auch eine Streaming-Videodatei von einem Flash Media Server oder FVSS sein.

**Wichtig:**

Sie können auch den Speicherort einer SMIL-Datei angeben, in der beschrieben wird, wie mehrere Videodateistreams für verschiedene Bandbreiten abgespielt werden sollen. Die Datei verwendet zur Beschreibung der FLV-Dateien Synchronized Multimedia Integration Language (SMIL). Eine Beschreibung von SMIL-Dateien finden Sie unter „[Verwenden einer SMIL-Datei](#)“ auf Seite 174.

Sie können den Namen und den Speicherort der Videodatei auch über die ActionScript-Eigenschaft `FLVPlayback.source` und die Methoden `FLVPlayback.play()` und `FLVPlayback.load()` angeben. Diese drei Alternativen haben Vorrang vor dem Parameter `source` im Komponenteninspektor. Weitere Informationen finden Sie in den Einträgen zu `FLVPlayback.source`, `FLVPlayback.play()` und `FLVPlayback.load()` für die FLVPlayback-Klasse im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

**Unterstützung für den Vollbildmodus**

Die ActionScript 3.0-Version der FLVPlayback-Komponente unterstützt den Vollbildmodus, der Flash Player 9.0.28.0 oder höher erfordert. Außerdem muss der HTML-Code richtig für die Anzeige im Vollbildmodus eingerichtet sein. Einige vordefinierte Skins enthalten eine Schaltfläche, mit der der Vollbildmodus ein- und ausgeschaltet werden kann. In der folgenden Abbildung wird das `FullScreenButton`-Symbol rechts in der Steuerungsleiste angezeigt.



Symbol für den Vollbildmodus in der Steuerungsleiste

Der Vollbildmodus wird nur unterstützt, wenn die Eigenschaft `fullScreenTakeOver` den Standardwert `true` aufweist.

Die Unterstützung für den Vollbildmodus ist mit und ohne Unterstützung für die Hardwarebeschleunigung möglich. Informationen zur Unterstützung der Hardwarebeschleunigung finden Sie unter „[Hardwarebeschleunigung](#)“ auf Seite 151.

**Implementieren der Vollbildmodus-Unterstützung für FLVPlayback:**

- 1 Fügen Sie die FLVPlayback-Komponente Ihrer Anwendung hinzu und weisen Sie ihr eine Videodatei zu.



**Verwenden der FLVPlayback-Komponente**

- 2 Wählen Sie für die FLVPlayback-Komponente eine Skin aus, die eine Schaltfläche für den Vollbildmodus aufweist (z. B. SkinUnderPlaySeekFullscreen.swf). Stattdessen können Sie auch den Bereich „Video“ im Bedienfeld „Komponenten“ verwenden, um der FLVPlayback-Komponente die FullScreenButton-UI-Komponente hinzuzufügen.
- 3 Wählen Sie „Datei“ > „Einstellungen für Veröffentlichungen“.
- 4 Klicken Sie im Dialogfeld „Einstellungen für Veröffentlichungen“ auf die Registerkarte „HTML“.
- 5 Wählen Sie auf der Registerkarte „HTML“ im Dropdownmenü „Vorlage“ die Option für Flash mit Vollbildmodus-Unterstützung.
- 6 Aktivieren Sie auf der Registerkarte „HTML“ das Kontrollkästchen „Flash-Version feststellen“ und geben Sie Version 9.0.28 oder höher an, je nach der von Ihnen verwendeten Flash Player-Version.
- 7 Wählen Sie auf der Registerkarte „Formate“ die Optionen für Flash (.swf) und HTML (.html) aus. Sie können die Standarddateinamen ersetzen.
- 8 Klicken Sie auf „Veröffentlichen“ und dann auf „OK“.

Als Alternative zu Schritt 7 können Sie auch auf „OK“ klicken und dann „Datei“ > „Vorschau für Veröffentlichungen“ > „Standard – (HTML)“ auswählen, um die exportierte HTML-Datei automatisch in Ihrem Standardbrowser zu öffnen. Andernfalls öffnen Sie die exportierte HTML-Datei im Browser, um den Vollbildmodus zu testen.

Um die FLVPlayback-Komponente mit Vollbildmodus-Unterstützung in Ihre Webseite einzufügen, öffnen Sie die exportierte HTML-Datei und kopieren Sie den Code, mit dem die SWF-Datei eingebettet wird, in die HTML-Datei für Ihre Webseite. Der Code sollte ungefähr folgendermaßen aussehen:

```
//from the <head> section

<script language="javascript"> AC_FL_RunContent = 0; </script>
<script language="javascript"> DetectFlashVer = 0; </script>
<script src="AC_RunActiveContent.js" language="javascript"></script>
<script language="JavaScript" type="text/javascript">
<!--
// -----
// Globals
// Major version of Flash required
var requiredMajorVersion = 9;
// Minor version of Flash required
var requiredMinorVersion = 0;
// Revision of Flash required
var requiredRevision = 28;
// -----
// -->
</script>

//and from the <body> section

<script language="JavaScript" type="text/javascript">
<!--
if (AC_FL_RunContent == 0 || DetectFlashVer == 0) {
 alert("This page requires AC_RunActiveContent.js.");
} else {
 var hasRightVersion = DetectFlashVer(requiredMajorVersion,
 requiredMinorVersion, requiredRevision);
 if(hasRightVersion) { // if we've detected an acceptable version
 // embed the Flash movie
```

```
AC_FL_RunContent (
 'codebase', 'http://download.macromedia.com/pub/
 shockwave/cabs/flash/swflash.cab#version=9,0,28,0';,
 'width', '550';,
 'height', '400';,
 'src', 'fullscreen';,
 'quality', 'high';,
 'pluginspage', 'http://www.macromedia.com/go/
 getflashplayer';,
 'align', 'middle';,
 'play', &apos>true';,
 'loop', &apos>true';,
 'scale', 'showall';,
 'wmode', 'window';,
 'devicefont', &apos>false';,
 'id', 'fullscreen';,
 'bgcolor', '#ffffff';,
 'name', 'fullscreen';,
 'menu', &apos>true';,
 'allowScriptAccess', 'sameDomain';,
 'allowFullScreen', &apos>true';,
 'movie', 'fullscreen';,
 'salign', '';); //end AC code
} else { // Flash is too old or we can't detect the plug-in.
 var alternateContent = 'Alternative HTML content should be placed
 here.';
 + 'This content requires Adobe Flash Player.';
 + 'Get Flash
 ';
 document.write(alternateContent); // Insert non-Flash content.
}
}
// -->
</script>
<noscript>
 // Provide alternative content for browsers that do not support scripting
 // or for those that have scripting disabled.
 Alternative HTML content should be placed here. This content requires Adobe Flash Player.
 Get Flash
</noscript>
```

Stattdessen können Sie auch die exportierte HTML-Datei als Vorlage für Ihre Webseite verwenden und ihr den weiteren Inhalt hinzufügen. In diesem Fall sollten Sie die HTML-Datei jedoch umbenennen, damit sie nicht versehentlich überschrieben wird, falls Sie die HTML-Datei mit der FLVPlayback-Komponente später erneut aus Flash exportieren.

In jedem Fall müssen Sie auch die Datei „AC\_RunActiveContent.js“, die in denselben Ordner wie die HTML-Datei exportiert wird, auf Ihren Webserver hochladen.

Die ActionScript-Unterstützung für den Vollbildmodus umfasst die Eigenschaften `fullScreenBackgroundColor`, `fullScreenSkinDelay` und `fullScreenTakeOver` sowie die Methode `enterFullScreenDisplayState()`. Informationen zu diesen ActionScript-Elementen finden Sie im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Verwenden von enterFullScreenDisplayState()

Sie können den Vollbildmodus auch über die ActionScript-Methode `enterFullScreenDisplayState()` aufrufen, wie im folgenden Beispiel gezeigt.

```
function handleClick(e:MouseEvent):void {
 myFLVPlayback.enterFullScreenDisplayState();
}
myButton.addEventListener(MouseEvent.CLICK, handleClick);
```

In diesem Beispiel wird der Vollbildmodus *nicht* durch Klicken auf die Vollbildmodus-Schaltfläche einer FLVPlayback-Skin aufgerufen, sondern durch Klicken auf eine Schaltfläche (MyButton), die der Ersteller der Webseite zum Aufrufen des Vollbildmodus eingefügt hat. Beim Klicken auf die Schaltfläche wird die Ereignisprozedur `handleClick` ausgelöst, die wiederum die Methode `enterFullScreenDisplayState()` aufruft.

Die Methode `enterFullScreenDisplayState()` stellt die Eigenschaft `Stage.displayState` auf den Wert `StageDisplayState.FULL_SCREEN` ein und bewirkt daher dieselben Einschränkungen wie auch die Eigenschaft `displayState`. Weitere Informationen zur Methode `enterFullScreenDisplayState()` und zur Eigenschaft `Stage.displayState` finden Sie im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Hardwarebeschleunigung

Flash Player 9.0.115.0 und neuere Versionen enthalten Code, der verfügbare Videohardware nutzt, um die Leistung und Genauigkeit beim Abspielen von FLV-Dateien zu erhöhen, die FLVPlayback im Vollbildmodus wiedergibt. Sofern die Voraussetzungen erfüllt sind und die Eigenschaft `fullScreenTakeOver` auf `true` eingestellt ist, skaliert Flash Player die Videodatei über die Hardwarebeschleunigung anstatt über die Software. Wenn die FLVPlayback-Komponente in einer früheren Flash Player-Version ausgeführt wird oder wenn die Voraussetzungen für die Hardwarebeschleunigung nicht erfüllt sind, erfolgt die Skalierung der Videodatei wie bisher durch Flash Player selbst.

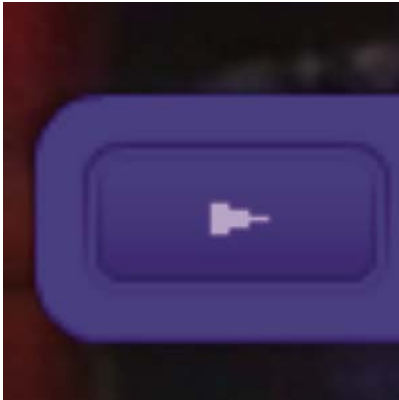
Um die Hardwarebeschleunigung für die Vollbildmodus-Unterstützung nutzen zu können, muss Ihr Computer mit einer DirectX 7-kompatiblen Grafikkarte mit mindestens 4 MB VRAM (Video RAM) ausgestattet sein. Diese Hardware-Unterstützung ist in Windows 2000 oder Mac OS X 10.2 sowie höheren Versionen dieser Betriebssysteme verfügbar. Direct X® enthält APIs, die eine Schnittstelle zwischen der Software und der Videohardware bilden und unter anderem zwei- und dreidimensionale Grafiken beschleunigen.

Zur Nutzung der Hardwarebeschleunigung müssen Sie zudem den Vollbildmodus mit einer der folgenden Methoden aufrufen:

- Vollbildmodus-Schaltfläche in einer FLVPlayback-Skin
- FullScreenButton-Videosteuerung
- ActionScript-Methode `enterFullScreenDisplayState()` Weitere Informationen finden Sie unter „[Verwenden von enterFullScreenDisplayState\(\)](#)“ auf Seite 151.

Wenn Sie den Vollbildmodus aufrufen, indem Sie die Eigenschaft `Stage.displayState` auf den Wert `StageDisplayState.FULLSCREEN` einstellen, verwendet FLVPlayback keine Hardwarebeschleunigung, selbst wenn Videohardware und VRAM verfügbar sind.

Die Verwendung der Hardwarebeschleunigung für die Vollbildmodus-Unterstützung bewirkt unter anderem, dass die FLVPlayback-Skins zusammen mit dem Video-Player und der Videodatei skaliert werden. Die folgende Abbildung zeigt die Auswirkungen des Vollbildmodus mit Hardwarebeschleunigung auf die FLVPlayback-Skin. Ein Detail ist in voller Auflösung hervorgehoben.



*Vollbildmodus auf einem Bildschirm mit 1600 x 1200 für ein Video mit 320 x 240 Pixel*

Diese Abbildung zeigt das Resultat bei Verwendung des Vollbildmodus auf einem Bildschirm mit 1600 x 1200 für eine Videodatei mit einer Breite von 320 und einer Höhe von 240 (dies sind die FLVPlayback-Standardabmessungen). Die Verzerrung der Skin fällt bei FLV-Dateien mit geringeren Abmessungen oder auf einem größeren Bildschirm stärker aus. Umgekehrt tritt bei größeren FLV-Dateien oder auf kleineren Bildschirmen eine geringere Verzerrung auf. Wenn Sie beispielsweise statt 640 x 480 einen Bildschirm mit 1600 x 1200 verwenden, wird die Skin größer, sieht aber weniger verzerrt aus.

Über die Eigenschaft `skinScaleMaximum` können Sie die Skalierung der FLVPlayback-Skin einschränken. Der Standardwert lautet 4,0 (400 %). Das Einschränken der Skin-Skalierung erfordert jedoch eine Kombination aus Hardware und Software zum Skalieren der FLV-Datei. Dies kann sich negativ auf die Leistung von großen FLV-Dateien auswirken, die mit einer hohen Bitrate kodiert sind. Wenn das Video groß ist (zum Beispiel 640 Pixel breit oder breiter, 480 Pixel hoch oder höher), sollten Sie die Eigenschaft `skinScaleMaximum` nicht auf einen niedrigen Wert setzen, da es sonst auf großen Bildschirmen zu deutlichen Leistungsproblemen kommen kann. Mithilfe der Eigenschaft `skinScaleMaximum` können Sie ein angemessenes Verhältnis zwischen Leistung und Qualität sowie der Darstellung einer großen Skin erzielen.

## Beenden des Vollbildmodus

Zum Beenden des Vollbildmodus klicken Sie noch einmal auf die Vollbild-Schaltfläche oder drücken Sie die Esc-Taste.

Beim Festlegen der folgenden Eigenschaften und Aufrufen der folgenden Methoden können Änderungen am Layout auftreten, die dazu führen, dass die FLVPlayback-Komponente den Vollbildmodus beendet: `height`, `registrationHeight`, `registrationWidth`, `registrationX`, `registrationY`, `scaleX`, `scaleY`, `width`, `x`, `y`, `setScale()` oder `setSize()`.

Wenn Sie die Eigenschaften `align` oder `scaleMode` festlegen, setzt FLVPlayback sie auf die Werte `center` bzw. `maintainAspectRatio`, bis der Vollbildmodus beendet wird.

Außerdem beendet Flash den Vollbildmodus, wenn Sie den Wert der Eigenschaft `fullScreenTakeOver` von `true` in `false` ändern, während der Vollbildmodus mit Hardwarebeschleunigung verwendet wird.

## Layout für das Abspielen mehrerer Videodateien

Die FLVPlayback-Komponente in ActionScript 3.0 verfügt über die Eigenschaft `align`, die festlegt, ob die Videodatei bei einer Größenänderung zentriert oder oben, unten, links oder rechts in der Komponente positioniert wird. Zusätzlich zu den Eigenschaften `x`, `y`, `width` und `height` verfügt die ActionScript 3.0-Komponente auch über die Eigenschaften `registrationX`, `registrationY`, `registrationWidth` und `registrationHeight`. Diese Eigenschaften entsprechen anfänglich den Eigenschaften `x`, `y`, `width` und `height`. Beim Laden nachfolgender Videodateien werden diese durch das automatische Layout nicht geändert, sodass die neue Videodatei an derselben Position zentriert werden kann. Wenn `scaleMode = VideoScaleMode.MAINTAIN_ASPECT_RATIO`, können nachfolgende FLV-Dateien in die ursprünglichen Abmessungen der Komponente eingepasst werden, anstatt Breite und Höhe der Komponente zu ändern.

## Automatisches Abspielen progressiv heruntergeladener Videodateien

Beim Laden einer Videodatei, die progressiv heruntergeladen wird, beginnt FLVPlayback erst dann mit dem Abspielen der Videodatei, wenn ein ausreichender Anteil der Datei heruntergeladen wurde, sodass die Videodatei von Anfang bis Ende abgespielt werden kann.

Wenn die Videodatei schon abgespielt werden soll, bevor ein ausreichender Anteil heruntergeladen wurde, rufen Sie die Methode `play()` ohne Parameter auf.

Wenn Sie in den Wartezustand zurückkehren möchten, bis ein ausreichender Teil der Videodatei heruntergeladen wurde, rufen Sie die Methode `pause()` und dann die Methode `playWhenEnoughDownloaded()` auf.

## Verwenden von Cue-Points

Ein Cue-Point ist ein Punkt, an dem der Video-Player ein `cuePoint`-Ereignis auslöst, während eine Videodatei abgespielt wird. Sie können einer FLV-Datei an den Stellen Cue-Points hinzufügen, an denen für andere Elemente auf der Webseite eine Aktion ausgeführt werden soll. Möglicherweise möchten Sie an einer bestimmten Stelle des Videos Text oder eine Grafik anzeigen, oder das Video mit einer Flash-Animation synchronisieren, oder das Abspielen der FLV-Datei unterbrechen, zu einem anderen Punkt spulen oder zu einer anderen FLV-Datei wechseln. Mit Cue-Points erhalten Sie in Ihrem ActionScript-Code Steuerungsmöglichkeiten und Sie können Punkte in der FLV-Datei mit anderen Aktionen auf der Webseite synchronisieren.

Es gibt drei Arten von Cue-Points: Navigation, Ereignis und ActionScript. Die Cue-Points für Navigation und Ereignisse werden auch als *eingebettete* Cue-Points bezeichnet, da sie in den FLV-Dateisteam und in das Metadatenpaket der FLV-Datei eingebettet sind.

Ein *Navigations-Cue-Point* ermöglicht die Suche nach einem bestimmten Bild in der FLV-Datei, da er ein Schlüsselbild in der FLV-Datei erstellt, das sich so nah wie möglich bei der von Ihnen angegebenen Zeit befindet. Ein *Schlüsselbild* ist ein Datensegment, das sich zwischen den Einzelbildern im FLV-Dateisteam befindet. Wenn Sie einen Vor- oder Rücklauf zu einem Navigations-Cue-Point ausführen, spult die Komponente bis zum Schlüsselbild vor oder zurück und startet das Ereignis `cuePoint`.

Ein *Ereignis-Cue-Point* ermöglicht es Ihnen, einen Zeitpunkt in der FLV-Datei mit einem externen Ereignis auf der Webseite zu synchronisieren. Das Ereignis `cuePoint` tritt genau zur angegebenen Zeit ein. Cue-Points für die Navigation und für Ereignisse können mithilfe des Videoimportassistenten oder mit Flash Video Encoder in einer FLV-Datei eingebettet werden. Weitere Informationen zum Videoimportassistenten und zum Flash Video Encoder finden Sie in der Dokumentation *Flash verwenden* in Kapitel 16, „Arbeiten mit Video“.

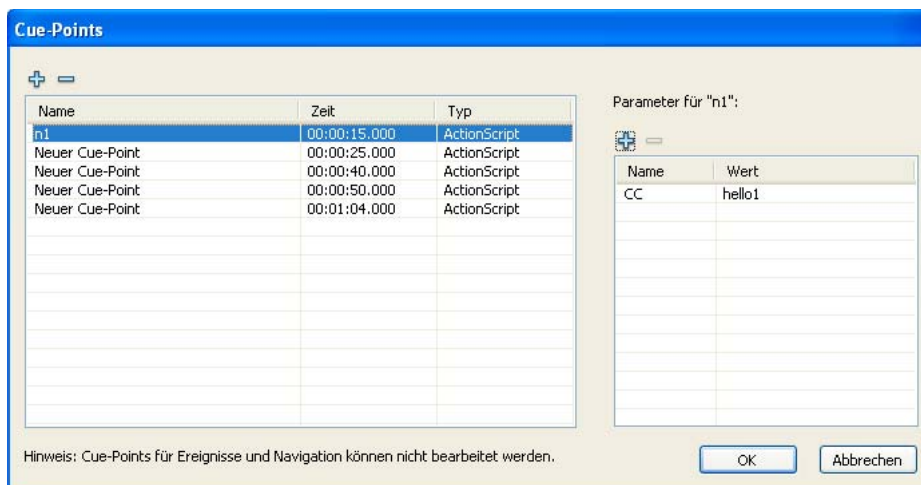
Ein *ActionScript-Cue-Point* ist ein externer Cue-Point, den Sie im Dialogfeld „Flash Video Cue-Points“ der Komponente oder mit der Methode `FLVPlayback.addASCuePoint()` hinzufügen können. Die Komponente speichert ActionScript-Cue-Points getrennt von der FLV-Datei, weshalb sie nicht so genau wie eingebettete Cue-Points sein können. ActionScript-Cue-Points sind bis auf eine Zehntelsekunde genau. Sie können die Genauigkeit von ActionScript-Cue-Points verbessern, indem Sie den Wert der Eigenschaft `playheadUpdateInterval` verringern, da die Komponente das Ereignis `cuePoint` für ActionScript-Cue-Points generiert, wenn der Abspielkopf aktualisiert wird. Weitere Informationen finden Sie im Abschnitt zur `FLVPlayback.playheadUpdateInterval`-Eigenschaft im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

In ActionScript und in den Metadaten der FLV-Datei wird ein Cue-Point als Objekt mit den folgenden Eigenschaften dargestellt: `name`, `time`, `type` und `parameters`. Die Eigenschaft `name` ist ein String, der den zugewiesenen Namen des Cue-Points enthält. Die Eigenschaft `time` ist eine Zahl, die in Stunden, Minuten, Sekunden und Millisekunden (HH:MM:SS.mmm) angibt, wann der Cue-Point eintritt. Die Eigenschaft `type` ist ein String mit dem Wert "navigation", "event" oder "actionscript", abhängig davon, welche Art Cue-Point Sie erstellt haben. Die Eigenschaft `parameters` ist ein Array von festgelegten Name/Wert-Paaren.

Wenn ein Ereignis `cuePoint` eintritt, ist das Cue-Point-Objekt im Ereignisobjekt über die Eigenschaft `info` verfügbar.

### Verwenden des Dialogfelds „Flash Video Cue-Points“

Öffnen Sie das Dialogfeld „Flash Video Cue-Points“, indem Sie im Komponenten-Inspektor auf die Zelle „Wert“ für den Parameter `cuePoints` doppelklicken. Das Dialogfeld sieht wie in der folgenden Abbildung aus:



Dialogfeld „Cue Points“

Im Dialogfeld sind eingebettete Cue-Points und ActionScript-Cue-Points aufgeführt. Sie können in diesem Dialogfeld ActionScript-Cue-Points sowie Cue-Point-Parameter hinzufügen und löschen. Außerdem können Sie eingebettete Cue-Points aktivieren oder deaktivieren. Es ist jedoch nicht möglich, in diesem Dialogfeld eingebettete Cue-Points hinzuzufügen, zu ändern oder zu löschen.

#### Hinzufügen eines ActionScript-Cue-Point:

- 1 Doppelklicken Sie im Komponenten-Inspektor auf die Zelle „Wert“ des Parameters `cuePoints`, um das Dialogfeld „Flash Video Cue-Points“ zu öffnen.
- 2 Klicken Sie oben links im Dialogfeld auf das Pluszeichen (+), um einen Standardeintrag für einen ActionScript-Cue-Point hinzuzufügen.
- 3 Klicken Sie in der Spalte „Name“ auf den Text „Neuer Cue-Point“ und geben Sie einen Namen für den Cue-Point ein.

**Verwenden der FLVPlayback-Komponente**

- 4 Klicken Sie auf den Zeitwert 00:00:00:000, um ihn zu bearbeiten, und geben Sie einen Zeitpunkt ein, zu dem der Cue-Point eintreten soll. Sie können die Zeit in Stunden, Minuten, Sekunden und Millisekunden (HH:MM:SS.mmm) angeben.

Falls mehrere Cue-Points vorhanden sind, wird der neue Cue-Point an die entsprechende Position in der chronologischen Reihenfolge der Liste verschoben.

- 5 Um einen Parameter für den ausgewählten Cue-Point hinzuzufügen, klicken Sie auf das Pluszeichen (+) über dem Bereich „Parameter“ und geben Sie Werte in die Spalten „Name“ und „Wert“ ein. Wiederholen Sie diesen Schritt für jeden Parameter.
- 6 Wenn Sie weitere ActionScript-Cue-Points hinzufügen möchten, wiederholen Sie jeweils die Schritte 2 bis 5.
- 7 Klicken Sie auf „OK“, um die Änderungen zu speichern.

**Löschen eines ActionScript-Cue-Point:**

- 1 Doppelklicken Sie im Komponenten-Inspektor auf die Zelle „Wert“ des Parameters `cuePoints`, um das Dialogfeld „Flash Video Cue-Points“ zu öffnen.
- 2 Wählen Sie den zu löschenden Cue-Point aus.
- 3 Klicken Sie oben links über der Liste der Cue-Points auf das Minuszeichen (-), um den Cue-Point zu löschen.
- 4 Wiederholen Sie die Schritte 2 und 3 für alle Cue-Points, die Sie löschen möchten.
- 5 Klicken Sie auf „OK“, um die Änderungen zu speichern.

**Aktivieren oder Deaktivieren eines eingebetteten Cue-Point in einer FLV-Datei:**

- 1 Doppelklicken Sie im Komponenten-Inspektor auf die Zelle „Wert“ des Parameters `cuePoints`, um das Dialogfeld „Flash Video Cue-Points“ zu öffnen.
- 2 Wählen Sie den Cue-Point aus, den Sie aktivieren oder deaktivieren möchten.
- 3 Klicken Sie auf den Wert in der Spalte „Typ“, um das Popupmenü zu öffnen, oder klicken Sie auf den Pfeil nach unten.
- 4 Klicken Sie auf den Namen des Cue-Point-Typs (z. B. Ereignis oder Navigation), um ihn zu aktivieren. Klicken Sie auf „Deaktiviert“, um den Cue-Point zu deaktivieren.
- 5 Klicken Sie auf „OK“, um die Änderungen zu speichern.

**Verwenden von Cue-Points mit ActionScript**

Mithilfe von ActionScript können Sie ActionScript-Cue-Points hinzufügen, auf `cuePoint`-Ereignisse warten, beliebige Cue-Points oder Cue-Points eines bestimmten Typs suchen, einen Vor- oder Rücklauf zu einem Navigations-Cue-Point ausführen, Cue-Points aktivieren oder deaktivieren, überprüfen, ob ein Cue-Point aktiviert ist und Cue-Points entfernen.

Für die Beispiele in diesem Abschnitt wird die FLV-Datei „`cuepoints.flv`“ verwendet, die die drei folgenden Cue-Points enthält:

Name	Zeit	Typ
point1	00:00:00.418	Navigation
point2	00:00:07.748	Navigation
point3	00:00:16.020	Navigation

### Hinzufügen von ActionScript-Cue-Points

Sie können einer FLV-Datei mithilfe der Methode `addASCuePoint()` ActionScript-Cue-Points hinzufügen. Im folgenden Beispiel werden der FLV-Datei zwei ActionScript-Cue-Points hinzugefügt, wenn sie zum Abspielen bereit ist. Der erste Cue-Point wird mithilfe eines Cue-Point-Objekts hinzugefügt, mit dessen Eigenschaften die Zeit, der Name und der Typ des Cue-Points festgelegt werden. Im zweiten Aufruf werden die Zeit und der Name mit den Parametern `time` und `name` der Methode festgelegt.

```
// 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");
```

Weitere Informationen finden Sie im Abschnitt zur `FLVPlayback.addASCuePoint()`-Methode im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

### Warten auf Cue-Point-Ereignisse

Durch das Ereignis `cuePoint` erhalten Sie in Ihrem ActionScript-Code die Kontrolle. Wenn im folgenden Beispiel Cue-Points eintreten, ruft der `cuePoint`-Listener eine Ereignisprozedur auf, die den Wert der Eigenschaft `playheadTime` sowie den Namen und den Typ des Cue-Points anzeigt. Verwenden Sie dieses Beispiel in Kombination mit dem Beispiel im vorigen Abschnitt, „Hinzufügen von ActionScript-Cue-Points“, um die Ergebnisse zu sehen.

```
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);
}
```

Weitere Informationen zum `cuePoint`-Ereignis finden Sie im Abschnitt zum `FLVPlayback.cuePoint`-Ereignis im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

### Suchen von Cue-Points

Mithilfe von ActionScript können Sie nach beliebigen Cue-Points, nach dem Cue-Point, der einem bestimmten Zeitwert am nächsten ist, oder nach einem Cue-Point mit einem bestimmten Namen suchen.

Die Ereignisprozedur `ready_listener()` im folgenden Beispiel ruft die Methode `findCuePoint()` auf, um den Cue-Point `ASpt1` zu suchen, und ruft dann die Methode `findNearestCuePoint()` auf, um den Navigations-Cue-Point zu finden, der dem Zeitwert des Cue-Points `ASpt1` am nächsten ist:



**Verwenden der FLVPlayback-Komponente**

```

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);
}

```

Im folgenden Beispiel sucht die Ereignisprozedur `ready_listener()` den Cue-Point `ASpt` und ruft die Methode `findNextCuePointWithName()` auf, um den nächsten Cue-Point mit demselben Namen zu finden:

```

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);
}

```

Weitere Informationen zum Suchen von Cue-Points finden Sie in den Abschnitten zu den Methoden `FLVPlayback.findCuePoint()`, `FLVPlayback.findNearestCuePoint()` und `FLVPlayback.findNextCuePointWithName()` im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

**Suchen von Navigations-Cue-Points**

Sie können einen Vor- oder Rücklauf zu einem Navigations-Cue-Point ausführen oder zum nächsten bzw. vorherigen Navigations-Cue-Point ab einem bestimmten Zeitpunkt. Im folgenden Beispiel wird die FLV-Datei „cuepoints.flv“ abgespielt und es wird ein Suchlauf zum Cue-Point bei 7.748 ausgeführt, wenn das Ereignis `ready` eintritt. Wenn das `cuePoint`-Ereignis eintritt, wird im Beispiel die Methode `seekToPrevNavCuePoint()` aufgerufen, um einen Suchlauf zum ersten Cue-Point auszuführen. Wenn dieses `cuePoint`-Ereignis eintritt, wird im Beispiel die Methode `seekToNextNavCuePoint()` aufgerufen, um einen Suchlauf zum letzten Cue-Point auszuführen, indem `eventObject.info.time` (dies ist der Zeitwert des aktuellen Cue-Points) 10 Sekunden hinzugefügt werden.

**Verwenden der FLVPlayback-Komponente**

```
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";
```

Weitere Informationen finden Sie in den Abschnitten zu den Methoden `FLVPlayback.seekToNavCuePoint()`, `FLVPlayback.seekToNextNavCuePoint()` und `FLVPlayback.seekToPrevNavCuePoint()` im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

**Aktivieren und Deaktivieren von eingebetteten Cue-Points in einer FLV-Datei**

Sie können in eine FLV-Datei eingebettete Cue-Points mithilfe der Methode `setFLVCuePointEnabled()` aktivieren oder deaktivieren. Deaktivierte Cue-Points lösen keine `cuePoint`-Ereignisse aus und werden von den Methoden `seekToCuePoint()`, `seekToNextNavCuePoint()` und `seekToPrevNavCuePoint()` nicht angesprochen. Sie können deaktivierte Cue-Points aber mit den Methoden `findCuePoint()`, `findNearestCuePoint()` und `findNextCuePointWithName()` suchen.

Mithilfe der Methode `isFLVCuePointEnabled()` können Sie testen, ob ein eingebetteter Cue-Point in einer FLV-Datei aktiviert ist oder nicht. Im folgenden Beispiel werden die eingebetteten Cue-Points `point2` und `point3` deaktiviert, wenn das Video zum Abspielen bereit ist. Wenn das erste `cuePoint`-Ereignis eintritt, testet die Ereignisprozedur, ob der Cue-Point `point3` deaktiviert ist, und aktiviert ihn, falls dies der Fall ist.

```
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");
 }
}
```

Weitere Informationen finden Sie in den Abschnitten zu den Methoden `FLVPlayback.isFLVCuePointEnabled()` und `FLVPlayback.setFLVCuePointEnabled()` im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

**Entfernen eines ActionScript-Cue-Point**

Sie können ActionScript-Cue-Points mithilfe der Methode `removeASCuePoint()` entfernen. Im folgenden Beispiel wird der Cue-Point `ASpt2` entfernt, wenn `ASpt1` eintritt.

**Verwenden der FLVPlayback-Komponente**

```
import fl.video.*;
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
my_FLVPlayback.addASCuePoint(2.02, "ASpt1");//add AS cue point
my_FLVPlayback.addASCuePoint(3.4, "ASpt2");//add 2nd Aspt
my_FLVPlayback.addEventListener(MouseEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
 trace("Cue point name is: " + eventObject.info.name);
 if (eventObject.info.name == "ASpt1") {
 my_FLVPlayback.removeASCuePoint("ASpt2");
 trace("Removed cue point ASpt2");
 }
}
```

Weitere Informationen finden Sie im `FLVPlayback.removeASCuePoint()`-Abschnitt im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

**Abspielen von mehreren Videodateien**

Sie können Videodateien in einer `FLVPlayback`-Instanz nacheinander abspielen, indem Sie einfach eine neue URL in die Eigenschaft `source` laden, wenn eine Videodatei fertig abgespielt wurde. Im folgenden ActionScript-Code wird z. B. auf das Ereignis `complete` gewartet, das eintritt, wenn eine Videodatei fertig abgespielt wurde. Wenn dieses Ereignis eintritt, setzt der Code den Namen und den Speicherort einer neuen Videodatei in die Eigenschaft `source` und ruft die Methode `play()` auf, um das neue Video abzuspielen.

```
import fl.video.*;
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/clouds.flv";
my_FLVPlayback.addEventListener(VideoEvent.COMPLETE, complete_listener);
// listen for complete event; play new FLV
function complete_listener(eventObject:VideoEvent):void {
 if (my_FLVPlayback.source == "http://www.helpexamples.com/flash/video/clouds.flv") {
 my_FLVPlayback.play("http://www.helpexamples.com/flash/video/water.flv");
 }
};
```

**Verwenden von mehreren Video-Playern**

In einer Instanz der `FLVPlayback`-Komponente lassen sich mehrere Video-Player öffnen, um mehrere Videos abzuspielen und während der Wiedergabe zwischen ihnen zu wechseln.

Sie erstellen den ursprünglichen, ersten Video-Player, wenn Sie die `FLVPlayback`-Komponente auf die Bühne ziehen. Die Komponente weist diesem ursprünglichen Video-Player die Nummer 0 zu und macht ihn zum Standard-Player. Wenn Sie einen weiteren Video-Player erstellen möchten, legen Sie einfach eine neue Nummer für die Eigenschaft `activeVideoPlayerIndex` fest. Indem Sie die Eigenschaft `activeVideoPlayerIndex` festlegen, wird der angegebene Video-Player zum *aktiven* Video-Player. Dies ist der Video-Player, auf den die Eigenschaften und Methoden der `FLVPlayback`-Klasse angewendet werden. Durch das Festlegen der Eigenschaft `activeVideoPlayerIndex` wird der jeweilige Video-Player jedoch nicht automatisch sichtbar. Damit der Video-Player sichtbar ist, legen Sie für die Eigenschaft `visibleVideoPlayerIndex` die Nummer des gewünschten Video-Players fest. Weitere Informationen zur Interaktion dieser Eigenschaften mit den Methoden und Eigenschaften der `FLVPlayback`-Klasse finden Sie in den Abschnitten zu den Eigenschaften `FLVPlayback.activeVideoPlayerIndex` und `FLVPlayback.visibleVideoPlayerIndex` im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

**Verwenden der FLVPlayback-Komponente**

Mit dem folgenden ActionScript-Code wird die Eigenschaft `source` geladen, um eine Videodatei im Standard-Player abzuspielen, und es wird ein Cue-Point hinzugefügt. Wenn das Ereignis `ready` auftritt, öffnet die Ereignisprozedur einen zweiten Video-Player, indem die Eigenschaft `activeVideoPlayerIndex` auf den Wert 1 eingestellt wird. Die Ereignisprozedur gibt eine FLV-Datei und einen Cue-Point für den zweiten Video-Player an und legt dann den Standard-Player (0) wieder als aktiven Video-Player fest.

```
/**
 * 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;
};
```

Wenn Sie beim Abspielen einer FLV-Datei zu einer anderen wechseln möchten, müssen Sie den Wechsel im ActionScript-Code vornehmen. Mithilfe von Cue-Points und `cuePoint`-Ereignissen können Sie an bestimmten Punkten in die FLV-Datei eingreifen. Im folgenden Code wird ein Listener für das Ereignis `cuePoint` erstellt und eine Prozedur aufgerufen, die den aktiven Video-Player (0) anhält, zum zweiten Player (1) wechselt und dessen FLV-Datei abspielt.

```
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);
 }
};
```

Wenn Sie einen neuen Video-Player hinzufügen, setzt die FLVPlayback-Instanz seine Eigenschaften auf die Werte des Standard-Players, mit Ausnahme der Eigenschaften `source`, `totalTime` und `isLive`, die von der FLVPlayback-Instanz immer auf die Standardwerte gesetzt werden, das heißt leerer String, 0 bzw. `false`. Die Eigenschaft `autoplay`, die für den Standard-Player standardmäßig den Wert `true` hat, wird auf `false` gesetzt. Die Eigenschaft `cuePoints` hat keinen Effekt, auch nicht für nachfolgende Ladevorgänge in den Standard-Player.

Die Methoden und Eigenschaften, mit denen Lautstärke, Position, Abmessungen, Sichtbarkeit und Benutzeroberflächen-Steuererelemente gesteuert werden, sind immer global. Ihr Verhalten wird durch das Einstellen der Eigenschaft `activeVideoPlayerIndex` nicht beeinflusst. Weitere Informationen zu diesen Methoden und Eigenschaften und den Auswirkungen der Einstellung für die `activeVideoPlayerIndex`-Eigenschaft finden Sie im Abschnitt zur `FLVPlayback.activeVideoPlayerIndex`-Eigenschaft im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#). Die verbleibenden Eigenschaften und Methoden gelten für den Video-Player, der vom Wert der Eigenschaft `activeVideoPlayerIndex` angegeben wird.

Eigenschaften und Methoden, die Abmessungen steuern, *interagieren* jedoch mit der Eigenschaft `visibleVideoPlayerIndex`. Weitere Informationen finden Sie im Abschnitt zur `FLVPlayback.visibleVideoPlayerIndex`-Eigenschaft im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

## Streamen von FLV-Dateien von einem Flash Media Server

Die Anforderungen für das Streamen von FLV-Dateien von einem Flash Media Server sind davon abhängig, ob die native Bandbreitenerkennung bei Ihrem Flash Video Streaming Service-Anbieter verfügbar ist. Eine native Bandbreitenerkennung ist in den Streaming-Server integriert und führt zu einer höheren Leistung. Fragen Sie bei Ihrem Anbieter nach, ob die native Bandbreitenerkennung verfügbar ist.

Verwenden Sie eine URL wie `rtmp://my_servername/my_application/stream.flv`, um auf Ihre FLV-Dateien auf dem Flash Media Server zuzugreifen.

Wenn Sie einen Live-Stream vom Flash Media Server abspielen, müssen Sie die FLVPlayback-Eigenschaft `isLive` auf den Wert `true` einstellen. Weitere Informationen finden Sie im Abschnitt zur FLVPlayback.isLive-Eigenschaft im [ActionScript 3.0 Referenzhandbuch für Flash Professional](#).

Weitere Informationen zum Verwalten von Flash Media Server und zum Einrichten von Live-Streams finden Sie in der Flash Media Server-Dokumentation unter [www.adobe.com/support/documentation/en/flashmediaserver/](http://www.adobe.com/support/documentation/en/flashmediaserver/).

## Native Bandbreitenerkennung oder keine Bandbreitenerkennung

Die `NCManagerNative`-Klasse ist eine Unterklasse der `NCManager`-Klasse. Sie unterstützt die native Bandbreitenerkennung, die einige Anbieter des Flash Video Streaming Service eventuell unterstützen. Bei Verwendung von `NCManagerNative` sind keine besonderen Dateien auf dem Flash Media Server erforderlich. `NCManagerNative` ermöglicht auch die Verbindung mit jeder Version von Flash Media Server, ohne eine Datei „main.asc“, falls keine Bandbreitenerkennung erforderlich ist.

Um `NCManagerNative` anstelle der Standardklasse `NCManager` zu verwenden, fügen Sie im ersten Bild Ihrer FLA-Datei die folgenden Codezeilen hinzu:

```
import fl.video*;
VideoPlayer.incManagerClass = fl.video.NCManagerNative;
```

## Nicht native Bandbreitenerkennung

Falls Ihr Flash Video Streaming Service-Anbieter keine native Bandbreitenerkennung zur Verfügung stellt, Sie jedoch Bandbreitenerkennung benötigen, müssen Sie Ihrer Flash Media Server FLV-Anwendung die Datei „main.asc“ hinzufügen. Sie finden die Datei „main.asc“ online unter [www.adobe.com/go/learn\\_fl\\_samples\\_de](http://www.adobe.com/go/learn_fl_samples_de). Sie befindet sich in der Datei „Samples.zip“ im folgenden Verzeichnis: `Samples\ComponentsAS2\FLVPlayback`.

### Einrichten von Flash Media Server zum Streamen von FLV-Dateien

- 1 Erstellen Sie in Ihrem Flash Media Server-Anwendungsordner einen Ordner und nennen Sie ihn z. B. **my\_application**.
- 2 Kopieren Sie die Datei „main.asc“ in den Ordner „my\_application“.
- 3 Erstellen Sie im Ordner „my\_application“ den Ordner **streams**.
- 4 Erstellen Sie im Ordner „streams“ den Ordner **\_definst\_**.
- 5 Legen Sie Ihre FLV-Dateien im Ordner **\_definst\_** ab.

## Anpassen der FLVPlayback-Komponente

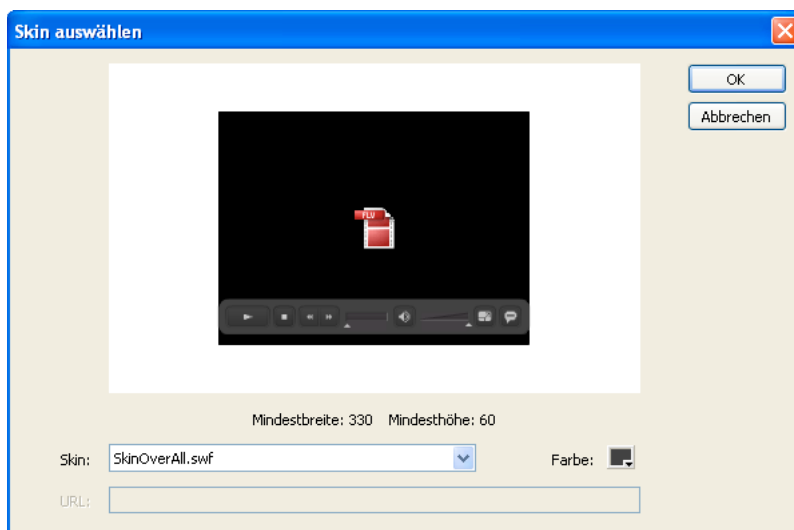
In diesem Abschnitt wird beschrieben, wie Sie die FLVPlayback-Komponente individuell anpassen. Die meisten Methoden zum Anpassen anderer Komponenten können jedoch mit der FLVPlayback-Komponente nicht verwendet werden. Sie können die FLVPlayback-Komponente nur mit den in diesem Abschnitt beschriebenen Verfahren individuell anpassen.

Zum Anpassen der FLVPlayback-Komponente stehen die folgenden Methoden zur Verfügung: Sie können eine vordefinierte Skin auswählen, die FLVPlayback Custom UI-Komponenten individuell mit einer Skin versehen oder eine neue Skin erstellen. Das Verhalten von Skins lässt sich mit den FLVPlayback-Eigenschaften modifizieren.

**Hinweis:** Sie müssen die Skin-SWF-Datei zusammen mit der Anwendungs-SWF-Datei auf den Webserver hochladen, damit die Skin mit der FLVPlayback-Komponente verwendet werden kann.

### Auswählen einer vordefinierten Skin

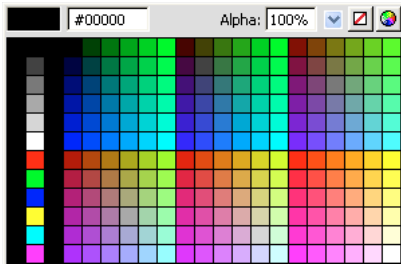
Sie können eine Skin für die FLVPlayback-Komponente auswählen, indem Sie im Komponenteninspektor auf die Zelle `wert` für den Parameter `skin` klicken. Klicken Sie dann auf das Lupensymbol, um das Dialogfeld „Skin auswählen“ zu öffnen, in dem Sie eine Skin auswählen oder eine URL eingeben können, die den Speicherort einer Skin-SWF-Datei angibt.



FLVPlayback-Dialogfeld „Skin auswählen“

Skins, die im Popupmenü „Skins“ aufgelistet sind, befinden sich im Flash-Anwendungsordner unter /Flash Configuration/FLVPlayback Skins/ActionScript 3.0. Sie können neue Skins in diesem Dialogfeld zur Verfügung stellen, indem Sie sie erstellen und die SWF-Datei im Ordner ablegen. Der Name wird im Popupmenü mit der Dateierweiterung „.swf“ angezeigt. Weitere Informationen zum Erstellen eines Skin-Satzes finden Sie unter „[Erstellen einer neuen Skin](#)“ auf Seite 169.

Für Skins, die Sie durch Festlegen der Eigenschaft `skin` zuweisen (entweder beim Authoring durch Festlegen des `skin`-Parameters oder zur Laufzeit mithilfe von ActionScript), können Sie unabhängig von der Auswahl der Skin Farben und Alphawerte (Transparenz) zuweisen. Um beim Authoring Farb- und Alphawerte zuzuweisen, öffnen Sie die Farbauswahl im Dialogfeld „Skin auswählen“, wie in der folgenden Abbildung gezeigt.



Farbauswahl im Dialogfeld „Skin auswählen“

Um die Farbe auszuwählen, klicken Sie auf ein Farbfeld oder geben Sie den numerischen Wert der Farbe in das Textfeld ein. Zum Festlegen des Alphawerts verwenden Sie den Schieberegler oder geben Sie einen Prozentwert in das Textfeld „Alpha“ ein.

Während der Laufzeit können Sie Farb- und Alphawerte zuweisen, indem Sie die Eigenschaften `skinBackgroundColor` und `skinBackgroundAlpha` festlegen. Stellen Sie die Eigenschaft `skinBackgroundColor` auf einen 0xRRGGBB-Wert ein (Rot, Grün, Blau). Stellen Sie die `skinBackgroundAlpha`-Eigenschaft auf einen Wert zwischen 0,0 und 1,0 ein. Das folgende Beispiel stellt `skinBackgroundColor` auf 0xFF0000 (Rot) und `skinBackgroundAlpha` auf 0,5 ein.

```
my_FLVPlybk.skinBackgroundColor = 0xFF0000;
my_FLVPlybk.skinBackgroundAlpha = .5;
```

Die Standardwerte sind die zuletzt vom Benutzer ausgewählten Werte.

Wenn Sie der FLVPlayback-Komponente Skins zuweisen möchten, indem Sie die FLV Playback Custom UI-Komponenten verwenden, wählen Sie im Pop-upmenü „Aus“.

## Separates Zuweisen von Skins für FLV Playback Custom UI-Komponenten

Die FLV Playback Custom UI-Komponenten ermöglichen Ihnen das individuelle Anpassen der FLVPlayback-Steuerungen in Ihrer FLA-Datei. Die Ergebnisse können Sie bei der Vorschau Ihrer Webseite überprüfen. Diese Komponenten wurden jedoch nicht für die Skalierung konzipiert. Sie sollten einen Movieclip und seinen Inhalt so bearbeiten, dass sie eine bestimmte Größe haben. Aus diesem Grund ist es am besten, wenn die FLVPlayback-Komponente in der gewünschten Größe auf der Bühne vorhanden ist, wobei die Eigenschaft `scaleMode` auf `exactFit` eingestellt ist.

Ziehen Sie zunächst einfach die gewünschten FLV Playback Custom UI-Komponenten aus dem Bedienfeld „Komponenten“, positionieren Sie sie auf der Bühne und benennen Sie die einzelnen Instanzen.

Diese Komponenten können ohne ActionScript-Code funktionieren. Wenn Sie sie auf derselben Zeitleiste und im selben Bild wie die FLVPlayback-Komponente platzieren und keine Skin in der Komponente festgelegt ist, stellt die FLVPlayback-Komponente automatisch eine Verbindung mit ihnen her. Wenn sich dagegen mehrere FLVPlayback-Komponenten auf der Bühne befinden oder wenn die benutzerdefinierte Steuerung und die FLVPlayback-Instanz sich nicht auf derselben Zeitleiste befinden, ist ActionScript erforderlich.

Wenn sich die Komponenten auf der Bühne befinden, können Sie sie wie jedes andere Symbol bearbeiten. Beim Öffnen der Komponenten erkennen Sie, dass sie sich jeweils etwas voneinander unterscheiden.



## Button-Komponenten

Die einzelnen Button-Komponenten weisen ähnliche Strukturen auf. Die verfügbaren Schaltflächen sind BackButton, ForwardButton, MuteButton, PauseButton, PlayButton, PlayPauseButton und StopButton. Die meisten verfügen über einen einzelnen Movieclip in Bild 1, der den Namen „placeholder\_mc“ hat. Dies ist normalerweise, aber nicht zwingend, eine Instanz der Schaltfläche im normalen Zustand. In Bild 2 befinden sich vier Movieclips auf der Bühne für jeden Anzeigestatus: normal, Over, Down und deaktiviert. (Zur Laufzeit geht die Komponente nie tatsächlich zu Bild 2; diese Movieclips wurden hier abgelegt, um die Bearbeitung zu erleichtern und um sie in die SWF-Datei zu laden, ohne dass das Kontrollkästchen „In erstes Bild exportieren“ im Dialogfeld „Symboleigenschaften“ ausgewählt werden muss. Die Option „Export für ActionScript“ muss aber trotzdem ausgewählt werden.)

Wenn Sie einer Schaltfläche eine Skin zuweisen möchten, bearbeiten Sie einfach diese Movieclips. Sie können dabei sowohl die Größe als auch das Erscheinungsbild ändern.

In Bild 1 erscheint normalerweise ActionScript-Code. Diesen brauchen Sie nicht zu ändern. Der Code stoppt nur den Abspielkopf in Bild 1 und legt fest, welcher Movieclip für welchen Status verwendet wird.

### Schaltflächen PlayPauseButton, MuteButton, FullScreenButton und CaptionButton

Die Schaltflächen PlayPauseButton, MuteButton, FullScreenButton und CaptionButton unterscheiden sich von den anderen Schaltflächen: Sie verfügen nur über ein Bild mit zwei Ebenen und kein Skript. In diesem Bild gibt es zwei Schaltflächen übereinander: für PlayPauseButton die Abspielen- und Pause-Schaltflächen, für MuteButton die Schaltflächen „Ton aus“ und „Ton ein“, für FullScreenButton die Schaltflächen „Vollbild ein“ und „Vollbild aus“ und für CaptionButton die Schaltflächen „Untertitel ein“ und „Untertitel aus“. Um diese Schaltflächen mit einer Skin zu versehen, weisen Sie jeder dieser beiden internen Schaltflächen eine Skin zu, wie unter [„Separates Zuweisen von Skins für FLV Playback Custom UI-Komponenten“](#) auf Seite 164 beschrieben. Es sind keine weiteren Aktionen erforderlich.

Die CaptionButton-Schaltfläche ist für die FLVPlaybackCaptioning-Komponente vorgesehen und muss an diese Komponente angefügt werden, nicht an die FLVPlayback-Komponente.

### Schaltflächen BackButton und ForwardButton

Die Schaltflächen „BackButton“ und „ForwardButton“ unterscheiden sich ebenfalls von anderen Schaltflächen. In Bild 2 verfügen sie über zusätzliche Movieclips, die Sie als Rahmen um eine oder beide Schaltflächen verwenden können. Diese Movieclips sind nicht erforderlich und haben keine besondere Bedeutung, sie stellen lediglich eine Gestaltungsmöglichkeit dar. Ziehen Sie sie einfach aus dem Bedienfeld „Bibliothek“ auf die Bühne und platzieren Sie sie an der gewünschten Stelle. Wenn Sie diese Movieclips nicht verwenden möchten, ignorieren Sie sie oder löschen Sie sie aus dem Bedienfeld „Bibliothek“.

Die meisten Schaltflächen basieren auf einem gemeinsamen Satz von Movieclips, sodass Sie das Erscheinungsbild aller Schaltflächen in einem Vorgang ändern können. Es ist jedoch auch möglich, diese gemeinsamen Movieclips zu ersetzen, sodass die einzelnen Schaltflächen unterschiedlich aussehen.

## BufferingBar-Komponente

Die BufferingBar-Komponente ist einfach: Sie besteht aus einer Animation, die sichtbar wird, wenn die Komponente in den Zwischenspeicherstatus wechselt. Es ist kein besonderer ActionScript-Code für die Konfiguration erforderlich. Standardmäßig handelt es sich um einen schräg gestreiften Balken, dessen Muster sich von links nach rechts bewegt, sodass ein Dreheffekt entsteht.

Obwohl die Zwischenspeicher-Leisten in den Skin-SWF-Dateien die Skalierung im 9-teiligen Segmentraster verwenden, da sie zur Laufzeit skaliert werden müssen, kann die FLV Custom UI-Komponente BufferingBar *keine* Skalierung im 9-teiligen Segmentraster verwenden, da sie über verschachtelte Movieclips verfügt. Wenn Sie die BufferingBar-Komponente breiter oder höher gestalten möchten, sollten Sie den Inhalt ändern, anstatt sie zu skalieren.

## SeekBar- und VolumeBar-Komponenten

Die SeekBar- und VolumeBar-Komponenten ähneln sich, obwohl sie unterschiedliche Funktionen haben. Beide Komponenten verfügen über Griffe, verwenden dieselben Mechanismen zur Griffverfolgung und unterstützen verschachtelte Clips, um den Fortschritt nachzuverfolgen.

In vielen Situationen geht der ActionScript-Code in der FLVPlayback-Komponente davon aus, dass der Registrierungspunkt (auch *Ursprung* oder *Nullpunkt* genannt) der SeekBar- oder VolumeBar-Komponente sich oben links im Inhalt befindet, deshalb ist es wichtig, dass Sie sich an diese Konvention halten. Andernfalls kann es zu Problemen mit Griffen und mit Fortschritt-Movieclips kommen.

Obwohl die Vor-/Rücklaufleisten in den Skin-SWF-Dateien die Skalierung im 9-teiligen Segmentraster verwenden, da sie zur Laufzeit skaliert werden müssen, kann die FLV Custom UI-Komponente SeekBar *keine* Skalierung im 9-teiligen Segmentraster verwenden, da sie über verschachtelte Movieclips verfügt. Wenn Sie die SeekBar-Komponente breiter oder höher gestalten möchten, sollten Sie den Inhalt ändern, anstatt sie zu skalieren.

### Handle

Eine Instanz des Handle-Movieclips befindet sich in Bild 2. Wie bei den BackButton- und ForwardButton-Komponenten geht die Komponente nie tatsächlich zu Bild 2; diese Movieclips wurden hier abgelegt, um die Bearbeitung zu erleichtern und um sie in die SWF-Datei zu laden, ohne dass das Kontrollkästchen „In erstes Bild exportieren“ im Dialogfeld „Symboleigenschaften“ ausgewählt werden muss. Die Option „Export für ActionScript“ muss aber trotzdem ausgewählt werden.

Vielleicht ist Ihnen im Hintergrund des Handle-Movieclips ein Rechteck mit dem Alphawert 0 aufgefallen. Dieses Rechteck vergrößert den Kollisionsbereich des Griffs, sodass dieser mit dem Mauszeiger leichter zu treffen ist, ohne dass das Erscheinungsbild geändert werden muss (ähnlich wie bei Schaltflächen). Da der Griff zur Laufzeit dynamisch erstellt wird, muss er ein Movieclip (keine Schaltfläche) sein. Dieses Rechteck mit dem Alphawert 0 ist aus keinem anderen Grund erforderlich, und im Allgemeinen können Sie das Innere des Griffs durch jedes beliebige Bild ersetzen. Der Registrierungspunkt sollte jedoch horizontal in der Mitte des Handle-Movieclips zentriert werden.

Der folgende ActionScript-Code befindet sich in Bild 1 der SeekBar-Komponente, um den Griff zu verwalten:

```
stop();
handleLinkageID = "SeekBarHandle";
handleLeftMargin = 2;
handleRightMargin = 2;
handleY = 11;
```

Der Aufruf der Funktion `stop()` ist wegen des Inhalts von Bild 2 erforderlich.

In der zweiten Zeile wird angegeben, welches Symbol als Griff verwendet wird. Es sollte nicht nötig sein, dieses zu ändern, wenn Sie nur die Handle-Movieclip-Instanz in Bild 2 bearbeiten. Zur Laufzeit erstellt die FLVPlayback-Komponente eine Instanz des angegebenen Movieclips auf der Bühne, und zwar als gleichgeordnete Instanz der Bar-Instanz. Dies bedeutet, dass beiden Komponenten derselbe Movieclip übergeordnet ist. Wenn sich die Bar-Instanz also auf der Stammebene befindet, muss sich der Griff ebenfalls auf der Stammebene befinden.

Die Variable `handleLeftMargin` legt die ursprüngliche Position des Griffs (0 %) fest, und die Variable `handleRightMargin` legt fest, wo sich der Griff am Schluss befindet (100 %). Die Zahlen geben den Offset vom linken und rechten Ende der Leiste an. Positive Zahlen geben die Grenzen innerhalb der Leiste, negative die Grenzen außerhalb der Leiste an. Diese Offsets geben an, wo sich der Griff – basierend auf dem Registrierungspunkt – befinden kann. Wenn Sie den Registrierungspunkt in der Mitte des Griffs positionieren, stehen die linke und rechte Seite des Griffs an den äußeren Positionen über die Ränder. Ein SeekBar-Movieclip muss über einen Registrierungspunkt in der oberen linken Ecke verfügen, damit der Inhalt korrekt funktioniert.

**Verwenden der FLVPlayback-Komponente**

Die Variable `handleY` legt die *y*-Position des Griffes relativ zur Bar-Instanz fest. Dies basiert auf dem Registrierungspunkt des Movieclips. Der Registrierungspunkt desselben Griffes befindet sich an der Spitze des Dreiecks, damit er relativ zum sichtbaren Teil platziert wird, unabhängig vom unsichtbaren Rechteck des Kollisionsbereichs. Der Bar-Movieclip muss seinen Registrierungspunkt in der oberen linken Ecke haben, damit der Inhalt korrekt funktioniert.

Mit diesen Beschränkungen kann der Griff für ein Bar-Steuerelement, das auf (100, 100) festgelegt ist und 100 Pixel breit ist, im Bereich von 102 bis 198 (horizontal) liegen und vertikal bei 111 bleiben. Wenn Sie Werte für `handleLeftMargin` und `handleRightMargin` zu -2 und `handleY` zu -11 ändern, kann der Griff im Bereich 98 bis 202 (horizontal) liegen und vertikal bei 89 bleiben.

**Movieclips für Fortschritt und Füllung**

Die SeekBar-Komponente verfügt über einen *Fortschritts*-Movieclip, die VolumeBar-Komponente über einen *Füllungs*-Movieclip. In der Praxis kann eine SeekBar- oder VolumeBar-Komponente allerdings einen beliebigen, keinen oder beide dieser Movieclips haben. Die Struktur dieser Leisten ist dieselbe, ihr Verhalten ist ähnlich, es werden jedoch unterschiedliche Werte angezeigt. Ein Fortschritts-Movieclip wird gefüllt, wenn die FLV-Datei heruntergeladen wird (dies ist nur bei HTTP-Downloads sinnvoll, beim Streaming von FMS ist die Leiste immer voll); ein Füllungs-Movieclip wird gefüllt, während sich der Griff von links nach rechts bewegt.

Die FLVPlayback-Komponente findet diese Movieclip-Instanzen, indem sie nach bestimmten Instanznamen sucht; deshalb benötigt die Fortschritts-Movieclip-Instanz den Bar-Movieclip als übergeordnete Instanz und muss den Instanznamen „`progress_mc`“ haben. Die Füllungs-Movieclip-Instanz benötigt den Instanznamen „`fullness_mc`“.

Sie können die Fortschritts- und Füllungs-Movieclips mit oder ohne darin verschachtelte Movieclip-Instanz „`fill_mc`“ festlegen. Der VolumeBar-Movieclip „`fullness_mc`“ weist die Methode *mit* dem Movieclip „`fill_mc`“ auf und der SeekBar-Movieclip „`progress_mc`“ weist die Methode *ohne* den Movieclip „`fill_mc`“ auf.

Die Methode *mit* dem darin verschachtelten Movieclip „`fill_mc`“ ist nützlich, wenn Sie eine Füllung benötigen, die nicht skaliert werden kann, ohne dass das Erscheinungsbild verzerrt wird.

Im VolumeBar-Movieclip „`fullness_mc`“ ist die darin verschachtelte Movieclip-Instanz „`fill_mc`“ maskiert. Sie können sie entweder beim Erstellen des Movieclips maskieren, oder zur Laufzeit wird eine Maske dynamisch erstellt. Wenn Sie sie mit einem Movieclip maskieren, nennen Sie die Instanz **mask\_mc** und richten Sie sie so ein, dass „`fill_mc`“ so erscheint, als ob der Prozentwert 100 % erreicht wäre. Wenn Sie „`fill_mc`“ nicht maskieren, erhält die dynamisch erstellte Maske eine rechteckige Form und dieselbe Größe wie „`fill_mc`“ bei 100 %.

Der Movieclip „`fill_mc`“ wird auf eine von zwei Arten mit der Maske eingeblendet, abhängig davon, ob „`fill_mc.slideReveal`“ den Wert `true` oder `false` hat.

Wenn „`fill_mc.slideReveal`“ mit `true` belegt ist, wird „`fill_mc`“ von links nach rechts bewegt, um durch die Maske eingeblendet zu werden. Bei 0 % befindet sich der Movieclip ganz links, sodass kein Teil durch die Maske sichtbar ist. Mit zunehmendem Prozentwert bewegt er sich nach rechts bis zum Wert 100 %, wo er sich dann wieder auf der Position befindet, an der er auf der Bühne erstellt wurde.

Wenn „`fill_mc.slideReveal`“ mit dem Wert `false` belegt ist oder nicht definiert ist (das Standardverhalten), wird die Größe der Maske von links nach rechts geändert, um mehr von „`fill_mc`“ anzuzeigen. Beim Wert 0 % wird die Maske horizontal auf 05 skaliert und bei zunehmendem Prozentwert wird `scaleX` vergrößert, bis bei 100 % „`fill_mc`“ vollständig zu sehen ist. Dies ist nicht unbedingt gleichbedeutend mit `scaleX = 100`, da „`mask_mc`“ beim Erstellen möglicherweise skaliert wurde.

Die Methode *ohne* „`fill_mc`“ ist einfacher als die Methode *mit* „`fill_mc`“, allerdings verzerrt sie die Füllung horizontal. Wenn Sie diese Verzerrung vermeiden möchten, müssen Sie „`fill_mc`“ verwenden. Die SeekBar-Komponente „`progress_mc`“ veranschaulicht diese Methode.

**Verwenden der FLVPlayback-Komponente**

Der Fortschritts- oder Füllungs-Movieclip wird basierend auf dem Prozentwert horizontal skaliert. Bei 0 % hat `scaleX` für die Instanz den Wert 0, sodass sie nicht sichtbar ist. Bei zunehmendem Prozentwert wird `scaleX` angepasst, bis der Clip bei 100 % dieselbe Größe wie beim Erstellen auf der Bühne hat. Auch dies ist nicht unbedingt gleichbedeutend mit `scaleX = 100`, da die Clip-Instanz beim Erstellen möglicherweise skaliert wurde.

**Verbinden von FLV Playback Custom UI-Komponenten**

Wenn Sie die benutzerdefinierten UI-Komponenten auf derselben Zeitleiste und im selben Bild wie die FLVPlayback-Komponente platzieren und die `skin`-Eigenschaft nicht festgelegt ist, stellt FLVPlayback automatisch eine Verbindung mit ihnen her, ohne dass ActionScript erforderlich ist.

Wenn sich dagegen mehrere FLVPlayback-Komponenten auf der Bühne befinden oder wenn die benutzerdefinierte Steuerung und FLVPlayback sich nicht auf derselben Zeitleiste befinden, müssen Sie ActionScript-Code schreiben, um die benutzerdefinierten UI-Komponenten mit Ihrer Instanz der FLVPlayback-Komponente zu verbinden. Dazu benennen Sie zunächst die FLVPlayback-Instanz und weisen dann Ihre Instanzen der FLV Playback Custom UI-Komponenten mithilfe von ActionScript den entsprechenden FLVPlayback-Eigenschaften zu. Im folgenden Beispiel heißt die FLVPlayback-Instanz „`my_FLVPlybk`“, die Namen der FLVPlayback-Eigenschaften stehen jeweils hinter dem Punkt (`.`), und die FLV Playback Custom UI-Instanzen stehen rechts neben dem Gleichheitszeichen (`=`):

```
//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;
```

Mit den folgenden Schritten werden die Steuerelemente StopButton, PlayPauseButton, MuteButton und SeekBar erstellt:

- 1 Ziehen Sie eine FLVPlayback-Komponente auf die Bühne und geben Sie ihr den Instanznamen **my\_FLVPlybk**.
- 2 Belegen Sie den Parameter `source` im Komponenten-Inspektor mit **<http://www.helpexamples.com/flash/video/cuepoints.flv>**.
- 3 Wählen Sie für den Skin-Parameter den Wert „Aus“.
- 4 Ziehen Sie Instanzen von StopButton, PlayPauseButton und MuteButton auf die Bühne und platzieren Sie sie auf der FLVPlayback-Instanz, links übereinander angeordnet. Geben Sie den einzelnen Schaltflächen im Eigenschafteninspektor Instanznamen (z. B. **my\_stopbtn**, **my\_plypausbtn** und **my\_mutebtn**).
- 5 Öffnen Sie im Bedienfeld „Bibliothek“ den Ordner „FLVPlayback Skins“ und öffnen Sie darin den Ordner „SquareButton“.
- 6 Wählen Sie den Movieclip „SquareBgDown“ aus und doppelklicken Sie darauf, um ihn auf der Bühne zu öffnen.
- 7 Klicken Sie mit der rechten Maustaste (Windows) bzw. bei gedrückter Ctrl-Taste (Macintosh), wählen Sie im Kontextmenü den Befehl „Alles auswählen“ und löschen Sie das Symbol.
- 8 Wählen Sie das Ellipsenwerkzeug aus, zeichnen Sie an derselben Stelle ein Oval und legen Sie die Füllung als blau (**#0033FF**) fest.
- 9 Stellen Sie im Eigenschafteninspektor die Breite (B:) auf **40** und die Höhe (H:) auf **20** ein. Legen Sie die X-Koordinate (X:) auf **0,0** und die Y-Koordinate (Y:) ebenfalls auf **0,0** fest.
- 10 Wiederholen Sie die Schritte 6 bis 8 für SquareBgNormal, ändern Sie jedoch die Füllung zu gelb (**#FFFF00**).

- 11 Wiederholen Sie die Schritte 6 bis 8 für SquareBgOver, ändern Sie jedoch die Füllung zu grün (#006600).
- 12 Bearbeiten Sie die Movieclips für die verschiedenen Symbole innerhalb der Schaltflächen (PauseIcon, PlayIcon, MuteOnIcon, MuteOffIcon und StopIcon). Sie finden diese Movieclips im Bedienfeld „Bibliothek“ unter „FLV Playback Skins/Bezeichnung Button/Assets“, wobei *Bezeichnung* für den Namen der Schaltfläche steht, z. B. Play, Pause usw. Führen Sie jeweils die folgenden Schritte aus:
- Wählen Sie die Option „Alles auswählen“.
  - Ändern Sie die Farbe zu rot (#FF0000).
  - Skalieren Sie auf 300 %.
  - Ändern Sie die X:-Position des Inhalts in 7.0, um die horizontale Platzierung des Symbols in jedem Schaltflächenstatus zu modifizieren.
- Hinweis: Indem Sie die Position auf diese Art ändern, müssen Sie nicht jeden Schaltflächenstatus öffnen und die Movieclip-Instanz des Symbols verschieben.*
- 13 Klicken Sie auf den blauen Zurück-Pfeil über der Zeitleiste, um zu Szene 1, Bild 1 zurückzukehren.
- 14 Ziehen Sie eine SeekBar-Komponente auf die Bühne und platzieren Sie sie rechts unten in der FLVPlayback-Instanz.
- 15 Doppelklicken Sie im Bedienfeld „Bibliothek“ auf die SeekBar-Komponente, um sie auf der Bühne zu öffnen.
- 16 Skalieren Sie sie auf 400 %.
- 17 Wählen Sie den Umriss aus und legen Sie die Farbe auf Rot (#FF0000) fest.
- 18 Doppelklicken Sie im Ordner „FLVPlayback Skins/Seek Bar“ auf „SeekBarProgress“ und legen Sie die Farbe auf Gelb (#FFFF00) fest.
- 19 Doppelklicken Sie im Ordner „FLVPlayback Skins/Seek Bar“ auf „SeekBarHandle“ und legen Sie die Farbe auf Rot (#FF0000) fest.
- 20 Klicken Sie auf den blauen Zurück-Pfeil über der Zeitleiste, um zu Szene 1, Bild 1 zurückzukehren.
- 21 Wählen Sie die SeekBar-Instanz auf der Bühne aus und nennen Sie die Instanz **my\_seekbar**.
- 22 Fügen Sie im Bedienfeld „Aktionen“ in Bild 1 der Hauptzeitleiste eine `Import`-Anweisung für die Videoklassen ein und weisen Sie die Namen der Schaltflächen und der Vor-/Rücklaufleiste den entsprechenden FLVPlayback-Eigenschaften zu, wie im folgenden Beispiel gezeigt:
- ```
import fl.video.*;
my_FLVPlybk.stopButton = my_stopbbtn;
my_FLVPlybk.playPauseButton = my_plypausbbtn;
my_FLVPlybk.muteButton = my_mutebbtn;
my_FLVPlybk.seekBar = my_seekbar;
```
- 23 Drücken Sie Strg+Eingabe, um den Film zu testen.

Erstellen einer neuen Skin

Am besten kopieren Sie eine der Skindateien, die in Flash enthalten sind, und verwenden sie als Ausgangsbasis, um eine neue Skin-SWF-Datei zu erstellen. Die FLA-Dateien für diese Skins finden Sie im Flash-Anwendungsordner unter „Configuration/FLVPlayback Skins/FLA/ActionScript 3.0“. Damit die fertige Skin-SWF-Datei als Option im Dialogfeld „Skin auswählen“ verfügbar ist, legen Sie sie im Ordner „Configuration/FLVPlayback Skins/ActionScript 3.0“ im Flash-Anwendungsordner oder im lokalen Benutzerordner „Configuration/FLVPlayback Skins/ActionScript 3.0“ ab.

Verwenden der FLVPlayback-Komponente

Da die Farbe einer Skin unabhängig von der Auswahl der Skin festgelegt werden kann, müssen Sie die FLA-Datei nicht bearbeiten, um die Farbe zu ändern. Wenn Sie eine Skin mit einer bestimmten Farbe erstellen und nicht möchten, dass die Farbe im Dialogfeld „Skin auswählen“ bearbeitet werden kann, verwenden Sie `this.border_mc.colorMe = false;` im ActionScript-Code der Skin-FLA-Datei. Informationen zum Festlegen der Farbe einer Skin finden Sie unter „[Auswählen einer vordefinierten Skin](#)“ auf Seite 163.

Wenn Sie die FLA-Dateien der installierten Flash-Skins betrachten, kommen Ihnen bestimmte Elemente auf der Bühne möglicherweise überflüssig vor, vieler dieser Elemente werden jedoch in Pfadenebenen platziert. In der Live-Vorschau mit dem 9-teiligen Segmentraster können Sie rasch feststellen, was tatsächlich in der SWF-Datei angezeigt wird.

In den folgenden Abschnitten werden komplexere Anpassungsvorgänge und Änderungen an den SeekBar-, BufferingBar- und VolumeBar-Movieclips beschrieben.

Verwenden des Skin-Layouts

Wenn Sie die FLA-Datei einer Flash-Skin öffnen, sind die Movieclips der Skin auf der Hauptzeitleiste angeordnet. Diese Clips und der ActionScript-Code im selben Bild legen die Anordnung der Steuerelemente zur Laufzeit fest.

Zwar sieht die Layout-Ebene weitgehend so aus wie die Skin zur Laufzeit, der Inhalt dieser Ebene ist zur Laufzeit jedoch nicht sichtbar. Der Clip wird nur verwendet, um die Platzierung der Steuerelemente zu berechnen. Die anderen Steuerelemente auf der Bühne werden zur Laufzeit verwendet.

Die Layout-Ebene enthält einen Platzhalter für die FLVPlayback-Komponente namens „video_mc“. Alle anderen Steuerelemente werden in Relation zu „video_mc“ angeordnet. Wenn Sie mit einer der FLA-Dateien von Flash beginnen und die Größe der Steuerelemente ändern, können Sie das Layout wahrscheinlich durch Verschieben dieser Platzhalterclips korrigieren.

Jeder Platzhalterclip hat einen spezifischen Instanznamen. Die Namen der Platzhalterclips lauten `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` und `volumeBarHandle_mc`. Das Element, das beim Auswählen einer Skinfarbe neu eingefärbt wird, ist „border_mc“.

Welcher Clip für ein Steuerelement verwendet wird, ist nicht wichtig. Im Allgemeinen wird für Schaltflächen der Clip für den normalen Status verwendet. Für die anderen Steuerelemente wird der Clip für das jeweilige Steuerelement verwendet, allerdings nur der Überschaubarkeit halber. Von Bedeutung sind nur die horizontale bzw. vertikale Position (*x* bzw. *y*) sowie die Höhe und die Breite des Platzhalters.

Neben den Standardsteuerelementen können Sie auch beliebig viele zusätzliche Clips verwenden. Die einzige Anforderung für diese Clips ist, dass für ihre Bibliothekssymbole im Dialogfeld „Verknüpfung“ die Option „Export für ActionScript“ aktiviert ist. Die benutzerdefinierten Clips in der Layout-Ebene können einen beliebigen Instanznamen haben, sofern er nicht mit einem der oben aufgeführten reservierten Instanznamen identisch ist. Ein Instanzname ist nur erforderlich, um ActionScript zum Bestimmen des Layouts für den Clip festzulegen.

Der Clip „border_mc“ ist ein Sonderfall. Wenn Sie die Eigenschaft `FLVPlayback.skinAutoHide` auf den Wert `true` einstellen, wird die Skin angezeigt, wenn sich der Mauszeiger über dem Clip „border_mc“ befindet. Dies ist wichtig für Skins, die außerhalb der Grenzen des Video-Players erscheinen. Informationen zur Eigenschaft `skinAutoHide` finden Sie unter „[Ändern des Skinverhaltens](#)“ auf Seite 174.

In den FLA-Dateien von Flash wird „border_mc“ für den Farbhintergrund und für den Rahmen um die Schaltflächen „Forward“ (Vor) und „Back“ (Zurück) verwendet.

Der Clip „border_mc“ ist auch der Teil der Skin, dessen Alpha- und Farbwerte mit den Eigenschaften `skinBackgroundAlpha` und `skinBackgroundColor` geändert werden. Damit Farbe und Alphawert angepasst werden können, muss der ActionScript-Code in der FLA-Datei der Skin Folgendes enthalten:

Verwenden der FLVPlayback-Komponente

```
border_mc.colorMe = true;
```

ActionScript und Skin-Layout

Der folgende ActionScript-Code gilt im Allgemeinen für alle Steuerelemente. Für einige Steuerelemente wird zusätzliches Verhalten durch besonderen ActionScript-Code definiert; dieser wird im Abschnitt für das jeweilige Steuerelement beschrieben.

Der ursprüngliche ActionScript-Code ist ein großer Abschnitt, der die Klassennamen für jeden Status der einzelnen Komponenten definiert. Diese Klassennamen werden in der Datei „SkinOverAll.fla“ aufgeführt. Für die Pause- und Abspielen-Schaltflächen sieht der Code zum Beispiel wie folgt aus:

```
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";
```

Die Klassennamen haben keine externen Klassendateien; sie werden nur im Dialogfeld „Verknüpfung“ für alle Movieclips in der Bibliothek angegeben.

In der ActionScript 2.0-Komponente hat es sich um Movieclips auf der Bühne gehandelt, die zur Laufzeit verwendet wurden. In der ActionScript 3.0-Komponente befinden sich diese Movieclips zwar immer noch in der FLA-Datei, jedoch nur, damit die Bearbeitung bequemer ist. Sie befinden sich nun alle auf Pfadenebenen und werden nicht exportiert. Alle Skin-Elemente in der Bibliothek sind für den Export in das erste Bild eingestellt und werden dynamisch erstellt, wie zum Beispiel mit dem folgenden Code:

```
new fl.video.skin.PauseButtonDisabled();
```

Auf diesen Abschnitt folgt ActionScript-Code, der die maximale Breite und Höhe für die Skin definiert. Im Dialogfeld „Skin auswählen“ werden diese Werte angezeigt. Zur Laufzeit werden sie verwendet, um zu verhindern, dass die Skin unter die Mindestgröße skaliert wird. Wenn Sie keine Mindestgröße festlegen möchten, lassen Sie den Wert undefiniert oder geben Sie einen Wert kleiner oder gleich null ein.

```
// 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;
```

Auf jeden Platzhalter können die folgenden Eigenschaften angewendet werden:

| Eigenschaft | Beschreibung |
|--------------|---|
| anchorLeft | Boolean. Positioniert das Steuerelement relativ zur linken Seite der FLVPlayback-Instanz. Der Standardwert ist <code>true</code> , es sei denn, <code>anchorRight</code> wurde ausdrücklich auf <code>true</code> festgelegt – in diesem Fall ist der Standardwert <code>false</code> . |
| anchorRight | Boolean. Positioniert das Steuerelement relativ zur rechten Seite der FLVPlayback-Instanz. Der Standardwert ist <code>false</code> . |
| anchorBottom | Boolean. Positioniert das Steuerelement relativ zum unteren Rand der FLVPlayback-Instanz. Der Standardwert ist <code>true</code> , es sei denn, <code>anchorTop</code> wurde ausdrücklich auf <code>true</code> festgelegt – in diesem Fall ist der Standardwert <code>false</code> . |
| anchorTop | Boolean. Positioniert das Steuerelement relativ zum oberen Rand der FLVPlayback-Instanz. Der Standardwert ist <code>false</code> . |

Wenn die Eigenschaften `anchorLeft` und `anchorRight` beide mit `true` belegt werden, wird das Steuerelement zur Laufzeit horizontal skaliert. Wenn die Eigenschaften `anchorTop` und `anchorBottom` beide mit `true` belegt werden, wird das Steuerelement zur Laufzeit vertikal skaliert.

Die Auswirkungen dieser Eigenschaften können Sie anhand der Flash-Skins erkennen. Nur die Steuerelemente `BufferingBar` und `SeekBar` werden skaliert. Sie befinden sich übereinander und die Eigenschaften `anchorLeft` und `anchorRight` wurden für beide Elemente jeweils auf `true` eingestellt. Alle Steuerelemente links von `BufferingBar` und `SeekBar` haben für die Eigenschaft `anchorLeft` den Wert `true`, für alle Steuerelemente auf der rechten Seite wurde `anchorRight` mit `true` belegt. Bei allen Steuerelementen wurde die Eigenschaft `anchorBottom` auf `true` eingestellt.

Sie können versuchen, den Movieclip „`layout_mc`“ auf der Layout-Ebene zu bearbeiten, um eine Skin zu erstellen, bei der sich die Steuerelemente nicht unten, sondern oben befinden. Dazu müssen Sie die Steuerelemente nur nach oben (relativ zu `video_mc`) verschieben und die Eigenschaft `anchorTop` für alle Elemente auf `true` einstellen.

Zwischenspeichern-Leiste

Die Zwischenspeichern-Leiste hat zwei Movieclips: `bufferingBar_mc` und `bufferingBarFill_mc`. Die Position der beiden Clips auf der Bühne in Relation zu den anderen Clips ist wichtig, da diese relative Positionierung beibehalten wird. Die Zwischenspeichern-Leiste benötigt zwei verschiedene Clips, da die Komponente „`bufferingBar_mc`“ skaliert, nicht jedoch „`bufferingBarFill_mc`“.

Auf den Movieclip „`bufferingBar_mc`“ wird die Skalierung im 9-teiligen Segmentraster angewendet, der Rahmen wird beim Skalieren also nicht verzerrt. Der Movieclip „`bufferingBarFill_mc`“ ist besonders breit, damit er auch ohne Skalierung immer breit genug ist. Er wird zur Laufzeit automatisch maskiert, sodass nur der Bereich über dem gestreckten Clip „`bufferingBar_mc`“ sichtbar ist. Standardmäßig sind die genauen Abmessungen der Maske so eingestellt, dass innerhalb von „`bufferingBar_mc`“ der gleiche Randabstand auf der linken und rechten Seite beibehalten wird, basierend auf der Differenz zwischen den horizontalen Positionen (x) von „`bufferingBar_mc`“ und „`bufferingBarFill_mc`“. Sie können die Positionierung mithilfe von ActionScript-Code anpassen.

Wenn Ihre Zwischenspeichern-Leiste nicht skaliert werden muss oder nicht die Skalierung im 9-teiligen Segmentraster verwendet, können Sie sie einrichten wie die FLV Playback Custom UI-Komponente `BufferingBar`. Weitere Informationen finden Sie unter „[BufferingBar-Komponente](#)“ auf Seite 165.

Die Zwischenspeichern-Leiste verfügt über die folgende zusätzliche Eigenschaft:

| Eigenschaft | Beschreibung |
|--------------------------------|--|
| <code>fill_mc:MovieClip</code> | Gibt den Instanznamen der Füllung für die Zwischenspeichern-Leiste an. Der Standardwert lautet „ <code>bufferingBarFill_mc</code> “. |

Vor-/Rücklaufleiste und Lautstärkeleiste

Die Vor-/Rücklaufleiste hat ebenfalls zwei Movieclips: `seekBar_mc` und `seekBarProgress_mc`. Die Position der beiden Clips auf der Layout-Ebene in Relation zu den anderen Clips ist wichtig, da diese relative Positionierung beibehalten wird. Zwar werden beide Clips skaliert, der Clip „`seekBarProgress_mc`“ kann jedoch nicht in „`seekBar_mc`“ verschachtelt sein, da „`seekBar_mc`“ die Skalierung im 9-teiligen Segmentraster verwendet, die mit verschachtelten Movieclips nicht gut funktioniert.

Auf den Movieclip „`seekBar_mc`“ wird die Skalierung im 9-teiligen Segmentraster angewendet, der Rahmen wird beim Skalieren also nicht verzerrt. Der Clip „`seekBarProgress_mc`“ wird ebenfalls skaliert, dabei treten aber Verzerrungen auf. Die Skalierung im 9-teiligen Segmentraster wird nicht angewendet, da es sich bei diesem Clip um eine Füllung handelt, die auch bei Verzerrungen korrekt aussieht.

Verwenden der FLVPlayback-Komponente

Der Clip „seekBarProgress_mc“ funktioniert ohne einen Clip „fill_mc“, ähnlich wie ein Movieclip „progress_mc“ in den FLV Playback Custom UI-Komponenten. Anders ausgedrückt: Der Clip wird nicht maskiert und wird horizontal skaliert. Die genauen Abmessungen des Clips „seekBarProgress_mc“ bei 100 % werden durch die linken und rechten Randabstände im Clip „seekBarProgress_mc“ definiert. Diese Abmessungen sind standardmäßig gleich und basieren auf der Differenz zwischen den horizontalen Positionen (x) der Clips „seekBar_mc“ und „seekBarProgress_mc“. Sie können die Abmessungen mithilfe von ActionScript-Code im Movieclip für die Vor-/Rücklaufleiste individuell anpassen, wie im folgenden Beispiel gezeigt:

```
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;
```

Sie können diesen Code entweder auf der Zeitleiste des SeekBar-Movieclips oder zusammen mit dem übrigen ActionScript-Code auf der Hauptzeitleiste platzieren. Wenn Sie die Anpassung mit Code anstatt durch Bearbeiten des Layouts vornehmen, muss sich die Füllung nicht auf der Bühne befinden. Sie muss nur in der Bibliothek vorhanden sein und für den Export in ActionScript in Bild 1 mit dem korrekten Klassennamen eingestellt sein.

Wie bei der FLV Playback Custom UI-Komponente SeekBar können Sie einen Füllungs-Movieclip für die Vor-/Rücklaufleiste erstellen. Falls Ihre Vor-/Rücklaufleiste nicht skaliert werden muss, oder falls sie skaliert wird, aber nicht mit der Skalierung im 9-teiligen Segmentraster, könnten Sie den Clip „progress_mc“ oder „fullness_mc“ mit beliebigen Methoden der FLV Playback Custom UI-Komponenten einrichten. Weitere Informationen finden Sie unter .

Da die Lautstärkeleiste in den Flash-Skins nicht skaliert wird, ist sie genau wie die FLV Playback Custom UI-Komponente „VolumeBar“ aufgebaut. Weitere Informationen finden Sie unter „[SeekBar- und VolumeBar-Komponenten](#)“ auf Seite 166. Eine Ausnahme ist der Griff, der auf andere Weise implementiert wird.

SeekBar- und VolumeBar-Griffe

Die SeekBar- und VolumeBar-Griffe werden auf der Layout-Ebene neben der Leiste platziert. Standardmäßig werden der linke Randabstand, der rechte Randabstand und die y -Achsen-Werte durch ihre Position relativ zum Leisten-Movieclip festgelegt. Der linke Randabstand wird durch die Differenz zwischen den horizontalen Positionen (x) des Griffs und der Leiste festgelegt; der rechte Randabstand entspricht dem linken Randabstand. Sie können diese Werte mithilfe von ActionScript-Code im SeekBar- oder VolumeBar-Movieclip individuell anpassen. Im folgenden Beispiel wird derselbe ActionScript-Code wie für die FLV Playback Custom UI-Komponenten verwendet:

```
this.seekBar_mc.handleLeftMargin = 2;
this.seekBar_mc.handleRightMargin = 2;
this.seekBar_mc.handleY = 11;
```

Sie können diesen Code entweder auf der Zeitleiste des SeekBar-Movieclips oder zusammen mit dem übrigen ActionScript-Code auf der Hauptzeitleiste platzieren. Wenn Sie die Anpassung mit Code anstatt durch Bearbeiten des Layouts vornehmen, muss sich die Griffkomponente nicht auf der Bühne befinden. Sie muss nur in der Bibliothek vorhanden sein und für den Export in ActionScript in Bild 1 mit dem korrekten Klassennamen eingestellt sein.

Abgesehen von diesen Eigenschaften handelt es sich bei den Griffen um einfache Movieclips, die genau wie in den FLV Playback Custom UI-Komponenten eingerichtet werden. Beide verfügen über einen rechteckigen Hintergrund, dessen Eigenschaft `alpha` den Wert 0 hat. Dieser Hintergrund vergrößert lediglich den Kollisionsbereich und ist nicht erforderlich.

Hintergrund- und Vordergrund-Clips

Die Movieclips „chrome_mc“ und „forwardBackBorder_mc“ werden als Hintergrund-Clips implementiert.

Von den Movieclips `ForwardBackBorder`, `ForwardBorder` und `BackBorder` auf der Bühne und den Platzhalter-Schaltflächen `Forward` und `Back` befindet sich lediglich `ForwardBackBorder` *nicht* auf einer Pfadebene. Dieser Movieclip befindet sich nur in Skins, die die `Forward`- und `Back`-Schaltflächen tatsächlich verwenden.

Die einzige Anforderung für diese Clips ist, dass sie in der Bibliothek für ActionScript in Bild 1 exportiert werden müssen.

Ändern des Skinverhaltens

Mithilfe der Eigenschaften `bufferingBarHidesAndDisablesOthers` und `skinAutoHide` können Sie das Verhalten Ihrer FLVPlayback-Skin individuell anpassen.

Wenn Sie die Eigenschaft `bufferingBarHidesAndDisablesOthers` auf den Wert `true` einstellen, werden die `SeekBar`-Komponente und ihr Griff von der FLVPlayback-Komponente ausgeblendet und die `Pause`- und `Abspielen`-Schaltflächen werden deaktiviert, wenn die Komponente mit dem Zwischenspeichern beginnt. Dies kann nützlich sein, wenn eine FLV-Datei über eine langsame Verbindung von einem FMS gestreamt wird und die Eigenschaft `bufferTime` auf einen hohen Wert (z. B. 10) gesetzt wurde. In diesem Fall könnte ein ungeduldiger Benutzer versuchen, einen Suchlauf mithilfe der `Abspiel`- und `Pause`-Schaltflächen auszuführen, wodurch das Abspielen der Datei noch weiter verzögert wird. Dies können Sie verhindern, indem Sie die Eigenschaft `bufferingBarHidesAndDisablesOthers` mit dem Wert `true` belegen und das `SeekBar`-Element sowie die `Pause`- und `Abspielen`-Schaltflächen deaktivieren, wenn die Komponente mit dem Zwischenspeichern beginnt.

Die Eigenschaft `skinAutoHide` wirkt sich nur auf vordefinierte SWF-Dateien aus, nicht jedoch auf Steuerelemente, die von den FLV Playback Custom UI-Komponenten erstellt werden. Bei der Einstellung `true` blendet die FLVPlayback-Komponente die Skin aus, wenn sich der Mauszeiger nicht über dem Anzeigebereich befindet. Der Standardwert dieser Eigenschaft lautet `false`.

Verwenden einer SMIL-Datei

Zur Verarbeitung mehrerer Streams für verschiedene Bandbreiten verwendet die `VideoPlayer`-Klasse eine Hilfsklasse (`NCManager`), die einen Teilsatz von SMIL unterstützt. SMIL wird verwendet, um die Position des Videostreams, das Layout (Breite und Höhe) der FLV-Datei und die FLV-Quelldateien für die verschiedenen Bandbreiten zu identifizieren. Mithilfe von SMIL können Sie auch die Bitrate und die Dauer der FLV-Datei angeben.

Den Speicherort einer SMIL-Datei spezifizieren Sie mit dem `source`-Parameter oder der `FLVPlayback.source`-Eigenschaft (ActionScript). Weitere Informationen finden Sie unter [und im Abschnitt zur `FLVPlayback.source`-Eigenschaft im *ActionScript 3.0 Referenzhandbuch für Flash Professional*.](#)

Das folgende Beispiel zeigt eine SMIL-Datei, die FLV-Dateien mit unterschiedlichen Bandbreiten über RTMP von einem FMS streamt:

```
<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>
```

Das Tag `<head>` kann die Tags `<meta>` und `<layout>` enthalten. Das Tag `<meta>` unterstützt nur das `base`-Attribut, mit dem die URL des Streaming-Video (RTMP von einem FMS) angegeben wird.

Das Tag `<layout>` unterstützt nur das Element `root-layout`, mit dem die Attribute `height` und `width` festgelegt werden, und bestimmt deshalb die Größe des Fensters, in dem die FLV-Datei gerendert wird. Diese Attribute akzeptieren nur Pixelwerte, keine Prozentwerte.

Im Textteil der SMIL-Datei können Sie einen einzelnen Link zu einer FLV-Quelldatei einfügen. Wenn Sie mehrere Dateien für verschiedene Bandbreiten von einem FMS streamen (wie im vorigen Beispiel), können Sie alternativ auch das Tag `<switch>` verwenden, um die Quelldateien aufzulisten.

Die Tags `video` und `ref` innerhalb des Tags `<switch>` sind synonym – beide können mithilfe des Attributs `src` FLV-Dateien angeben. Darüber hinaus können Sie die Attribute `region`, `system-bitrate` und `dur` verwenden, um die Region, die erforderliche Mindestbandbreite und die Dauer der FLV-Datei anzugeben.

Im Tag `<body>` darf das Tag `<video>`, `<src>` oder `<switch>` nur einmal vorkommen.

Im folgenden Beispiel wird ein progressiver Download für eine einzelne FLV-Datei ohne Bandbreitenerkennung veranschaulicht:

```
<smil>
    <head>
        <layout>
            <root-layout width="240" height="180" />
        </layout>
    </head>
    <body>
        <video src="myvideo.flv" />
    </body>
</smil>
```

<smil>

Verfügbarkeit

Flash Professional 8.

Verwendung

```
<smil>  
...  
child tags  
...  
</smil>
```

Attribute

Keine

Untergeordnete Tags

<head>, <body>

Übergeordnetes Tag

Keine

Beschreibung

Tag der obersten Ebene, das eine SMIL-Datei kennzeichnet.

Beispiel

Im folgenden Beispiel wird eine SMIL-Datei verwendet, die drei FLV-Dateien angibt:

```
<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>

Verfügbarkeit

Flash Professional 8.

Verwendung

```
<head>  
...  
child tags  
...  
</head>
```

Attribute

Keine

Untergeordnete Tags

<meta>, <layout>

Übergeordnetes Tag

<smil>

Beschreibung

Unterstützt die Tags <meta> und <layout>, gibt die Position und das Standardlayout (Höhe und Breite) der FLV-Quelldateien an.

Beispiel

Im folgenden Beispiel wird das Stammlayout als 240 Pixel mal 180 Pixel festgelegt:

```
<head>
  <meta base="rtmp://myserver/myapp/" />
  <layout>
    <root-layout width="240" height="180" />
  </layout>
</head>
```

<meta>

Verfügbarkeit

Flash Professional 8.

Verwendung

```
<meta/>
```

Attribute

base

Untergeordnete Tags

<layout>

Übergeordnetes Tag

Keine

Beschreibung

Enthält das Attribut `base`, das den Speicherort (RTMP-URL) der FLV-Quelldateien angibt.

Beispiel

Im folgenden Beispiel wird ein Meta-Tag für einen base-Speicherort auf `myserver` verwendet:

```
<meta base="rtmp://myserver/myapp/" />
```

<layout>

Verfügbarkeit

Flash Professional 8.

Verwendung

```
<layout>  
...  
child tags  
...  
</layout>
```

Attribute

Keine

Untergeordnete Tags

```
<root-layout>
```

Übergeordnetes Tag

```
<meta>
```

Beschreibung

Gibt die Breite und Höhe der FLV-Datei an.

Beispiel

Im folgenden Beispiel wird das Layout als 240 Pixel mal 180 Pixel festgelegt:

```
<layout>  
  <root-layout width="240" height="180" />  
</layout>
```

<root-layout>

Verfügbarkeit

Flash Professional 8.

Verwendung

```
<root-layout...attributes.../>
```

Attribute

Breite, Höhe

Untergeordnete Tags

Keine

Übergeordnetes Tag

```
<layout>
```

Beschreibung

Gibt die Breite und Höhe der FLV-Datei an.

Beispiel

Im folgenden Beispiel wird das Layout als 240 Pixel mal 180 Pixel festgelegt:

```
<root-layout width="240" height="180" />
```

<body>

Verfügbarkeit

Flash Professional 8.

Verwendung

```
<body>  
...  
child tags  
...  
</body>
```

Attribute

Keine

Untergeordnete Tags

<video>, <ref>, <switch>

Übergeordnetes Tag

<smil>

Beschreibung

Enthält die Tags <video>, <ref> und <switch>, die den Namen der FLV-Quelldatei, die Mindestbandbreite und die Dauer der FLV-Datei angeben. Das Attribut `system-bitrate` wird nur unterstützt, wenn das Tag <switch> verwendet wird. Im Tag <body> ist nur eine Instanz des Tags <switch>, <video> oder <ref> zulässig.

Beispiel

Im folgenden Beispiel werden drei FLV-Dateien angegeben; zwei verwenden das Tag `video`, eine verwendet das Tag `ref`:

```
<body>  
  <switch>  
    <ref src="myvideo_cable.flv" dur="3:00.1"/>  
    <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>  
    <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1"/>  
  </switch>  
</body>
```

<video>

Verfügbarkeit

Flash Professional 8.

Verwendung

```
<video...attributes.../>
```

Attribute

src, system-bitrate, dur

Untergeordnete Tags

Keine

Übergeordnetes Tag

```
<body>
```

Beschreibung

Synonym mit dem Tag `<ref>`. Unterstützt die Attribute `src` und `dur`, die den Namen und die Dauer der FLV-Quelldatei angeben. Das Attribut `dur` unterstützt das vollständige Zeitformat (00:03:00:01) und das Kurzformat (03:00:01).

Beispiel

Im folgenden Beispiel werden die Quelle und die Dauer eines Videos festgelegt:

```
<video src="myvideo_mdm.flv" dur="3:00.1"/>
```

<ref>

Verfügbarkeit

Flash Professional 8.

Verwendung

```
<ref...attributes.../>
```

Attribute

src, system-bitrate, dur

Untergeordnete Tags

Keine

Übergeordnetes Tag

```
<body>
```

Beschreibung

Synonym mit dem Tag `<video>`. Unterstützt die Attribute `src` und `dur`, die den Namen und die Dauer der FLV-Quelldatei angeben. Das Attribut `dur` unterstützt das vollständige Zeitformat (00:03:00:01) und das Kurzformat (03:00:01).

Beispiel

Im folgenden Beispiel werden die Quelle und die Dauer eines Videos festgelegt:

```
<ref src="myvideo_cable.flv" dur="3:00.1"/>
```


<switch>

Verfügbarkeit

Flash Professional 8.

Verwendung

```
<switch>
...
child tags
...
</switch/>
```

Attribute

Keine

Untergeordnete Tags

<video>, <ref>

Übergeordnetes Tag

<body>

Beschreibung

Wird mit dem untergeordneten Tag <video> oder <ref> verwendet, um die FLV-Dateien für das Videostreaming mit mehreren Bandbreiten aufzulisten. Das Tag <switch> unterstützt das Attribut `system-bitrate`, das die Mindestbandbreite angibt, sowie die Attribute `src` und `dur`.

Beispiel

Im folgenden Beispiel werden drei FLV-Dateien angegeben; zwei verwenden das Tag `video`, eine verwendet das Tag `ref`:

```
<switch>
  <ref src="myvideo_cable.flv" dur="3:00.1"/>
  <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
  <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1" />
</switch>
```

Kapitel 7: Verwenden der FLVPlaybackCaptioning-Komponente

Mit der FLVPlayback-Komponente können Sie einen Video-Player in Ihre Adobe Flash CS5 Professional-Anwendung einbinden, um heruntergeladene Adobe Flash Video-Dateien (FLV oder F4V) und Streaming-FLV- oder -F4V-Dateien abzuspielen. Weitere Informationen zu FLVPlayback finden Sie im Kapitel „[Verwenden der FLVPlayback-Komponente](#)“ auf Seite 143.

Die FLVPlaybackCaptioning-Komponente ermöglicht Ihnen, Unterstützung für geschlossene Untertitelung (Closed Captioning) in Ihr Video einzubinden. Diese Untertitelungskomponente unterstützt den W3C-Standard „Timed Text“, ein XML-Format, und umfasst zudem die folgenden Funktionen:

Untertitelung mit eingebetteten Ereignis-Cue-Points. Sie können eingebettete Ereignis-Cue-Points in einer FLV-Datei mit XML-Daten verknüpfen, um Untertitel hinzuzufügen, anstatt eine Timed Text-XML-Datei zu verwenden.

Mehrfache FLVPlayback-Untertitelung. Für mehrere FLVPlayback-Instanzen lassen sich mehrere FLVPlaybackCaptioning-Instanzen erstellen.

Steuerung durch Ein-/Aus-Schaltflächen. Ermöglichen Sie dem Benutzer das Ein- und Ausschalten von Untertiteln oder Begleittext durch eine Schaltfläche mit Umschaltfunktion.

Arbeiten mit der FLVPlaybackCaptioning-Komponente

Die FLVPlaybackCaptioning-Komponente wird mit einer oder mehreren FLVPlayback-Komponenten zusammen verwendet. Im einfachsten Fall ziehen Sie eine FLVPlayback-Komponente auf die Bühne, ziehen eine FLVPlaybackCaptioning-Komponente auf dieselbe Bühne, geben die URL für die Untertitel an und stellen die Untertitel für die Anzeige ein. Sie können aber auch mehrere Parameter einstellen und dadurch Ihre FLVPlayback-Untertitelung anpassen.

Einfügen von Untertiteln in die FLVPlayback-Komponente

Sie können die FLVPlaybackCaptioning-Komponente in jede FLVPlayback-Komponente einfügen. Wie Sie eine FLVPlayback-Komponente in eine Anwendung einbinden, wird im Abschnitt „[Erstellen einer Anwendung mit der FLVPlayback-Komponente](#)“ auf Seite 145 erläutert.

Hinzufügen der FLVPlaybackCaptioning-Komponente mithilfe des Bedienfelds „Komponenten“:

- 1 Öffnen Sie im Bedienfeld „Komponenten“ den Ordner „Video“.
- 2 Ziehen Sie die FLVPlaybackCaptioning-Komponente auf die Bühne, auf der sich auch die FLVPlayback-Komponente befindet, die Sie untiteln möchten.


***Hinweis:** Adobe stellt zwei Dateien zur Verfügung, mit denen Sie den Umgang mit der FLVPlaybackCaptioning-Komponente schnell erlernen können: `caption_video.flv` (ein FLVPlayback-Beispiel) und `caption_video.xml` (ein Untertitel-Beispiel). Die Dateien stehen unter www.helpexamples.com/flash/video/caption_video.flv und www.helpexamples.com/flash/video/caption_video.xml zur Verfügung.*

- 3 (Optional.) Ziehen Sie die CaptionButton-Komponente auf die Bühne. Diese Komponente ermöglicht es dem Benutzer, die Untertitelung an- und auszuschalten.

Hinweis: Um die CaptionButton-Komponente zu aktivieren, müssen Sie sie auf dieselbe Bühne ziehen, auf der sich auch die Komponenten „FLVPlayback“ und „FLVPlaybackCaptioning“ befinden.

- 4 Während auf der Bühne die FLVPlaybackCaptioning-Komponente ausgewählt ist, legen Sie auf der Registerkarte „Parameter“ des Eigenschafteninspektors die folgenden erforderlichen Informationen fest:

- Setzen Sie showCaptions auf true.
- Geben Sie die Quelle (source) der Timed Text-XML-Datei an, die heruntergeladen werden soll.

 Währen Sie in Flash arbeiten, um Ihre Untertitel zu testen, sollten Sie die Eigenschaft showCaptions auf true setzen. Wenn Sie jedoch die Komponente CaptionButton einbinden, damit die Benutzer die Untertitelung an- und ausschalten kann, sollten Sie die Eigenschaft showCaptions auf false setzen.

Es stehen weitere Parameters für die Anpassung der FLVPlaybackCaptioning-Komponente zur Verfügung. Weitere Informationen finden Sie im Abschnitt „Anpassen der FLVPlaybackCaptioning-Komponente“ auf Seite 194 sowie im *ActionScript 3.0 Referenzhandbuch für Flash Professional*.

- 5 Wählen Sie „Steuerung“ > „Film testen“, um das Video zu starten.

Dynamisches Erstellen einer Instanz mithilfe von ActionScript :

- 1 Ziehen Sie die FLVPlayback-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 2 Ziehen Sie die FLVPlaybackCaptioning-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“.
- 3 Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Zeitleiste den folgenden Code hinzu:

```
import fl.video.*;
var my_FLVPlybk = new FLVPlayback();
my_FLVPlybk.x = 100;
my_FLVPlybk.y = 100;
addChild(my_FLVPlybk);
my_FLVPlybk.skin = "install_drive:/Program Files/Adobe/Adobe Flash
CS5/en/Configuration/FLVPlayback Skins/ActionScript 3.0/SkinUnderPlaySeekCaption.swf";
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/caption_video.flv";
var my_FLVPlybkcap = new FLVPlaybackCaptioning();
addChild(my_FLVPlybkcap);
my_FLVPlybkcap.source = "http://www.helpexamples.com/flash/video/caption_video.xml";
my_FLVPlybkcap.showCaptions = true;
```

- 4 Geben Sie dabei für install_drive das Laufwerk ein, auf dem Flash installiert wurde, und ändern Sie den Pfad, um den Speicherort des Ordners „Skins“ in Ihrer Installation anzugeben:

Hinweis: Wenn Sie eine FLVPlayback-Instanz mithilfe von ActionScript erstellen, müssen Sie ihr dynamisch eine Skin zuweisen, indem Sie, ebenfalls mithilfe von ActionScript-Code, die Eigenschaft „skin“ setzen. Wenn Sie eine Skin mithilfe von ActionScript anwenden, wird sie nicht automatisch mit der SWF-Datei veröffentlicht. Kopieren Sie die SWF-Datei der Skin und die Anwendungs-SWF-Datei auf Ihren Server, andernfalls ist die Skin-SWF-Datei nicht verfügbar, wenn der Benutzer die Anwendung ausführt.

Parameter der FLVPlaybackCaptioning-Komponente

Für jede Instanz der FLVPlaybackCaptioning-Komponente können Sie im Eingeschafften- oder Komponenten-Inspektor die folgenden Parameter festlegen. Es folgt eine Liste dieser Eigenschaften, jeweils mit einer kurzen Erläuterung:

autoLayout: Bestimmt, ob die FLVPlaybackCaptioning-Komponente die Größe des Untertitelungsbereichs steuert. Der Standardwert lautet `true`.

captionTargetName: Identifiziert den Instanznamen des Textfelds oder Movieclips mit den Untertiteln. Der Standardwert ist „auto“.

flvPlaybackName Identifiziert den Namen der FLVPlayback-Instanz, die Sie untertiteln möchten. Der Standardwert ist „auto“.

simpleFormatting: Beschränkt mit der Einstellung „true“ die Formatierungsanweisungen der Timed Text-XML-Datei. Der Standardwert lautet `false`.

showCaptions: Legt fest, ob Untertitel angezeigt werden. Der Standardwert lautet `true`.

source Identifiziert die Quelle der Timed Text-XML-Datei.

Ausführliche Informationen zu allen FLVPlaybackCaptioning-Parametern finden Sie im *ActionScript 3.0 Referenzhandbuch für Flash Professional*.

Festlegen des Parameters „source“

Verwenden Sie den Parameter `source`, um Name und Speicherort der Timed Text-XML-Datei anzugeben, die die Untertitel für Ihren Film enthält. Geben Sie den URL-Pfad direkt in das Feld „source“ auf der Registerkarte „Parameter“ des Komponenten-Inspektors ein.

Anzeigen von Untertiteln

Damit die Untertitel angezeigt werden, setzen Sie den Parameter `showCaptions` auf `true`.

Ausführliche Informationen zu allen Parametern der FLVPlaybackCaptioning-Komponente finden Sie im *ActionScript 3.0 Referenzhandbuch für Flash Professional*.

In den obigen Beispielen haben Sie gelernt, wie Sie die FLVPlaybackCaptioning-Komponente erstellen und aktivieren, um Untertitel anzuzeigen. Es gibt zwei Arten von Quellen, die Sie für Ihre Untertitel verwenden können: 1.) eine Timed Text-XML-Datei, die die Untertitel enthält und 2.) eine XML-Datei mit dem Text der Untertitel, die Sie mit eingebetteten Ereignis-Cue-Points verknüpfen.

Verwenden von Timed Text-Untertiteln

Die FLVPlaybackCaptioning-Komponente ermöglicht die Untertitelung der zugehörigen FLVPlayback-Komponente durch den Download einer Timed Text (TT)-XML-Datei. Weitere Informationen zum Timed Text-Format finden Sie unter den Themen Audio, Video und Timed Text auf der Website www.w3.org.

In diesem Abschnitt finden Sie einen Überblick über die unterstützten Timed Text-Tags, die erforderlichen Tags in der Untertiteldatei sowie ein Beispiel für eine TT-XML-Datei. Ausführliche Informationen zu allen unterstützten Timed Text-Tags finden Sie im Abschnitt „[Timed Text-Tags](#)“ auf Seite 186.

Die FLVPlaybackCaptioning-Komponente unterstützt Timed Text-Tags für die folgenden Aufgaben:

Kategorie	Aufgabe
Absatzformatierung	Die Absätze der Untertitel können rechtsbündig, linksbündig oder zentriert ausgerichtet werden.
Textformatierung	<ul style="list-style-type: none">• Der Schriftgrad kann absolut, in Pixel, angegeben werden oder im Deltastil (z. B. +2, -4).• Die Angabe einer Schriftart und -farbe wird unterstützt.• Text kann fett und kursiv ausgezeichnet werden.• Der Text kann im Blocksatz angezeigt werden.
Weitere Formatierungen	<ul style="list-style-type: none">• Dem Textfeld (TextField) für die Untertitel kann eine Hintergrundfarbe zugewiesen werden.• Die Hintergrundfarbe des Untertiteltextfelds kann auch auf transparent (Alpha 0) gesetzt werden.• Der Zeilenumbruch des Untertiteltextfelds kann an- und ausgeschaltet werden.

Die FLVPlaybackCaptioning-Komponente verwendet denselben Zeitcode wie die FLV-Datei. Jeder Untertitel muss über das `begin`-Attribut verfügen, das festlegt, wann der Untertitel eingeblendet wird. Falls der Untertitel kein `dur`- oder `end`-Attribut hat, wird er beim Erscheinen des nächsten Untertitels bzw. am Ende der FLV-Datei ausgeblendet.

Es folgt ein Beispiel für eine Timed Text-XML-Datei. Diese Datei (`caption_video.xml`) enthält die Untertitel für die Datei `„caption_video.flv“`. Die Dateien stehen unter www.helpexamples.com/flash/video/caption_video.flv und www.helpexamples.com/flash/video/caption_video.xml zur Verfügung.

```
<?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-Tags

Die FLVPlaybackCaptioning-Komponente unterstützt Timed Text-Tags aus Untertitelungs-XML-Dateien. Weitere Informationen über die Timed Text-Tags für Audio und Video finden Sie unter www.w3.org. Die folgende Tabelle umfasst unterstützte und nicht unterstützte TT-Tags.

Funktion	Tag/Wert	Verwendung/Beschreibung	Beispiel
Ignorierte Tags	metadata	Wird ignoriert. / Wird auf jeder Ebene des Dokuments zugelassen.	
	set	Wird ignoriert. / Wird auf jeder Ebene des Dokuments zugelassen.	
	xml:lang	Wird ignoriert.	
	xml:space	Wird ignoriert / Verhalten wird auf: xml:space="default" gesetzt.	

Funktion	Tag/Wert	Verwendung/Beschreibung	Beispiel
	layout	Wird ignoriert / einschließlich etwaiger region-Tags in einem layout-Tag-Abschnitt.	
	br-Tag	Alle Attribute und Inhalte werden ignoriert.	
Zeitliche Medienkoordination für Untertitel	begin-Attribute	Nur in p-Tags zulässig. Erforderlich für die Medienzeitbereitstellung von Captions.	<p begin="3s">
	dur-Attribute	Nur in p-Tags zulässig. Empfohlen. Falls nicht verwendet, wird der Untertitel mit der FLV-Datei beendet oder bei Beginn des nächsten Untertitels ausgeblendet.	
	end-Attribute	Nur in p-Tags zulässig. Empfohlen. Falls nicht verwendet, wird der Untertitel mit der FLV-Datei beendet oder bei Beginn des nächsten Untertitels ausgeblendet.	
Uhrzeitkoordination für Untertitel	00:03:00.1	Vollständiges Zeitformat	
	03:00.1	Teilweises Zeitformat	
	10	Abweichungszeiten ohne Einheiten repräsentieren Sekunden.	
	00:03:00:05 00:03:00:05.1 30f 30t	Nicht unterstützt. Zeitformate, die Bilder oder Ticks enthalten, werden nicht unterstützt.	
Body-Tag	body	Erforderlich. Nur das body-Tag wird unterstützt.	<body><div>...</div></body>
Inhalts-Tags	div-Tag	Zugelassen sind null oder mehr. Es wird das erste Tag verwendet.	
	p-Tag	Zugelassen sind null oder mehr.	
	span-Tag	Ein logischer Container für eine Sequenz an Textinhalteinheiten. Verschachtelte Bereiche werden nicht unterstützt. Formatierungsattribute werden unterstützt.	
	br-Tag	Dies ist ein Tag für einen ausdrücklichen Zeilenumbruch.	
Styling-Tags (Alle style-Tags werden innerhalb des p-Tags verwendet)	style	Es können ein oder mehrere style-Elemente angegeben werden. Kann als Tag und als Attribut verwendet werden. Als Tag ist ein ID-Attribut erforderlich (das Format kann im Dokument wiederverwendet werden). Unterstützung für ein oder mehrere style-Tags innerhalb des style-Tags.	

Funktion	Tag/Wert	Verwendung/Beschreibung	Beispiel
	tts:background Color	Formatierungseigenschaft, die die Hintergrundfarbe einer Region definiert. Alpha wird ignoriert, wenn es nicht auf Null gesetzt ist (alpha 0), um einen transparenten Hintergrund zu definieren. Das Farbformat ist #RRGGBBAA.	
	tts:color	Legt eine style-Eigenschaft fest, die die Vordergrundfarbe definiert. Alpha wird für keine Farbe unterstützt. Der Wert <code>transparent</code> wird als Schwarz umgesetzt.	<pre><style id="3" style="2" tts:backgroundColor="white"/> "transparent" = #00000000 (Transparent) "black" = #000000FF (Schwarz) "silver" = #C0C0C0FF (Silber) "grey" = #808080FF (Grau) "white" = #FFFFFF (Weiß) "maroon" = #800000FF (Kastanienbraun) "red" = #FF0000FF (Rot) "purple" = #800080FF (Purpur) "fuchsia" ("magenta") = #FF00FFFF (Pink) "green" = #008000FF (Dunkelgrün) "lime" = #00FF00FF (Hellgrün) "olive" = #808000FF (Olivgrün) "yellow" = #FFFF00FF (Gelb) "navy" = #000080FF (Marineblau) "blue" = #0000FFFF (Blau) "teal" = #008080FF (Blaugrün, dunkles Türkis) "aqua" ("cyan") = #00FFFFFF (Türkis)</pre>
	tts:fontFamily	Legt eine style-Eigenschaft fest, die die Schriftartfamilie oder Schriftform definiert.	<pre>"default" = _serif "monospace" = _typewriter "sansSerif" = _sans "serif" = _serif "monospaceSansSerif" = _typewriter "monospaceSerif" = _typewriter "proportionalSansSerif" = _sans</pre>
	tts:fontSize	Legt eine style-Eigenschaft fest, die den Schriftgrad definiert. Nur der erste (vertikale) Wert wird verwendet, wenn zwei Werte angegeben werden. Prozentangaben und Einheiten werden ignoriert. Es werden absolute Werte in Pixel (z. B. 12) sowie relative Größenangaben (z. B. +2) akzeptiert.	

Funktion	Tag/Wert	Verwendung/Beschreibung	Beispiel
	tts:fontStyle	Legt eine style-Eigenschaft fest, die den Schriftschnitt definiert.	"normal" "italic" (kursiv) "inherit" (geerbt)* * Das Standardverhalten; es wird der Schriftschnitt vom übergeordneten Tag geerbt.
	tts:fontWeight	Legt eine style-Eigenschaft fest, die die Schriftstärke definiert.	"normal" "bold" (fett) "inherit" (geerbt)* * Das Standardverhalten; es wird die Eigenschaft vom übergeordneten Tag geerbt.

Funktion	Tag/Wert	Verwendung/Beschreibung	Beispiel
	tts:textAlign	Legt eine style-Eigenschaft fest, die definiert, wie Text innerhalb eines Container-Blocks ausgerichtet wird.	"left" (linksbündig) "right" (rechtsbündig) "center" (zentriert) "start" ("=left") "end" ("=right") "inherit" (geerbt)* * Erbt die Eigenschaft vom übergeordneten Tag. Ist das textAlign-Tag nicht gesetzt, ist "left" der Standard.
	tts:wrapOption	Legt eine style-Eigenschaft fest, die ob auf den Inhalt des betreffenden Elements der automatische Zeilenumbruch angewendet wird. Diese Einstellung wirkt sich auf alle Absätze im Untertitelelement aus.	"wrap" (Zeilenumbruch) "noWrap" (Kein Zeilenumbruch) "inherit" (geerbt)* * Erbt die Eigenschaft vom übergeordneten Tag. Ist das wrapOption-Tag nicht gesetzt, ist "wrap" der Standard.
Nicht unterstützte Attribute	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		

Verwenden von Cue-Points bei der Untertitelung

Cue-Points erlauben Ihnen die Interaktion mit einem Video. So können Sie anhand von Cue-Points etwa den Zeitpunkt für das Abspielen einer FLV-Datei festlegen oder das Einblenden von Text zu bestimmten Zeiten im Video. Wenn Sie keine Timed Text-XML-Datei haben, die mit einer FLV-Datei verwendet werden kann, können Sie Ereignis-Cue-Points in die FLV-Datei einbetten und diese dann mit Text verknüpfen. Dieser Abschnitt informiert Sie über die Cue-Points-Standards der FLVPlaybackCaptioning-Komponente und bietet eine kurze Übersicht dazu, wie Sie in Untertiteln Cue-Points mit Text verknüpfen. Weitere Informationen über das Einbetten von Ereignis-Cue-Points mithilfe des Videoimportassistenten oder dem Flash Video Encoder finden Sie im Handbuch *Verwenden von Flash* im Kapitel 16, „Arbeiten mit Videos“.

Cue-Point-Standards für die FLVPlaybackCaptioning-Komponente

In den Metadaten der FLV-Datei wird ein Cue-Point als Objekt mit den folgenden Eigenschaften dargestellt: `name`, `time`, `type` und `parameters`. Die ActionScript-Cue-Points für FLVPlaybackCaptioning haben die folgenden Attribute:

name Die Eigenschaft `name` ist ein String, der den zugewiesenen Namen des Cue-Points enthält. Die Eigenschaft `name` muss mit dem Präfix *fl.video.caption.2.0.* beginnen, danach muss ein String folgen. Der String ist eine Serie von positiven Ganzzahlen, die jeweils hochgezählt werden, sodass ein eindeutiger Name entsteht. Das Präfix enthält die Versionsnummer, die gleichzeitig der FLVPlayback-Versionsnummer entspricht. Für Adobe Flash CS4 und höher müssen Sie die Versionsnummer auf `2.0` setzen.

time: Die Eigenschaft `time` gibt an, wann der Untertitel eingeblendet werden soll.

type Die Eigenschaft `type` ist ein String mit dem Wert "event".

parameters: Die Eigenschaft `parameters` ist ein Array, das die folgenden Namen/Wert-Paare unterstützt:

- **text:String**: Der in HTML formatierte Text für den Untertitel. Dieser Text wird direkt an die Eigenschaft `TextField.htmlText` übergeben. Die FLVPlaybackCaptioning-Komponente unterstützt eine optionale Eigenschaft, nämlich `text:n`, welche die Verwendung mehrerer Sprachspuren unterstützt. Weitere Information finden Sie im Abschnitt „[Unterstützung mehrerer Sprachspuren mit eingebetteten Cue-Points](#)“ auf Seite 193.
- **endTime:Number**: Der Zeitpunkt, zu dem ein Untertitel wieder ausgeblendet werden soll. Wenn Sie diese Eigenschaft nicht festlegen, geht die FLVPlaybackCaptioning-Komponente davon aus, dass es sich nicht um eine Zahl handelt (not a number, NaN), und der Untertitel wird angezeigt, bis die FLV-Datei vollständig abgespielt ist (d. h. wenn die FLVPlayback-Instanz das Ereignis `VideoEvent.COMPLETE` auslöst). Geben Sie die Eigenschaft `endTime:Number` in Sekunden an.
- **backgroundColor:uint**: Dieser Parameter legt die Hintergrundfarbe (`TextField.backgroundColor`) fest. Die Angabe dieser Eigenschaft ist optional.
- **backgroundColorAlpha:Boolean**: Wenn `backgroundColor` einen Alphawert von 0% aufweist, legt der Parameter `TextField.background = !backgroundColor` fest. Die Angabe dieser Eigenschaft ist optional.
- **wrapOption:Boolean**: Dieser Parameter legt den Zeilenumbruch des Textfelds (`TextField.wordWrap`) fest. Die Angabe dieser Eigenschaft ist optional.

Untertitelung für eingebettete Ereignis-Cue-Points

Wenn Sie keine Timed Text XML-Datei haben, die die Untertitel für Ihre FLV-Datei enthält, können Sie Untertitel erstellen, indem Sie eine XML-Datei, die Untertitel enthält, mit eingebetteten Ereignis-Cue-Points verknüpfen. Im folgenden XML-Beispiel wird vorausgesetzt, dass Sie die folgenden Schritte ausgeführt haben, um in Ihrem Video eingebettete Ereignis-Cue-Points zu erstellen:

- Fügen Sie die Ereignis-Cue-Points hinzu (gemäß den FLVPlaybackCaptioning-Standards) und kodieren Sie das Video.
- Ziehen Sie in Flash eine FLVPlayback-Komponente sowie eine FLVPlaybackCaptioning-Komponente auf die Bühne.
- Legen Sie die source-Eigenschaften für die FLVPlayback-Komponente und die FLVPlaybackCaptioning-Komponente fest (Speicherort der FLV-Datei und Speicherort der XML-Datei).
- Veröffentlichen Sie die Datei.

Mit dem folgenden Beispiel werden XML-Daten in den Encoder importiert:

```
<?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>
```

Die FLVPlaybackCaptioning-Komponente unterstützt auch mehrerer Sprachspuren eingebetteten Cue-Points. Weitere Information finden Sie im Abschnitt „[Unterstützung mehrerer Sprachspuren mit eingebetteten Cue-Points](#)“ auf Seite 193.

Unterstützung mehrerer Sprachspuren mit eingebetteten Cue-Points

Die FLVPlaybackCaptioning-Eigenschaft `track` unterstützt mehrerer Sprachspuren mit eingebetteten Cue-Points, vorausgesetzt die Timed Text-XML-Datei befolgt die Cue-Points-Standards für die FLVPlaybackCaptioning-Komponente. (Weitere Information finden Sie im Abschnitt „[Cue-Point-Standards für die FLVPlaybackCaptioning-Komponente](#)“ auf Seite 191.) Allerdings unterstützt die FLVPlaybackCaptioning-Komponente keine mehrfachen Sprachspuren in verschiedenen XML-Dateien. Um mehrere Sprachspuren zu definieren, setzen Sie die Eigenschaft `track` auf einen Wert ungleich 0. Wenn Sie die Eigenschaft beispielsweise auf 1 setzen (`track == 1`), wird die FLVPlaybackCaptioning-Komponente die Cue-Point-Parameter suchen. Wird eine Übereinstimmung gefunden, wird die Texteingenschaft in den Cue-Point-Parametern verwendet. Weitere Informationen finden Sie im Abschnitt zur `track`-Eigenschaft im *ActionScript 3.0 Referenzhandbuch für Flash Professional*.

Abspielen mehrerer FLV-Dateien mit Untertiteln

In einer Instanz der FLVPlayback-Komponente lassen sich mehrere Video-Player öffnen, um mehrere Videos abzuspielen und während der Wiedergabe zwischen ihnen zu wechseln. Sie können auch mit jedem Video-Player innerhalb der FLVPlayback-Komponente Untertitel verknüpfen. Weitere Informationen zum Öffnen mehrerer Video-Player finden Sie im Abschnitt „[Verwenden von mehreren Video-Playern](#)“ auf Seite 159. Um in mehreren Video-Playern Untertitel zu verwenden, erstellen Sie eine Instanz der FLVPlaybackCaptioning-Komponente für jeden `videoPlayer` und setzen die FLVPlaybackCaptioning-Eigenschaft `videoPlayerIndex` auf den entsprechenden Index. Wenn nur ein `videoPlayer` verwendet wird, hat der `videoPlayer`-Index standardmäßig den Wert 0.

Es folgt ein Codebeispiel, in dem eindeutigen Videos eine eindeutige Untertitelung zugewiesen wird. Um dieses Beispiel nachzuvollziehen, müssen Sie die erfundenen URLs durch funktionierende URLs ersetzen.

```
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");
```

Anpassen der FLVPlaybackCaptioning-Komponente

Um schnell mit der FLVPlaybackCaptioning-Komponente zu arbeiten, können Sie die FLVPlaybackCaptioning-Standards verwenden, d. h., die Untertitel werden direkt über die FLVPlayback-Komponente gelegt. Sie haben aber auch die Möglichkeit, die FLVPlaybackCaptioning-Komponente so anzupassen, dass die Untertitel neben dem Video stehen.

Der folgende Code zeigt, wie Sie dynamisch ein FLVPlayback-Objekt mit einer Schaltfläche zum Umschalten der Untertitelung erstellen:

- 1 Platzieren Sie die FLVPlayback-Komponente auf der Bühne an Position 0,0 und geben Sie den Instanznamen **player** an.
- 2 Platzieren Sie die FLVPlaybackCaptioning-Komponente auf der Bühne an Position 0,0 und geben Sie den Instanznamen **captioning** an.
- 3 Platzieren Sie die CaptionButton-Komponente auf der Bühne.
- 4 Setzen Sie die Variable `testVideoPath:String` im folgenden Beispiel auf eine FLV-Datei (mit einem absoluten oder relativen Pfad).

***Hinweis:** Im Beispielcode wurde die Variable `testVideoPath` auf das Flash-Videobeispiel `caption_video.flv` gesetzt. Tragen Sie in diese Variable den Pfad der Captioning-Video-Komponente ein, in die Sie die CaptionButton-Komponente einfügen wollen.*

- 5 Setzen Sie die Variable `testCaptioningPath:String` im folgenden Beispiel auf eine geeignete Timed Text XML-Datei (mit einem absoluten oder relativen Pfad).

***Hinweis:** Im Beispielcode wurde die Variable `testCaptioningPath` auf die Timed Text XML-Datei `caption_video.xml` gesetzt. Ändern Sie diese Variable zum Pfad der Timed Text XML-Datei, welche die Untertitel für Ihr Video enthält.*

- 6 Speichern Sie den folgenden Code als „FLVPlaybackCaptioningExample.as“ in demselben Verzeichnis wie die FLA-Datei.
- 7 Legen Sie für „DocumentClass“ in der FLA-Datei „FLVPlaybackCaptioningExample“ fest.

```
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;
        }
    }
}
```

Ausführliche Informationen zu allen FLVPlaybackCaptioning-Parametern finden Sie im *ActionScript 3.0 Referenzhandbuch für Flash Professional*.