

ACTIONSCRIPT™ 3.0- KOMPONENTEN VERWENDEN

© 2007 Adobe Systems Incorporated. Alle Rechte vorbehalten.

ActionScript™ 3.0-Komponenten verwenden

Wenn dieses Handbuch mit einer Software ausgeliefert wird, für die ein Lizenzvertrag für Endbenutzer besteht, wird dieses Handbuch sowie die darin beschriebene Software gemäß einer Lizenz bereitgestellt und darf nur in Übereinstimmung mit den Bedingungen in diesem Lizenzvertrag verwendet und kopiert werden. Kein Teil dieser Dokumentation darf, außer durch das Lizenzabkommen ausdrücklich erlaubt, ohne vorherige, schriftliche Genehmigung von Adobe Systems Incorporated reproduziert, in Datenbanken gespeichert oder in irgendeiner Form - elektronisch, mechanisch, auf Tonträger oder auf irgendeine andere Weise - übertragen werden. Der Inhalt dieses Handbuchs ist urheberrechtlich geschützt, auch wenn es nicht mit der Software ausgeliefert wird, für die ein Lizenzvertrag für Endbenutzer besteht.

Der Inhalt dieses Handbuchs dient ausschließlich Informationszwecken, kann ohne Vorankündigung verändert werden und ist nicht als Verpflichtung von Adobe Systems Incorporated anzusehen. Adobe Systems Incorporated gibt keine Gewähr oder Garantie hinsichtlich der Richtigkeit oder Genauigkeit der Angaben in dieser Dokumentation.

Denken Sie daran, dass bestehende Grafiken oder Bilder, die Sie in Projekte einfügen möchten, möglicherweise urheberrechtlich geschützt sind. Die unerlaubte Einfügung derartiger Materialien in Ihre Projekte ist unter Umständen eine Verletzung der Rechte des Urheberrechtsinhabers. Stellen Sie sicher, dass Sie alle erforderlichen Berechtigungen von dem Urheberrechtsinhaber erhalten.

Verweise auf Firmennamen in Beispielvorgaben dienen ausschließlich zu Demonstrationszwecken und verweisen nicht auf tatsächlich bestehende Organisationen.

Adobe, das Adobe-Logo, ActionScript, Flash, Flash Player und Flash Video sind Marken oder eingetragene Marken von Adobe Systems Incorporated in den USA und/oder anderen Ländern.

Macintosh ist eine Marke von Apple Inc. in den USA und/oder anderen Ländern. Windows ist eine eingetragene Marke oder eine Marke der Microsoft Corporation in den USA und/oder anderen Ländern. Alle anderen Marken sind das Eigentum der jeweiligen Rechteinhaber.

Dieses Produkt enthält Software, die von der Apache Software Foundation (<http://www.apache.org/>) entwickelt wurde. MPEG Layer-3 Audiokomprimierungstechnologie lizenziert durch Fraunhofer IIS und Thomson Multimedia (<http://www.mp3licensing.com>). Das in der Software enthaltene MP3-komprimierte Audio kann nicht für Echtzeit- oder Live-Sendungen verwendet werden. Wenn Sie einen MP3-Decoder für Echtzeit- oder Live-Übertragungen benötigen, sind Sie dafür zuständig, die entsprechende MP3-Lizenz zu erhalten. Technologie zur Sprachkomprimierung/-dekomprimierung lizenziert von Nellymoser, Inc. (www.nellymoser.com). Flash CS3 Video wird von der Videotechnologie On2 TrueMotion betrieben. © 1992-2005 On2 Technologies, Inc. Alle Rechte vorbehalten. <http://www.on2.com>. Dieses Produkt enthält Software, die von der OpenSymphony Group (<http://www.opensymphony.com/>) entwickelt wurde.



Sorenson Spark™-Technologie zur Videokomprimierung/-dekomprimierung lizenziert von Sorenson Media, Inc.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Hinweise für Endbenutzer der Regierung der Vereinigten Staaten. Kommerzieller Gegenstand. Die Software und die Dokumentation sind „Kommerzielle Gegenstände“ im Sinne von 48 C.F.R. Absatz 2.101, bestehend aus „kommerzieller Computersoftware“ bzw. „kommerzieller Computersoftware-dokumentation“ nach 48 C.F.R. Absatz 12.212 bzw. 48 C.F.R. Absatz 227.7202. In Übereinstimmung mit 48 C.F.R. Absatz 12.212 oder 48 C.F.R. Abschnitt 227.7202-1 bis 227.7202-4, werden die kommerzielle Computersoftware bzw. die kommerzielle Computersoftware-dokumentation an die Endbenutzer der Regierung der Vereinigten Staaten (a) nur als kommerzielle Gegenstände und (b) nur mit den Rechten lizenziert, die gemäß den Vertragsbedingungen allen anderen Endbenutzern gewährt werden. Nicht veröffentlichte Rechte werden gemäß dem Urheberrecht der Vereinigten Staaten vorbehalten. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. Hinsichtlich Endbenutzern der Regierung der Vereinigten Staaten erklärt sich Adobe bereit, alle geltenden Gleichbehandlungsrechte einzuhalten, einschließlich, soweit anwendbar, die Vorschriften zur Durchführungsverordnung 11246 in ihrer jeweiligen geltenden Fassung, Abschnitt 402 des „Vietnam Era Veterans Readjustment Assistance Acts“ von 1974 (38 USC 4212) und Abschnitt 503 des „Rehabilitation Acts“ von 1973 in seiner jeweiligen geltenden Fassung, und den Regelungen in 41 CFR Teile 60-1 bis 60-60, 60-250 und 60-741. Die im vorstehenden Satz enthaltenen positiven Handlungsgebote und Vorschriften werden durch die entsprechende Bezugnahme in diese Vereinbarung einbezogen.

Inhaltsverzeichnis

Einführung	11
Zielpublikum	12
Systemanforderungen	12
Informationen zur Dokumentation	12
Typografische Konventionen	13
Begriffserläuterungen	13
Weitere Informationsquellen	13
 Kapitel 1: ActionScript 3.0-Komponenten	 15
Vorteile von Komponenten	16
Komponententypen	18
Hinzufügen und Entfernen	21
Löschen von Komponenten	24
Feststellen der Version	24
ActionScript 3.0-Ereignisverarbeitungsmodell	26
Eine einfache Anwendung	27
Entwurf der Anwendung	27
Erstellen der Anwendung „Greetings“	28
Ausführen der nachfolgenden Beispiele	35
 Kapitel 2: Arbeiten mit Komponenten	 37
Komponentenarchitektur	37
FLA-basierte Komponenten von ActionScript 3.0	38
SWC-basierte Komponenten	39
Die ActionScript 3.0-Komponenten-API	40
Verwenden von Komponentendateien	41
Speicherort von Komponentendateien	41
Speicherort von Komponentenquelldateien	42
Komponentenquelldateien und Klassenpfad	42
Ändern von Komponentendateien	43
Debuggen von Komponentenanwendungen	44
Festlegen von Parametern und Eigenschaften	45
Festlegen von Komponenteneigenschaften in ActionScript	47
Die Bibliothek	48

Ändern der Komponentengröße	50
Live-Vorschau	51
Verarbeiten von Ereignissen	52
Ereignis-Listener	52
Ereignisobjekte	53
Verwenden von Anzeigelisten	54
Hinzufügen von Komponenten zur Anzeigeliste	55
Verschieben von Komponenten in der Anzeigeliste	56
Entfernen von Komponenten aus der Anzeigeliste	56
Verwenden des FocusManagers	57
Verwenden von List-basierten Komponenten	59
Verwenden von Datenprovidern	61
Erstellen von Datenprovidern	61
Verwenden des Parameters „dataProvider“	61
Verwenden von ActionScript	63
Bearbeiten von Datenprovidern	67
Verwenden von CellRenderern	71
Formatieren von Zellen	71
Benutzerdefinierte CellRenderer-Klassen	72
CellRenderer-Eigenschaften	78
CellRenderer für eine Spalte eines DataGrid-Objekts	79
CellRenderer für eine bearbeitbare Zelle	79
Verwenden von Bildern, SWF-Dateien oder Movieclips als CellRenderer	80
Barrierefreiheit von Komponenten	80
Kapitel 3: Verwenden von UI-Komponenten	83
Button-Komponente	83
Benutzerinteraktion der Button-Komponente	84
Button-Parameter	85
Erstellen von Anwendungen mit der Button-Komponente	85
CheckBox-Komponente	88
Benutzerinteraktion mit der CheckBox-Komponente	88
CheckBox-Parameter	89
Erstellen von Anwendungen mit der CheckBox-Komponente	90
ColorPicker-Komponente	92
Benutzerinteraktion mit der ColorPicker-Komponente	93
ColorPicker-Parameter	93
Erstellen von Anwendungen mit der ColorPicker-Komponente	94

ComboBox-Komponente	96
Benutzerinteraktion mit der ComboBox-Komponente	96
ComboBox-Parameter	98
Erstellen von Anwendungen mit der ComboBox-Komponente	98
DataGrid-Komponente	100
Benutzerinteraktion mit der DataGrid-Komponente	101
DataGrid-Parameter	103
Erstellen von Anwendungen mit der DataGrid-Komponente	103
Label-Komponente	108
Benutzerinteraktion mit der Label-Komponente	108
Label-Parameter	108
Erstellen von Anwendungen mit der Label-Komponente	109
List-Komponente	111
Benutzerinteraktion mit der List-Komponente	111
List-Parameter	113
Erstellen von Anwendungen mit der List-Komponente	113
NumericStepper-Komponente	117
Benutzerinteraktion mit der NumericStepper-Komponente	117
NumericStepper-Parameter	118
Erstellen von Anwendungen mit der NumericStepper-Komponente	119
ProgressBar-Komponente	121
Benutzerinteraktion mit der ProgressBar-Komponente	122
ProgressBar-Parameter	122
Erstellen von Anwendungen mit der ProgressBar-Komponente	123
RadioButton-Komponente	129
Benutzerinteraktion mit der RadioButton-Komponente	129
RadioButton-Parameter	130
Erstellen von Anwendungen mit der RadioButton-Komponente	130
ScrollPane-Komponente	133
Benutzerinteraktion mit der ScrollPane-Komponente	134
ScrollPane-Parameter	134
Erstellen von Anwendungen mit der ScrollPane-Komponente	135
Slider-Komponente	136
Benutzerinteraktion mit der Slider-Komponente	137
Slider-Parameter	138
Erstellen von Anwendungen mit der Slider-Komponente	138
TextArea-Komponente	140
Benutzerinteraktion mit der TextArea-Komponente	141
TextArea-Parameter	142
Erstellen von Anwendungen mit der TextArea-Komponente	142

TextInput-Komponente	144
Benutzerinteraktion mit der TextInput-Komponente	145
TextInput-Parameter	145
Erstellen von Anwendungen mit der TextInput-Komponente.	146
TileList-Komponente	149
Benutzerinteraktion mit der TileList-Komponente	149
TileList-Parameter	150
Erstellen von Anwendungen mit der TileList-Komponente.	151
UILoader-Komponente	153
Benutzerinteraktion mit der UILoader-Komponente	153
UILoader-Parameter	153
Erstellen von Anwendungen mit der UILoader-Komponente.	154
UIScrollBar-Komponente	155
Benutzerinteraktion mit der UIScrollBar-Komponente	155
UIScrollBar-Parameter	155
Erstellen von Anwendungen mit der UIScrollBar-Komponente	156
Kapitel 4: Anpassen von UI-Komponenten	159
Anpassung von UI-Komponenten	160
Stile	160
Arbeiten mit Stileinstellungen	161
Zugriff auf die Standardstile einer Komponente	162
Festlegen und Abrufen von Stilen für eine Komponenteninstanz	162
Festlegen von Texteigenschaften mit dem TextFormat-Objekt	163
Festlegen eines Stils für alle Instanzen einer Komponente.	163
Festlegen eines Stils für alle Komponenten	164
Skins	165
Erstellen neuer Skins	168
Erstellen einer Skin für alle Instanzen	169
Erstellen von Skins für einige Instanzen	169
Anpassen der Button-Komponente	170
Stile für die Button-Komponente	171
Skins für die Button-Komponente	172
Anpassen der CheckBox-Komponente	174
Stile für die CheckBox-Komponente	174
Skins für die CheckBox-Komponente	175
ColorPicker-Komponente anpassen	176
Stile für die ColorPicker-Komponente	176
Skins für die ColorPicker-Komponente	177

ComboBox-Komponente anpassen	178
Stile für die ComboBox-Komponente	179
Skins für die ComboBox-Komponente	180
DataGrid-Komponente anpassen	181
Stile für die DataGrid-Komponente	181
Festlegen von Stilen für einzelne Spalten	182
Festlegen von Überschriftenstilen	184
Skins für die DataGrid-Komponente	185
Label-Komponente anpassen	187
Stile für die Label-Komponente	188
Skins für die Label-Komponente	188
List-Komponente anpassen	189
Stile für die List-Komponente	189
Skins für die List-Komponente	190
NumericStepper-Komponente anpassen	192
Stile für die NumericStepper-Komponente	193
Skins für die NumericStepper-Komponente	194
ProgressBar-Komponente anpassen	195
Stile für die ProgressBar-Komponente	195
Skins für die ProgressBar-Komponente	196
RadioButton-Komponente anpassen	197
Stile für die RadioButton-Komponente	198
Skins für die RadioButton-Komponente	199
ScrollPane-Komponente anpassen	200
Stile für die ScrollPane-Komponente	201
Skins für die ScrollPane-Komponente	202
Slider-Komponente anpassen	202
Stile für die Slider-Komponente	202
Skins für die Slider-Komponente	203
TextArea-Komponente anpassen	204
Stile für die TextArea-Komponente	205
Skins für die TextArea-Komponente	206
TextInput-Komponente anpassen	207
Stile für die TextInput-Komponente	208
Skins für die TextInput-Komponente	209
TileList-Komponente anpassen	210
Stile für die TileList-Komponente	210
Skins für die TileList-Komponente	211
UINavigationController-Komponente anpassen	212
UISearchBar-Komponente anpassen	213
Stile für die UISearchBar-Komponente	213
Skins für die UISearchBar-Komponente	214

Kapitel 5: Die FLVPlayback-Komponente	217
Verwenden der FLVPlayback-Komponente.....	218
Erstellen von Anwendungen mit der FLVPlayback-Komponente.....	220
Parameter der FLVPlayback-Komponente	224
Festlegen des „source“-Parameters	224
Verwenden der Live-Vorschau	227
Vollbildunterstützung	228
Layout für das Abspielen mehrerer FLV-Dateien	228
Automatisches Abspielen progressiv heruntergeladener FLV-Dateien	228
Verwenden von Cue-Points	229
Dialogfeld „Flash Video Cue-Points“.....	231
Verwenden von ActionScript mit Cue-Points	233
Hinzufügen von ActionScript-Cue-Points	233
Abfragen von cuePoint-Ereignissen	234
Suchen von Cue-Points	234
Vor- oder Rücklauf zu Navigations-Cue-Points	235
Aktivieren und Deaktivieren von eingebetteten FLV-Datei-Cue-Points.....	236
Entfernen von ActionScript-Cue-Points.....	237
Abspielen von mehreren FLV-Dateien	237
Verwenden mehrerer Video-Player	238
Streaming von FLV-Dateien von einem Flash Media Server... ..	240
Bei nativer Bandbreitenerkennung oder nicht vorhandener Bandbreitenerkennung.....	241
Bei nicht-nativer Bandbreitenerkennung	241
Anpassen der FLVPlayback-Komponente	242
Auswahl vordefinierter Skins	243
Zuweisen von Skins für einzelne FLV Playback Custom UI-Komponenten	245
Button-Komponenten	245
BufferingBar-Komponente	247
SeekBar- und VolumeBar-Komponenten	247
Verbinden von FLV Playback Custom UI-Komponenten	250
Erstellen neuer Skins	253
Verwenden des Skin-Layouts	254
Zwischenspeichern-Leiste	257
Vor-/Rücklaufleiste und Lautstärkeleiste	258
Hintergrund- und Vordergrund-Clips	259
Ändern des Skinverhaltens	260

Verwenden von SMIL-Dateien	260
<smil>	262
<head>	263
<meta>	263
<layout>	264
<root-layout>	265
<body>	265
<video>	266
<ref>	267
<switch>	268
Kapitel 6: Die FLVPlaybackCaptioning-Komponente	269
Verwenden der FLVPlaybackCaptioning-Komponente	270
Hinzufügen von Untertiteln zur FLVPlayback-Komponente ...	270
Festlegen der Parameter für die	
FLVPlaybackCaptioning-Komponente	272
Festlegen des „source“-Parameters	273
Anzeigen von Untertiteln	273
Verwenden von Timed Text-Untertiteln	273
Verwenden von Cue-Points mit Untertiteln	275
Cue-Point-Standards für die	
FLVPlaybackCaptioning-Komponente	275
Erstellen von Untertiteln für eingebettete	
Ereignis-Cue-Points	276
Unterstützung mehrerer Sprachspuren mit	
eingebetteten Cue-Points	278
Abspielen mehrerer FLV-Dateien mit Untertiteln	278
Anpassen der FLVPlaybackCaptioning-Komponente	279
Anhang A: Timed Text-Tags	281
Index	287

Einführung

Adobe® Flash® CS3 Professional ist das 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 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.

Mit Komponenten können Sie schnell und einfach robuste Anwendungen mit einem einheitlichen Aussehen und Verhalten erstellen. In diesem Handbuch wird beschrieben, wie Sie mit ActionScript 3.0-Komponenten Anwendungen erstellen können. Im *Komponenten-Referenzhandbuch für ActionScript 3.0* werden die einzelnen Komponenten mit ihren Programmierschnittstellen (API) beschrieben.

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

Dieses Kapitel enthält die folgenden Abschnitte:

Zielpublikum	12
Systemanforderungen.....	12
Informationen zur Dokumentation	12
Typografische Konventionen.....	13
Begriffserläuterungen	13
Weitere Informationsquellen	13

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 ActionScript noch nicht vertraut sind, können Sie einem Dokument Komponenten hinzufügen, die Parameter der Komponenten im Eigenschafteninspektor oder im Komponenten-Inspektor festlegen und die zugehörigen Ereignisse im Bedienfeld „Verhalten“ verwalten. 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 in [Kapitel 2, „Arbeiten mit Komponenten“](#) auf Seite 37.

Systemanforderungen

Für das Arbeiten mit Flash-Komponenten müssen neben den Systemanforderungen für Flash keine weiteren Anforderungen erfüllt werden.

Alle SWF-Dateien, die Flash CS3-Komponenten 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. Allgemeine Vorkenntnisse in Flash und ActionScript 3.0 werden vorausgesetzt. Dokumentationen zu Flash und ähnlichen Produkten sind separat erhältlich.

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

Weitere Informationen über Flash finden Sie in den folgenden Dokumenten:

- *Flash verwenden*
- *Programmieren mit ActionScript 3.0*
- *Komponenten-Referenzhandbuch für ActionScript 3.0*

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 Informationsquellen

Zusätzlich zu diesen Handbüchern bietet Adobe regelmäßig aktualisierte Artikel, Designvorschläge und Beispiele im Adobe Developer Center und im Adobe Design Center.

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

Adobe Developer Center

Das Adobe Developer Center ist die beste Quelle für aktuellste Informationen zu ActionScript, Artikeln über realistische Anwendungsentwicklung und andere wichtige Themen. Sie finden das Developer Center unter www.adobe.com/go/flash_devcenter_de.

Adobe Design Center

Informieren Sie sich aus erster Hand über das Neueste im Bereich digitales Design und animierte Grafiken. Sehen Sie sich Arbeiten renommierter Künstler an, entdecken Sie neue Trends und erweitern Sie Ihre Fertigkeiten mit Lernprogrammen, wichtigen Arbeitsabläufen und fortschrittlichen Techniken. Etwa alle zwei Wochen sollten Sie hier vorbeischauen, um nach neuen Tutorials und Artikeln sowie neuen Elementen in der Galerie zu sehen. Sie finden das Developer Center unter www.adobe.com/go/fl_designcenter_de.

ActionScript 3.0-Komponenten

Adobe® Flash® CS3 Professional-Komponenten sind Movieclips mit Parametern. Mithilfe dieser Parameter können Sie die Gestaltung und das Verhalten der Movieclips modifizieren. Bei einer Komponente kann es sich um ein einfaches Steuerelement für die Benutzeroberfläche handeln (beispielsweise ein Optionsfeld oder ein Kontrollkästchen), sie kann aber auch Inhalte enthalten (beispielsweise eine Liste oder ein Datenraster).

Mit Komponenten können Sie schnell und einfach robuste Flash-Anwendungen mit einem einheitlichen Verhalten und Aussehen erstellen. Anstatt Schaltflächen, Kombinationsfelder und Listen selbst zu erstellen, können Sie die Flash-Komponenten verwenden, die diese Steuerelemente implementieren. Ziehen Sie sie einfach aus dem Bedienfeld „Komponenten“ in das Anwendungsdokument. Außerdem können Sie das Erscheinungsbild dieser Komponenten an Ihr Anwendungsdesign anpassen.

Diese Vorgänge lassen sich ohne umfassende ActionScript-Kenntnisse ausführen. Sie können ActionScript 3.0 aber auch verwenden, um das Verhalten einer Komponente zu modifizieren oder ein neues Verhalten zu implementieren. Jede Komponente verfügt über einen eigenen Satz von ActionScript-Methoden, -Eigenschaften und -Ereignissen, welche die *API* (Application Programming Interface) der Komponente bilden. Mit der API können Sie Komponenten erstellen und bearbeiten, während die Anwendung ausgeführt wird.

Mithilfe der API können Sie auch eigene neue Komponenten erstellen. Von der Adobe Exchange-Website unter http://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.

Die ActionScript 3.0-Komponentenarchitektur beinhaltet Klassen, auf denen alle Komponenten basieren, Skins und Stile, mit denen Sie das Aussehen der Komponente anpassen können, ein Ereignisverarbeitungsmodell, Fokusverwaltung, Implementierung von Eingabehilfen und vieles mehr.

HINWEIS

Adobe Flash CS3 enthält sowohl ActionScript 2.0-Komponenten als auch ActionScript 3.0-Komponenten. Diese beiden Komponentensätze können nicht gemischt verwendet werden. Sie dürfen für eine Anwendung jeweils nur einen Satz verwenden. Abhängig davon, ob Sie eine ActionScript 2.0- oder eine ActionScript 3.0-Datei öffnen, bietet Flash CS3 entweder ActionScript 2.0-Komponenten oder ActionScript 3.0-Komponenten an. Wenn Sie ein neues Flash CS3-Dokument erstellen, müssen Sie entweder eine Flash-Datei (ActionScript 3.0) oder eine Flash-Datei (ActionScript 2.0) auswählen. Wenn Sie ein vorhandenes Dokument öffnen, prüft Flash die Einstellungen für Veröffentlichungen, um festzustellen, welcher Komponentensatz zu verwenden ist. Informationen zu ActionScript 2.0-Komponenten finden Sie in der Dokumentation *ActionScript 2.0-Komponenten verwenden*.

Eine vollständige Liste der Flash ActionScript 3.0-Komponenten finden Sie unter „Komponententypen“ auf Seite 18.

Dieses Kapitel enthält die folgenden Abschnitte:

Vorteile von Komponenten	16
Komponententypen	18
Hinzufügen und Entfernen	21
Feststellen der Version	24
ActionScript 3.0-Ereignisverarbeitungsmodell	26
Eine einfache Anwendung	27

Vorteile von Komponenten

Mit Komponenten können Sie Design und Programmierung voneinander trennen. Entwickler erstellen mit Komponenten Funktionen, die von Designern in Anwendungen verwendet werden können. Entwickler können häufig verwendete Funktionen in Komponenten einschließen, und Designer können die Größe, die Position und das Verhalten der Komponenten anpassen, indem sie deren Parameter ändern. Das Erscheinungsbild von Komponenten kann durch das Bearbeiten ihrer grafischen Elemente oder Skins modifiziert werden.

Komponenten nutzen Grundfunktionen wie Stile, Skins und Fokusverwaltung gemeinsam. Wenn Sie einer Anwendung die erste Komponente hinzufügen, belegt diese Grundfunktionalität ungefähr 20 KB. Diese 20 KB werden auch von den weiteren Komponenten, die Sie hinzufügen, genutzt. Ihre Anwendung wird dadurch also nicht sehr viel größer.

In diesem Abschnitt werden einige Vorteile der ActionScript 3.0-Komponenten vorgestellt.

Die Leistungsfähigkeit von ActionScript 3.0 bietet eine leistungsstarke, objektorientierte Programmiersprache, die einen wichtigen Faktor bei der Entwicklung von Flash Player-Fähigkeiten darstellt. Die Sprache wurde für die Erstellung von Rich-Internet-Anwendungen mithilfe wiederverwendbaren Codes entwickelt. ActionScript 3.0 basiert auf ECMAScript, der international standardisierten Skriptsprache. Die Sprache entspricht der Sprachspezifikation ECMAScript (ECMA-262) Edition 3. Eine umfassende Einführung in ActionScript 3.0 finden Sie im Handbuch *Programmieren mit ActionScript 3.0*. Referenzinformationen zur Sprache finden Sie im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

FLA-basierte UI-Komponenten ermöglichen den unkomplizierten Zugriff auf Skins zum problemlosen Anpassen beim Authoring. 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 in [Kapitel 4, „Anpassen von UI-Komponenten“](#) auf Seite 159 und im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Neue FVLPlayback-Komponente fügt FLVPlaybackCaptioning-Komponente hinzu, zusammen mit Vollbildunterstützung, verbesserter Live-Vorschau, Skins, um Farbe und Alpha-Einstellungen hinzuzufügen, und verbesserten FLV-Download- und Layoutfunktionen.

Der Eigenschafteninspektor und der Komponenten-Inspektor ermöglichen Ihnen das Ändern von Komponentenparametern beim Authoring in Flash. Weitere Informationen hierzu finden Sie unter [„Hinzufügen und Entfernen“](#) auf Seite 21 und [„Festlegen von Parametern und Eigenschaften“](#) auf Seite 45.

Ein neues Sammlungsdialogfeld für die ComboBox-, List- und TileList-Komponenten ermöglicht das Ausfüllen der dataProvider-Eigenschaft über die Benutzeroberfläche. Weitere Informationen finden Sie unter [„Erstellen von Daten Providern“](#) auf Seite 61.

Das ActionScript 3.0-Ereignismodell ermöglicht der Anwendung, Ereignisse abzufragen und Ereignisprozeduren aufzurufen. Weitere Informationen hierzu finden Sie unter [„ActionScript 3.0-Ereignisverarbeitungsmodell“](#) auf Seite 26 und [„Verarbeiten von Ereignissen“](#) auf Seite 52.

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

Die UIComponent-Basisklassen bieten Kernmethoden, -eigenschaften und -ereignisse für die Komponenten, die sie erweitern. Alle ActionScript 3.0-UI-Komponenten übernehmen von der UIComponent-Klasse. Weitere Informationen finden Sie im Abschnitt über die UIComponent-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Die Verwendung einer SWC-Datei in den UI FLA-basierten Komponenten bietet ActionScript-Definitionen als ein Element in der Zeitleiste der Komponente, um die Kompilierung zu beschleunigen.

Die 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 [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

HINWEIS

Flash CS3 unterstützt sowohl FLA-basierte und SWC-basierte Komponenten. Weitere Informationen finden Sie unter „[Komponentenarchitektur](#)“ auf [Seite 37](#).

Komponententypen

Sie installieren die Flash-Komponenten, wenn Sie Flash CS3 installieren.

ActionScript 3.0-Komponenten enthalten die folgenden UI-Komponenten (UI = User Interface, Benutzeroberfläche):

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

Neben den UI-Komponenten gehören die folgenden Komponenten und zugehörigen Klassen zu den Flash ActionScript 3.0-Komponenten:

- 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 Macromedia® Flash® Media Server von Adobe (FMS) abzuspielen. Weitere Informationen finden Sie in [Kapitel 5, „Die FLVPlayback-Komponente“ auf Seite 217](#).
- Die FLVPlayback Custom UI-Komponenten, die FLA-basiert sind und sowohl mit der ActionScript 2.0- als auch mit der ActionScript 3.0-Version der FLVPlayback-Komponente verwendet werden können. Weitere Informationen finden Sie in [Kapitel 5, „Die FLVPlayback-Komponente“ auf Seite 217](#).
- Die FLVPlaybackCaptioning-Komponente, die die Verwendung von Closed Captioning (codierte Untertitel) für FLVPlayback ermöglicht. [Kapitel 6, „Die FLVPlaybackCaptioning-Komponente“ auf Seite 269](#)

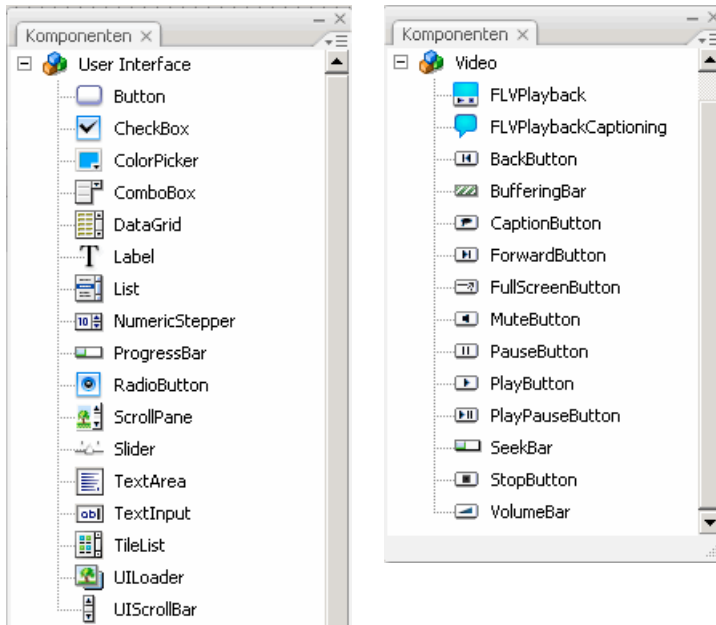
Eine vollständige Liste der ActionScript 3.0-Komponentenklassen und der zugehörigen Klassen finden Sie im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

So zeigen Sie Flash-Komponenten an:

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 UI- und Videokomponenten

Die UI- und die Videokomponenten werden getrennt angezeigt, um Platz zu sparen. Das Bedienfeld „Komponenten“ enthält alle in der Abbildung angezeigten Komponenten.

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

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

So installieren Sie 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\Flash CS3*Sprache*\Configuration\Components
 - Macintosh:
Macintosh HD:Applications:Adobe Flash CS3: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 zu Komponentendateien finden Sie unter [„Verwenden von Komponentendateien“ auf Seite 41](#).

Hinzufügen und Entfernen

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.

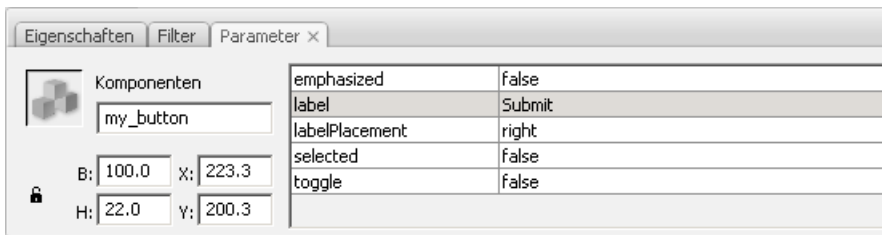
Hinzufügen von Komponenten beim Authoring

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

So fügen Sie einem Flash-Dokument mit dem Bedienfeld „Komponenten“ eine Komponente hinzu:

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. Wenn der Eigenschafteninspektor nicht angezeigt wird, wählen Sie „Fenster“ > „Eigenschaften“ > „Eigenschaften“.
5. Geben Sie im Eigenschafteninspektor einen Instanznamen für die Komponenteinstanz an.
6. Klicken Sie auf die Registerkarte „Parameter“ und legen Sie die Parameter für die Instanz fest.

In der folgenden Abbildung ist der Eigenschafteninspektor für eine Button-Komponente zu sehen.



Komponenteneinstellungen im Eigenschafteninspektor

Weitere Informationen finden Sie unter [„Festlegen von Parametern und Eigenschaften“ auf Seite 45](#).

7. Ändern Sie bei Bedarf die Größe der Komponente, indem Sie die Werte für Breite (B) und Höhe (H) ändern.

Weitere Informationen zur Größeneinstellung bestimmter Komponententypen finden Sie in [Kapitel 4, „Anpassen von UI-Komponenten“ auf Seite 159](#).

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 die Farbe und die Textformatierung einer Komponente ändern, indem Sie Stileigenschaften festlegen oder das Erscheinungsbild anpassen, indem Sie die Skins der Komponente bearbeiten. Weitere Informationen zu diesen Themen finden Sie in [Kapitel 4, „Anpassen von UI-Komponenten“ auf Seite 159](#).

Wenn Sie beim Authoring eine Komponente auf die Bühne ziehen, können Sie einfach 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 zunächst 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 [„Die Bibliothek“ auf Seite 48](#).

Sie müssen auch die Klassendatei der Komponente importieren, damit deren API in Ihrer Anwendung verfügbar ist. Komponentenklassendateien werden in *Paketen* installiert, die eine oder mehrere Klassen enthalten. Um eine Komponentenkategorie zu importieren, verwenden Sie die `import`-Anweisung und geben Sie den Paketnamen und den Klassennamen an. Die `Button`-Kategorie wird zum Beispiel mit der folgenden `import`-Anweisung importiert:

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

Informationen über die Pakete, in denen sich bestimmte Komponenten befinden, finden Sie im [Komponenten-Referenzhandbuch für ActionScript 3.0](#). Informationen zum Speicherort der Komponentenquelldateien finden Sie unter [„Verwenden von Komponentendateien“ auf Seite 41](#).

Um eine Instanz der 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();
```

Der letzte Schritt besteht darin, die statische Methode `addChild()` aufzurufen, um die Komponenteninstanz der Bühne oder dem Anwendungscontainer hinzuzufügen. Mit der folgenden Anweisung wird zum Beispiel die Instanz `aButton` hinzugefügt:

```
addChild(aButton);
```

Zu diesem Zeitpunkt können Sie die API der Komponente verwenden, um die Größe und Position der Komponente auf der Bühne dynamisch anzugeben, auf Ereignisse zu warten und Eigenschaften zum Ändern des Verhaltens festzulegen. Weitere Informationen zur API für bestimmte Komponenten finden Sie im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Weitere Informationen zur Methode `addChild()` finden Sie unter [„Verwenden von Anzeigelisten“ auf Seite 54](#).

Löschen von Komponenten

Um eine Komponenteninstanz während des Authoring von der Bühne zu löschen, wählen Sie sie aus und drücken Sie die Taste <Entf>. Damit wird die Instanz von der Bühne entfernt, die Komponente wird jedoch nicht aus der Anwendung gelöscht.

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.

So entfernen Sie eine Komponente aus einem Dokument:

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 zur Komponente gehörigen Elemente zu löschen.

Informationen zum Entfernen einer Komponente aus ihrem Container, während die Anwendung ausgeführt wird, finden Sie unter [„Entfernen von Komponenten aus der Anzeigelist“](#) auf Seite 56.

Feststellen der Version

Flash ActionScript 3.0-Komponenten verfügen über eine Versionseigenschaft, die Sie anzeigen können, falls Sie sie dem technischen Support von Adobe mitteilen müssen oder falls Sie wissen möchten, welche Version der Komponente Sie verwenden.

So zeigen Sie die Versionsnummer einer UI-Komponente an:

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 zum Beispiel 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.



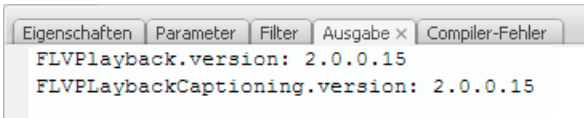
Für die FLVPlayback- und FLVPlaybackCaptioning-Komponenten müssen Sie auf den Klassennamen anstatt auf den Instanznamen zurückgreifen, da die Versionsnummer in einer Klassenkonstante gespeichert wird.

So zeigen Sie die Versionsnummer der FLVPlayback- und FLVPlaybackCaptioning-Komponenten an:

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 sollten ähnlich wie in der folgenden Abbildung im Bedienfeld „Ausgabe“ angezeigt werden.



Versionsnummern der FLVPlayback- und FLVPlaybackCaptioning-Komponenten

ActionScript 3.0- Ereignisverarbeitungsmodell

In ActionScript 3.0 wird ein Ereignisverarbeitungsmodell eingeführt, das die verschiedenen Ereignisverarbeitungsmechanismen ersetzt, die in früheren ActionScript-Versionen verwendet wurden. Das neue Ereignismodell beruht auf der DOM3-Ereignisspezifikation (Document Object Model Level 3).

Für Entwickler, die Erfahrung mit der ActionScript 2.0-Methode `addListener()` haben, ist es u. U. hilfreich, die Unterschiede zwischen dem Listener-Ereignismodell von ActionScript 2.0 und dem ActionScript 3.0-Ereignismodell herauszustellen. Nachstehend sind einige Hauptunterschiede zwischen den beiden Modellen der Ereignisverarbeitung aufgeführt:

- Um in ActionScript 2.0 Ereignis-Listener hinzuzufügen, wird in einigen Fällen `addListener()`, in anderen Fällen `addEventListener()` verwendet, während in ActionScript 3.0 immer `addEventListener()` verwendet wird.
- In ActionScript 2.0 gibt es keinen Ereignisablauf, weshalb die `addListener()`-Methode nur für das Objekt, welches das Ereignis überträgt, aufgerufen werden kann. In ActionScript 3.0 kann die `addEventListener()`-Methode dagegen für jedes Objekt, das Teil des Ereignisablaufs ist, aufgerufen werden.
- In ActionScript 2.0 können Ereignis-Listener Funktionen, Methoden oder Objekte sein, während in ActionScript 3.0 nur Funktionen oder Methoden Ereignis-Listener sein können.
- Die Syntax `on(event)` wird in ActionScript 3.0 nicht mehr unterstützt, sodass Sie einem Movieclip keinen ActionScript-Ereigniscode anhängen können. Sie können `addEventListener()` nur 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 ActionScript 3.0-Ereignisverarbeitung finden Sie im Handbuch *Programmieren mit ActionScript 3.0*. Weitere Informationen zur ActionScript 3.0-Ereignisverarbeitung für Komponenten finden Sie unter „[Verarbeiten von Ereignissen](#)“ auf Seite 52.

Eine einfache Anwendung

In diesem Abschnitt wird Schritt für Schritt beschrieben, wie Sie mithilfe von Flash-Komponenten und des Flash-Authoring-Tools 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. Im Allgemeinen werden Sie größere Anwendungen mithilfe externer Klassendateien erstellen, damit Sie Code auch für andere Klassen und Anwendungen verwenden und Ihre Anwendungen besser warten können. Weitere Informationen zur Programmierung mit ActionScript 3.0 finden Sie im Handbuch *Programmieren mit ActionScript 3.0*.

Entwurf der Anwendung

Das erste Beispiel einer ActionScript-Komponentenanwendung ist eine Variante der Standardanwendung „Hello World“; das Design ist also ziemlich schlicht:

- Die Anwendung soll „Greetings“ heißen.
- Es wird eine TextArea-Komponente verwendet, um eine Begrüßung (anfänglich „Hello World“) anzuzeigen.
- Es wird eine ColorPicker-Komponente verwendet, mit deren Hilfe Sie die Textfarbe ändern können.
- Es werden drei RadioButton-Komponenten verwendet, sodass Sie den Text klein, größer oder sehr groß anzeigen können.
- Es wird eine ComboBox-Komponente verwendet, sodass Sie aus einer Dropdownliste eine andere Begrüßung auswählen können.
- Für die Anwendung werden Komponenten aus dem Bedienfeld „Komponenten“ verwendet, und es werden Anwendungselemente mithilfe von ActionScript-Code erstellt.

Sie können jetzt damit beginnen, die Anwendung zu erstellen.

Erstellen der Anwendung „Greetings“

Mit den nachstehenden Schritten wird die Anwendung „Greetings“ erstellt, indem mithilfe des Flash-Authoring-Tools eine FLA-Datei erstellt wird, Komponenten auf der Bühne platziert werden und der Zeitleiste ActionScript-Code hinzugefügt wird.

So erstellen Sie die Anwendung „Greetings“ 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“.
Es wird ein neues Flash-Fenster angezeigt.
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. Während die TextArea-Komponente auf der Bühne ausgewählt ist, geben Sie im Fenster Eigenschaften den Instanznamen **aTa** ein und legen Sie die folgenden Informationen fest:
 - Geben Sie **230** als B-Wert (Breite) ein.
 - Geben Sie **44** als H-Wert (Höhe) ein.
 - Geben Sie **165** als X-Wert (horizontale Position) ein.
 - Geben Sie **57** als Y-Wert (vertikale Position) ein.
 - Geben Sie auf der Registerkarte „Parameter“ als Textparameter **Hello World!** ein.
6. Ziehen Sie eine ColorPicker-Komponente auf die Bühne, platzieren Sie sie links neben der TextArea-Komponente und geben Sie ihr den Instanznamen **txtCp**. Geben Sie im Eigenschafteninspektor die folgenden Informationen ein:
 - Geben Sie **96** als X-Wert ein.
 - Geben Sie **72** als Y-Wert ein.
7. Ziehen Sie nacheinander drei RadioButton-Komponenten auf die Bühne und geben Sie den Instanzen die Namen **smallRb**, **largerRb** und **largestRb**. Geben Sie im Eigenschafteninspektor die folgenden Informationen ein:
 - Geben Sie für jede Instanz als Breite (B) **100** und als Höhe (H) **22** ein.
 - Geben Sie **155** als X-Wert ein.
 - Geben Sie als Y-Wert **120** für **smallRb**, **148** für **largerRb** und **175** für **largestRb** ein.
 - Geben Sie für jede Instanz als groupName-Parameter **fontRbGrp** ein.
 - Geben Sie auf der Registerkarte „Parameter“ die Bezeichnungen **Small**, **Larger** und **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:
- Geben Sie **130** als Breite (B) ein.
 - Geben Sie **265** als X-Wert ein.
 - Geben Sie **120** als Y-Wert ein.
 - 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 vorstehenden Schritt, um die label-Werte **Have a nice day!** und **Top of the Morning!** hinzuzufügen.
 - 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);
```

Die ersten drei Zeilen importieren die Ereignisklassen, die die Anwendung verwendet. Wenn ein Benutzer mit einer der Komponenten interagiert, tritt ein Ereignis ein. Die nächsten fünf Zeilen registrieren Ereignisprozeduren für das Ereignis, auf das die Anwendung wartet. Ein `click`-Ereignis tritt für eine `RadioButton`-Komponente ein, wenn ein Benutzer darauf klickt. Ein `change`-Ereignis tritt ein, wenn ein Benutzer mithilfe der `ColorPicker`-Komponente eine andere Farbe auswählt. Ein `change`-Ereignis tritt für die `ComboBox`-Komponente ein, wenn ein Benutzer eine andere Begrüßung aus der Dropdownliste auswählt.

In der vierten Zeile wird die `RadioButtonGroup`-Klasse importiert, sodass die Anwendung der Gruppe der Optionsfelder (`RadioButtons`) einen Ereignis-Listener zuweisen kann, anstatt jedem einzelnen Optionsfeld einen Listener zuzuweisen.

11. Führen Sie dem Bedienfeld „Aktionen“ die folgenden Codezeilen 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 Ereignisverarbeitungsfunktion `rbHandler` zu erstellen. Diese Funktion verarbeitet ein `click`-Ereignis, wenn ein 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);  
}
```

Diese Funktion verwendet eine `switch`-Anweisung, um die Eigenschaft `target` des `event`-Objekts zu untersuchen, um festzustellen, welche `RadioButton`-Komponente das Ereignis ausgelöst hat. Die Eigenschaft `currentTarget` enthält den Namen des Objekts, welches das Ereignis ausgelöst hat. Je nachdem, auf welche `RadioButton`-Komponente der Benutzer geklickt hat, ändert die Anwendung die Größe des Texts in der `TextArea`-Komponente auf 14, 18 oder 24 Punkt.

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

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

Diese Funktion setzt die Eigenschaft `color` 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 in der `ComboBox`-Komponente `event.target.selectedItem.label` ausgewählten Text.

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

So erstellen Sie die Anwendung „Greetings2“ mithilfe 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“.
Es wird ein neues Flash-Fenster angezeigt.
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 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. In den folgenden Schritten definieren Sie die `Greetings2`-Klasse. Diese Klasse definiert die Hauptfunktionen der 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“.

Es wird ein neues Skript-Fenster angezeigt.

9. Geben Sie den folgenden Code in das Skriptfenster ein:

```
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 Folgendes aus:

- Es importiert Klassen, die in der Datei verwendet werden. Normalerweise würden Sie diese Importanweisungen hinzufügen, wenn Sie auf die einzelnen Klassen im Code verweisen, um das Beispiel kurz zu halten; hier werden jedoch alle Anweisungen in einem Schritt importiert.
 - Es vereinbart Variablen, die die verschiedenen Typen von Komponentenobjekten repräsentieren, die dem Code hinzugefügt werden. Eine weitere Variable erstellt das TextFormat-Objekt tf.
 - Es definiert eine Konstruktorfunktion, Greetings2(), für die Klasse. In den folgenden Schritten werden dieser Funktion Zeilen hinzugefügt, ebenso wie der Klasse weitere Methoden hinzugefügt werden.
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(smallerRb);
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 führen Folgendes aus:

- Die Komponenten instanziiieren, die in der Anwendung verwendet werden.
- Die Größe, die Position und die Eigenschaften der Komponente festlegen.
- Jede Komponente mithilfe der Methode `addChild()` der Bühne hinzufügen.

13. Fügen Sie hinter der schließenden Klammer der Methode `bldRadioButtons()` die folgenden Codezeilen für die Methode `setUpHandlers()` 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 "smallerRb":
            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 „Greeting2s“ zu testen.

Ausführen der nachfolgenden Beispiele

Nachdem Sie die Greetings-Anwendung entwickelt und ausgeführt haben, verfügen Sie über die grundlegenden Kenntnisse, die Sie benötigen, um die anderen Codebeispiele in diesem Handbuch auszuführen. Der relevante ActionScript 3.0-Code in jedem Beispiel wird hervorgehoben und erläutert, und Sie können jedes Beispiel im vorliegenden Handbuch in eine FLA-Datei kopieren, kompilieren und ausführen.

In diesem Kapitel erfahren Sie, wie Sie in einem Dokument mit Komponenten arbeiten. Dieses Kapitel enthält die folgenden Themen:

Komponentenarchitektur	37
Verwenden von Komponentendateien	41
Debuggen von Komponentenanwendungen	44
Festlegen von Parametern und Eigenschaften	45
Die Bibliothek	48
Ändern der Komponentengröße	50
Live-Vorschau	51
Verarbeiten von Ereignissen	52
Verwenden von Anzeigelisten	54
Verwenden des FocusManagers	57
Verwenden von List-basierten Komponenten	59
Verwenden von Datenprovidern	61
Verwenden von CellRenderern	71
Barrierefreiheit von Komponenten	80

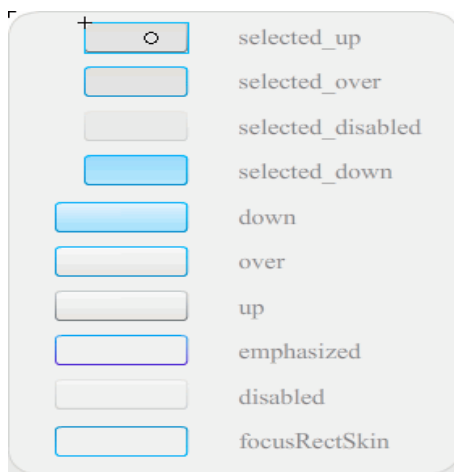
Komponentenarchitektur

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 mit Versionen vor Flash CS3 erstellt wurden. Informationen zur Verwendung von ActionScript 2.0-Komponenten finden Sie in den Handbüchern *ActionScript 2.0-Komponenten verwenden* und *ActionScript 2.0-Referenzhandbuch*.

Die ActionScript 3.0-UI-Komponenten werden als FLA-basierte Komponenten implementiert, Flash CS3 unterstützt jedoch auch SWC-Komponenten. So sind beispielsweise die FLVPlayback- und die FLVPlaybackCaptioning-Komponenten 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, deshalb werden sie hier separat beschrieben.

FLA-basierte Komponenten von ActionScript 3.0

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, springt Flash automatisch zu Bild 2 und öffnet eine Palette mit den Skins der Komponente. Die folgende Abbildung zeigt die Palette der Skins für die Button-Komponente.



Skins der Button-Komponente

Weitere Informationen zu Komponentenskins und zum Anpassen von Komponenten finden Sie in [Kapitel 4, „Anpassen von UI-Komponenten“](#) auf Seite 159 und unter „Anpassen der FLVPlayback-Komponente“ auf Seite 242.

Um die Kompilierung Ihrer Anwendung zu beschleunigen und Konflikte mit Ihren ActionScript 3.0-Einstellungen zu vermeiden, enthalten die FLA-basierten UI-Komponenten in Flash CS3 auch eine SWC-Datei, 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. Um eine Komponente mit ActionScript zu erstellen, müssen Sie mithilfe einer `import`-Anweisung auch die Klasse importieren, um darauf zugreifen zu können. Informationen zur `import`-Anweisung finden Sie im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

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, die das erneute Kompilieren von Symbolen und Codeabschnitten, die sich nicht ändern, überflüssig macht.

Die FLVPlayback- und die FLVPlaybackCaptioning-Komponenten sind SWC-basierte Komponenten. Sie verfügen nicht über interne, sondern über 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 oder eine benutzerdefinierte Skin erstellen. Weitere Informationen finden Sie unter [„Anpassen der FLVPlayback-Komponente“](#) auf Seite 242.

In Flash können Sie einen Movieclip in einen kompilierten Clip konvertieren, indem Sie wie nachstehend beschrieben vorgehen.

So kompilieren Sie einen Movieclip:

- Klicken Sie mit der rechten Maustaste (Windows) bzw. bei gedrückter Taste <Ctrl> (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 normale Movieclips. Kompilierte Clips können zwar nicht bearbeitet werden, aber ihre Eigenschaften können im Eigenschafteninspektor und im Komponenten-Inspektor angezeigt werden.

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

So exportieren Sie SWC-Dateien:

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

HINWEIS

Das Format einer Flash CS3-SWC-Datei ist kompatibel mit dem Flex SWC-Format, sodass SWC-Dateien zwischen den beiden Produkten ausgetauscht werden können, allerdings sind dabei u. U. Modifikationen erforderlich.

Informationen zum Erstellen von SWC-basierten Komponenten finden Sie unter www.adobe.com/go/learn_fl_creating_components_de.

Die ActionScript 3.0-Komponenten-API

Jede ActionScript 3.0-Komponente baut auf einer ActionScript 3.0-Klasse auf, die sich in einem Paketordner befindet und einen Namen im Format `fl.Paketname.Klassenname` hat. Die Button-Komponente ist zum Beispiel eine Instanz der Button-Klasse und hat den Paketnamen `fl.controls.Button`. Sie müssen auf den Paketnamen verweisen, wenn Sie eine Komponenteklasse in Ihre Anwendung importieren. Die Button-Klasse wird mit der folgenden Anweisung importiert:

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

Weitere Informationen zum Speicherort der Komponentenklassendateien finden Sie unter „[Verwenden von Komponentendateien](#)“ auf Seite 41.

Die Klasse einer Komponente definiert die Methoden, Eigenschaften, Ereignisse und Stile, die Ihnen die Interaktion mit der Komponente in der Anwendung ermöglichen. Die ActionScript 3.0-UI-Komponenten sind Unterklassen der Sprite- und UIComponent-Klassen und übernehmen Eigenschaften, Methoden und Ereignisse von diesen. Die Sprite-Klasse ist der Grundbaustein der Anzeigekiste und ähnelt einem Movieclip, verfügt jedoch nicht über eine Zeitleiste. Die UIComponent-Klasse ist die Basisklasse für alle visuellen Komponenten, sowohl interaktive als auch nicht interaktive. Der Vererbungspfad der einzelnen Komponenten sowie deren Eigenschaften, Methoden, Ereignisse und Stile werden im *Komponenten-Referenzhandbuch für ActionScript 3.0* beschrieben.

Alle ActionScript 3.0-Komponenten verwenden das ActionScript 3.0 Ereignisverarbeitungsmodell. Weitere Informationen zur Ereignisverarbeitung finden Sie unter „[Verarbeiten von Ereignissen](#)“ auf Seite 52 und im Handbuch *Programmieren mit ActionScript 3.0*.

Verwenden von 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.

Speicherort von Komponentendateien

Flash-Komponenten werden im Anwendungsordner „Configuration“ gespeichert.

HINWEIS

Informationen zu diesen Ordnern finden Sie unter „Mit Flash installierte Konfigurationsordner“ in *Flash verwenden*.

Komponenten werden in den folgenden Verzeichnissen installiert:

- Windows 2000 oder Windows XP:
C:\Programme\Adobe\Adobe Flash CS3*Sprache*\Configuration\Components
- Mac OS X:
Macintosh HD:Applications:Adobe Flash CS3:Configuration:Components

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

Sie können Komponenten auch in den folgenden benutzerbasierten Ordnern speichern:

- Windows 2000 oder Windows XP:
C:\Dokumente und Einstellungen*Benutzername*\Lokale Einstellungen
\Anwendungsdaten\Adobe\Adobe Flash CS3\en\Configuration\Components
- Mac OS X:
Macintosh HD:Users:<Benutzername>:Library:Application Support:Adobe:Flash
CS3:Configuration:Components

Speicherort von Komponenten Quelldateien

Die ActionScript (.as)-Klassendateien (oder *Quelldateien*) für Komponenten werden unter Windows 2000 oder Windows XP in den folgenden Anwendungsordnern gespeichert:

- UI-Komponenten
C:\Programme\Adobe\Adobe Flash CS3\en\Configuration\Component
Source\ActionScript 3.0\User Interface\fl
- FLVPlayback
C:\Programme\Adobe\Adobe Flash CS3\en\Configuration\Component
Source\ActionScript 3.0\FLVPlayback\fl\video
- FLVPlaybackCaptioning
C:\Programme\Adobe\Adobe Flash CS3\en\Configuration\Component
Source\ActionScript 3.0\FLVPlaybackCaptioning\fl\video

Auf Mac OS X-Systemen befinden sich die Quelldateien hier:

- UI-Komponenten
Macintosh HD:Applications:Adobe Flash CS3:Configuration:Component
Source>ActionScript 3.0:User Interface:fl
- FLVPlayback
Macintosh HD:Applications:Adobe Flash CS3:Configuration:Component
Source>ActionScript 3.0:FLVPlayback:fl:video
- FLVPlaybackCaptioning
Macintosh HD:Applications:Adobe Flash CS3:Configuration:Component
Source>ActionScript 3.0:FLVPlaybackCaptioning:fl:video

Komponentenquelldateien und Klassenpfad

Da bei den ActionScript 3.0-Klassen der Code kompiliert wird, sollten Sie den Speicherort der ActionScript-Klassendateien nicht in der Classpath-Variablen angeben. Wenn Sie ihn im Klassenpfad angeben, dauert die Kompilierung der Anwendungen länger. Falls Flash Komponentenklassendateien in der Classpath-Einstellung findet, hat die Klassendatei immer Vorrang vor dem einkompilierten Code der Komponente.

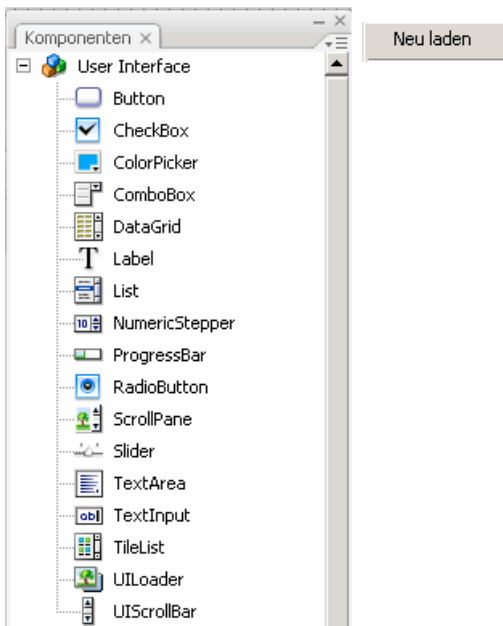
Es kann jedoch Gelegenheiten geben, bei denen Sie der Classpath-Einstellung den Speicherort der Komponentenquelldateien hinzufügen möchten, zum Beispiel beim Debugging einer Anwendung mit Komponenten. Weitere Informationen finden Sie unter [„Debuggen von Komponentenanwendungen“ auf Seite 44](#).

Ändern von Komponentendateien

Wenn Sie SWC-basierte Komponenten aktualisieren, hinzufügen oder entfernen oder wenn Sie Flash neue FLA-basierte Komponenten 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 dem Ordner „Components“ hinzugefügt haben.

So laden Sie Komponenten neu in das Bedienfeld „Komponenten“, während Flash ausgeführt wird:

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



Menüpunkt „Neu laden“ im Menü des Bedienfelds „Komponenten“

So entfernen Sie eine 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.

TIPP

Adobe empfiehlt, zunächst eine Kopie der Flash-Komponentendatei (.fla oder .as) zu erstellen, bevor Sie sie ändern. So können Sie die Datei bei Bedarf wiederherstellen.

Debuggen von Komponentenanwendungen

Die ActionScript 3.0-Komponenten enthalten ihren gesamten Quellcode, um die Kompilierung der Anwendung zu beschleunigen. Der Flash Debugger kann jedoch Code innerhalb von kompilierten Clips nicht überprüfen. Wenn Sie eine Anwendung bis zum Quellcode der Komponenten debuggen möchten, müssen Sie die Komponentenquelldateien deshalb der Classpath-Einstellung hinzufügen.

Der Speicherort des Komponentenpaketordners ist relativ zum Speicherort der Quelldateien für den Komponententyp. Um alle ActionScript 3.0-Quelldateien für alle UI-Komponenten zu referenzieren, fügen Sie dem Klassenpfad (Classpath) für die UI-Pakete den folgenden Speicherort hinzu:

```
$(AppConfig)/Component Source/ActionScript 3.0/User Interface
```

HINWEIS

Damit wird der einkompilierte Code für alle UI-Komponenten übergangen, und die Kompilierung der Anwendung dauert länger. Wenn Sie die Quelldatei einer Komponente geändert haben, zeigt diese Komponente möglicherweise ein anderes Verhalten.

Um den Klassenpfad festzulegen, wählen Sie „Bearbeiten“ > „Voreinstellungen“ und wählen dann in der Kategorieliste „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 mit den aktuellen Einstellungen.

Die Variable \$(AppConfig) bezieht sich auf den Flash CS3-Ordner „Configuration“ im Installationsverzeichnis von Flash CS3. Typischerweise sieht der Pfad folgendermaßen aus:

Windows 2000 oder Windows XP

```
C:\Program Files\Adobe\Adobe Flash CS3\language\Configuration\
```

Mac OS X

```
Macintosh HD:Applications:Adobe Flash CS3:Configuration:
```

HINWEIS

Wenn Sie eine Komponentenquelldatei ändern müssen, empfiehlt Adobe dringend, die Originalquelldatei in ein anderes Verzeichnis zu kopieren und dem Klassenpfad diesen Speicherort hinzuzufügen.

Weitere Informationen zum Speicherort der Komponentenquelldateien finden Sie unter [„Speicherort von Komponentenquelldateien“ auf Seite 42](#).

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 Komponentenkasse und wird im Eigenschafteninspektor und im Komponenten-Inspektor angezeigt. Die am häufigsten verwendeten Eigenschaften werden als Authoring-Parameter angezeigt, während andere mit ActionScript festgelegt werden 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.

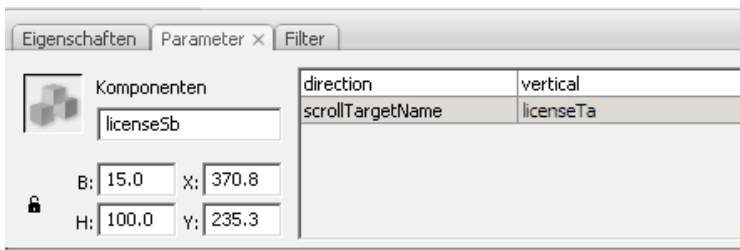
Die meisten ActionScript 3.0-UI-Komponenten übernehmen Eigenschaften und Methoden von der UIComponent-Klasse sowie von einer Basisklasse. Die Button- und CheckBox-Klassen übernehmen zum Beispiel Eigenschaften von der UIComponent-Klasse und von der BaseButton-Klasse. Sie haben Zugriff auf die übernommenen Eigenschaften sowie auf die eigenen Klasseneigenschaften einer Komponente. Die ProgressBar-Komponente übernimmt zum Beispiel die Eigenschaft `ProgressBar.enabled` von der UIComponent-Klasse, verfügt aber auch über die eigene Eigenschaft `ProgressBar.percentComplete`. Sie können auf beide Eigenschaften zugreifen, um eine Instanz der ProgressBar-Komponente zu bearbeiten. Weitere Informationen zu den Eigenschaften einer Komponente finden Sie unter dem jeweiligen Klasseneintrag im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Sie können Parameter für eine Komponenteninstanz im Eigenschafteninspektor oder im Komponenten-Inspektor festlegen.

So geben Sie einen Instanznamen für eine Komponente im Eigenschafteninspektor an:

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> unter der Dropdownliste „Movieclip“ ein. Alternativ dazu können Sie auch auf die Registerkarte „Parameter“ klicken und den Namen in das Feld unter *Komponenten* eingeben. Geben Sie Werte für alle Parameter, die Sie festlegen möchten, ein.

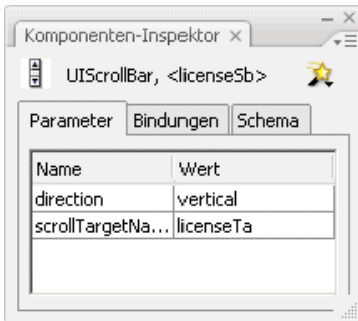
Es empfiehlt sich, dem Instanznamen ein Suffix hinzuzufügen, das die Art der Komponente kennzeichnet. Dies erleichtert das Lesen des ActionScript-Codes. In diesem Beispiel lautet der Instanzname **licenseSb**, da es sich bei der Komponente um eine Bildlaufleiste (scroll bar) handelt, mit der der Benutzer durch eine Lizenzvereinbarung im Textbereich **licenseTa** blättern kann.



Feld für den Instanznamen der Komponente

So geben Sie die Parameter einer Komponenteninstanz im Komponenten-Inspektor ein:

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 die aufgeführten Parameter ein.



Komponentenparameter 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. Mit dem folgenden ActionScript-Code wird zum Beispiel die Eigenschaft `width` (Breite) der `CheckBox`-Instanz `aCh` auf 50 Pixel festgelegt:

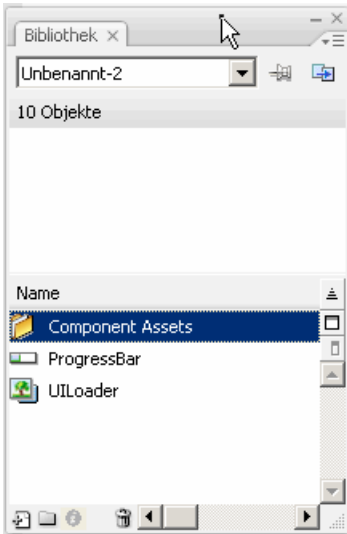
```
aCh.width = 50;
```

Mit der folgenden `if`-Anweisung wird überprüft, ob der Benutzer das Kontrollkästchen ausgewählt hat:

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

Die Bibliothek

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

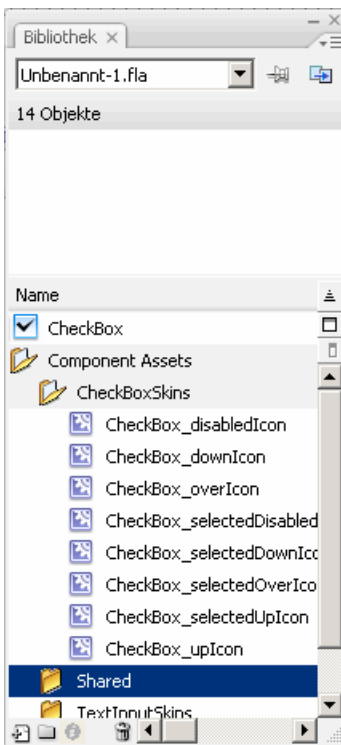


ProgressBar-Komponente im Bedienfeld „Bibliothek“

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 den Paketnamen und 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 bestehen aus einer Sammlung von Symbolen, welche die grafische Anzeige in der Anwendung ausmachen. Eine einzelne Skin ist eine grafische Darstellung, oder ein Movieclip, und gibt einen bestimmten Zustand der Komponente an. Im Ordner „Component Assets“ der CheckBox-Komponente repräsentiert die Skin „CheckBox_disabledIcon“ zum Beispiel die grafische Darstellung der Komponente im deaktivierten Zustand. Die Skin „CheckBox_selectedDownIcon“ repräsentiert das grafische Bild des Kontrollkästchens, wenn Sie darauf klicken und die Maustaste gedrückt halten.



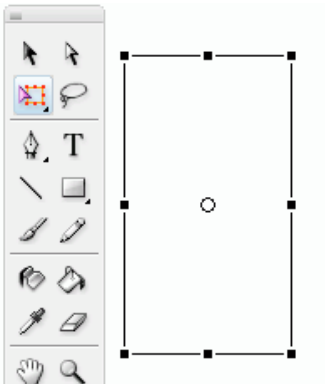
„Component Assets“ im Bedienfeld „Bibliothek“

Der Inhalt des Ordners „Component Assets“ ermöglicht Ihnen, bei Bedarf die Skin der Komponente zu ändern. Weitere Informationen finden Sie in [Kapitel 4, „Anpassen von UI-Komponenten“](#) auf Seite 159.

Nachdem der Bibliothek eine Komponente 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.

Ändern der Komponentengröße

Die Größe der Komponenteninstanzen kann mit dem Werkzeug „Frei transformieren“ oder mit der Methode `setSize()` angepasst werden.

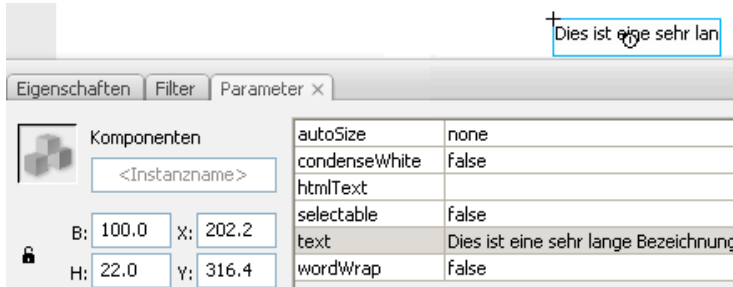


Ändern der Größe der List-Komponente auf der Bühne mit dem Werkzeug „Frei transformieren“

Die Methode `setSize()` kann von jeder Komponenteninstanz aufgerufen werden, um die Größe zu ändern (siehe `UIComponent.setSize()`). Mit dem folgenden Code wird die Größe einer Instanz der 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 Komponentenbezeichnungen angepasst. Wenn die Komponenteninstanz, die einem Dokument hinzugefügt wurde, für die Darstellung ihrer Bezeichnung nicht ausreicht, wird der Text in der Anzeige abgeschnitten. Sie müssen die Größe der Komponente so ändern, dass sie der Bezeichnung entspricht.



Abgeschnittener Text in einer Label-Komponente

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

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.

So schalten Sie die Live-Vorschau ein und aus:

- 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. Welche Komponentenparameter in der Live-Vorschau wiedergegeben werden, können Sie in den einzelnen Einträgen zu den Komponenten im [Komponenten-Referenzhandbuch für ActionScript 3.0](#) nachlesen.



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 überträgt Ereignisse, wenn ein Benutzer mit ihr interagiert. Wenn ein Benutzer zum Beispiel auf eine Schaltfläche klickt, überträgt die Button-Komponente das Ereignis `MouseEvent.CLICK`, und wenn ein Benutzer einen Eintrag in einer Liste auswählt, überträgt die List-Komponente das Ereignis `Event.CHANGE`. Ein Ereignis kann auch eintreten, wenn etwas Bestimmtes mit einer Komponente geschieht; wenn beispielsweise für eine UILoader-Instanz das Laden des Inhalts abgeschlossen ist, wird das Ereignis `Event.COMPLETE` generiert. Zum Verarbeiten eines Ereignisses schreiben Sie ActionScript-Code, der ausgeführt wird, wenn das Ereignis auftritt.

Zu den Ereignissen einer Komponente gehören die Ereignisse aller Klassen, von denen die Komponente übernimmt. 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 eine Komponente überträgt, finden Sie in den Abschnitten zu Ereignissen in den Klasseneinträgen der jeweiligen Komponente im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Eine umfassende Erläuterung der Ereignisverarbeitung in ActionScript 3.0 finden Sie in *Programmieren mit ActionScript 3.0*.

Ereignis-Listener

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

- Alle Ereignisse werden von einer Instanz einer Komponentenkasse übertragen. Die Komponenteninstanz ist der *Broadcaster*.
- Sie registrieren einen Ereignis-*Listener*, indem Sie die Methode `addEventListener()` für die Komponenteninstanz aufrufen. Die folgende Codezeile fügt zum Beispiel der Button-Instanz `aButton` einen Listener für das `MouseEvent.CLICK`-Ereignis 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 *Callback*-Funktion 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 unter „Ereignisobjekte“ auf Seite 53.
- Der Listener bleibt aktiv, bis die Anwendung beendet wird oder Sie ihn ausdrücklich mithilfe der `removeEventListener()`-Methode entfernen. Die folgende Codezeile entfernt zum Beispiel den Listener für das `MouseEvent.CLICK`-Ereignis von `aButton`:

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

Ereignisobjekte

Das Ereignisobjekt übernimmt 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 mit dem Typ des Ereignisses.
<code>target</code>	Ein Verweis auf die Komponenteninstanz, von der das Ereignis übertragen wird.

Wenn ein Ereignis über zusätzliche Eigenschaften verfügt, werden sie in der Beschreibung der Ereignisklasse im *Komponenten-Referenzhandbuch für ActionScript 3.0* beschrieben.

Das Ereignisobjekt wird automatisch generiert und an die Ereignisprozedur übergeben, wenn ein 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");
}
```

Verwenden von Anzeigelisten

Alle ActionScript 3.0-Komponenten übernehmen von der DisplayObject-Klasse und haben deshalb Zugriff auf deren Methoden und Eigenschaften, um mit der Anzeigeliste zu interagieren. Die *Anzeigeliste* ist die Hierarchie von angezeigten Objekten und visuellen Elementen in einer Anwendung. Diese Hierarchie enthält die folgenden Elemente:

- Die Bühne, die der Container auf oberster Ebene ist.
- Anzeigeobjekte, wozu u. a. Formen, Movieclips und Textfelder gehören.
- Anzeigeobjekt-Container, bei denen es sich um spezielle Arten von Anzeigeobjekten handelt, die untergeordnete Anzeigeobjekte haben.

Die Reihenfolge der Objekte in der Liste bestimmt ihre Tiefe im übergeordneten Container. Die Tiefe eines Objekts bezieht sich auf seine Position von oben nach unten bzw. von vorne nach hinten auf der Bühne oder seinem Anzeige-Container. Die Reihenfolge der Tiefe wird deutlich, wenn sich Objekte überlappen; sie ist aber auch vorhanden, wenn sie sich nicht überlappen. 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 der Objekte 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.

Hinzufügen von Komponenten zur Anzeigeliste

Sie können einem `DisplayObjectContainer`-Objekt ein Objekt hinzufügen, indem Sie die Methode `addChild()` oder `addChildAt()` des Containers aufrufen. Im Fall der Bühne können Sie der Anzeigeliste ein Objekt auch beim Authoring hinzufügen, indem Sie es erstellen, oder, wenn es sich um Komponenten handelt, indem Sie es aus dem Bedienfeld „Komponenten“ auf die Bühne ziehen. Wenn Sie einem Container ein Objekt mithilfe von ActionScript hinzufügen möchten, müssen Sie zunächst eine Instanz davon erstellen, indem Sie seinen Konstruktor mit dem Operator `new` und dann die Methode `addChild()` oder `addChildAt()` aufrufen, um sie auf der Bühne und in der Anzeigeliste zu platzieren. Die Methode `addChild()` platziert das Objekt an der nächsten Position in der Anzeigeliste, während `addChildAt()` die Position angibt, an der das Objekt hinzugefügt werden soll. Wenn Sie eine Position angeben, die bereits besetzt ist, werden die an dieser Position und darüber befindlichen Objekte um 1 nach oben verschoben. Die Eigenschaft `numChildren` eines `DisplayObjectContainer`-Objekts enthält die Anzahl der Anzeigeobjekte, die es enthalten kann. Sie können ein Objekt aus der Anzeigeliste abrufen, indem Sie die Methode `getChildAt()` aufrufen und die Position angeben. Wenn Ihnen der Name des Objekts bekannt ist, können Sie auch die Methode `getChildByName()` verwenden.

HINWEIS

Wenn Sie eine Komponente mithilfe von ActionScript hinzufügen, müssen Sie ihrer Nameeigenschaft einen Namen zuweisen, damit Sie in der Anzeigeliste über den Namen auf die Komponente zugreifen können.

Im folgenden Beispiel werden die Namen und die Positionen von drei Komponenten in der Anzeigeliste angezeigt. Ziehen Sie zunächst je eine Instanz der Komponenten `NumericStepper`, `Button` und `ComboBox` so auf die Bühne, dass sie sich überlappen, und geben Sie ihnen die Instanznamen **aNs**, **aButton** und **aCb**. 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 in der Anzeigelite

Sie können die Position eines Objekts in der Anzeigelite und seine Anzeigtiefe ändern, indem Sie die Methode `addChildAt()` aufrufen und den Namen des Objekts sowie die Position, an die Sie es verschieben möchten, als Parameter der Methode 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 Anzeigelite anzuzeigen:

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

Im Bedienfeld „Ausgabe“ wird Folgendes angezeigt:

```
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 `numChildren` die Nummer der Objekte (von 1 bis n) in der Anzeigelite ist, während die erste Position in der Liste 0 ist. Wenn die Liste zum Beispiel drei Objekte enthält, ist die Indexposition des dritten Objekts 2. Sie können also mit `numChildren - 1` auf die letzte Position in der Anzeigelite bzw. auf das oberste Objekte in Bezug auf die Anzeigtiefe verweisen.

Entfernen von Komponenten aus der Anzeigelite

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

So entfernen Sie eine Komponente aus der Anzeigeliste:

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 hinzu:

```
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 Beschreibung der Anzeigeliste finden Sie in [Kapitel 12, „Programmieren von Anzeigebjekten“](#) in *Programmieren mit ActionScript 3.0*.

Verwenden des FocusManagers

Wenn ein Benutzer in einer Flash-Anwendung mit der Tabulatortaste navigiert oder mit der Maustaste klickt, bestimmt die FocusManager-Klasse, welche Komponente den Eingabefokus erhält. Falls Sie keine Komponente erstellen, ist es nicht erforderlich, einer Anwendung eine FocusManager-Instanz hinzuzufügen oder Code zu schreiben, um den FocusManager zu aktivieren.

Wenn ein RadioButton-Objekt den Fokus erhält, wird dieses Objekt sowie alle Objekte mit demselben Wert für `groupName` vom FocusManager überprüft, und der Fokus wird dem Objekt gegeben, dessen Eigenschaft `selected` auf `true` eingestellt 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 zu 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 zu den einzelnen Klassen finden Sie unter „[Verwenden von Anzeigelisten](#)“ auf Seite 54.

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 Betätigen 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 einstellen. Dies wird im folgenden Code gezeigt:

```

import fl.managers.FocusManager;

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

```

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

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

Verwenden von List-basierten Komponenten

Die List-, DataGrid- und TileList-Komponenten übernehmen von der `SelectableList`-Basisklasse. Aus diesem Grund werden diese Komponenten List-basierte Komponenten genannt. Eine ComboBox-Komponente besteht aus einem Textfeld und einer Liste und ist deshalb ebenfalls eine List-basierte Komponente.

Eine List-Komponente besteht aus Zeilen. DataGrid- und TileList-Komponenten bestehen aus Zeilen, die in mehrere Spalten unterteilt sein können. Der Schnittpunkt einer Zeile und einer Spalte ist eine Zelle. In einer List-Komponente, die nur aus einer einzelnen Spalte mit Zeilen besteht, ist jede Zeile eine Zelle. Eine Zelle hat die beiden folgenden wichtigen Merkmale:

- Die Datenwerte in den Zellen werden Elemente genannt. 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 List-Komponente 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. Mit einem Datenprovider können Sie eine List-basierte Komponente mit Daten versehen, indem Sie ihn einfach der Eigenschaft `dataProvider` der Komponente zuweisen.
- Eine Zelle kann verschiedene Datentypen enthalten, darunter Text, Bilder, Movieclips oder beliebige Klassen, die Sie erstellen können. Aus diesem Grund muss eine Zelle auf eine Weise gezeichnet oder „gerendert“ werden, die ihrem Inhalt entspricht. Deshalb verfügen List-basierte Komponenten über einen *CellRenderer*, der die Zellen wiedergibt. Im Fall der DataGrid-Komponente ist jede Spalte ein `DataGridColumn`-Objekt, das ebenfalls über eine `cellRenderer`-Eigenschaft verfügt, sodass jede Spalte ihrem Inhalt entsprechend wiedergegeben werden kann.

Alle List-basierten Komponenten verfügen über die Eigenschaften `cellRenderer` und `dataProvider`, die Sie einstellen können, um die Zellen dieser Komponenten zu laden und wiederzugeben. Informationen zur Verwendung dieser Eigenschaften und zur Arbeit mit List-basierten Komponenten finden Sie unter „[Verwenden von Datenprovidern](#)“ auf Seite 61 und „[Verwenden von CellRenderern](#)“ auf Seite 71.

Verwenden von Daten Providern

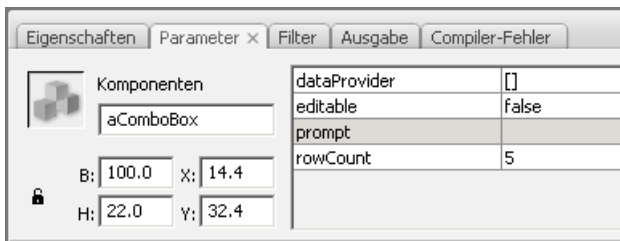
Ein Datenprovider ist eine Datenquelle, mit der Sie Daten an die ComboBox-, DataGrid-, List- und TileList-Komponenten übergeben können. Jeder dieser Komponentenklassen verfügt über die Eigenschaft `dataProvider`, der Sie ein `DataProvider`-Objekt zuweisen können, um die Zellen der Komponente mit Daten zu versehen. Normalerweise ist ein Datenprovider eine Datensammlung wie ein Array oder ein XML-Objekt.

Erstellen von Daten Providern

Für die ComboBox-, List- und TileList-Komponenten können Sie in der Authoring-Umgebung mithilfe des Parameters `dataProvider` einen Datenprovider erstellen. Die DataGrid-Komponente verfügt im Eigenschafteninspektor nicht über den `dataProvider`-Parameter, da sie mehrere Spalten haben kann und der Datenprovider deshalb komplexer sein muss. Sie können den Datenprovider für diese Komponenten auch mithilfe von ActionScript erstellen (so wie für die DataGrid-Komponente).

Verwenden des Parameters „dataProvider“

Sie können einen einfachen Datenprovider für die ComboBox-, List- und TileList-Komponenten erstellen, indem Sie auf der Registerkarte „Parameter“ des Eigenschafteninspektors oder des Komponenten-Inspektors auf den Parameter „dataProvider“ klicken. In der folgenden Abbildung ist dieser Parameter im Eigenschafteninspektor zu sehen.



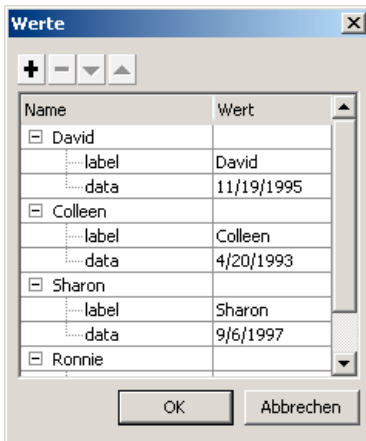
Parameter „dataProvider“ im Eigenschafteninspektor

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.



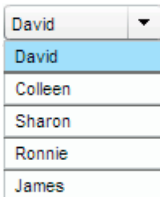
Dialogfeld „Werte“ für dataProvider

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



Dialogfeld „Werte“ mit Daten

Das Array, das Sie erstellen, besteht aus Paaren aus Namen- und Wert-Feldern. Die Namen-Felder sind `label` und `data`, und die Wert-Felder sind die Namen und Geburtstage der Kinder. Das Feld `label` gibt den Inhalt an, der in der Liste angezeigt wird, in diesem Fall die Namen der Kinder. Die fertig gestellte `ComboBox`-Instanz sieht wie folgt aus:



Die `ComboBox`-Komponente mit Daten vom `Datenprovider`

Wenn Sie alle Daten hinzugefügt haben, klicken Sie auf „OK“, um das Dialogfeld zu schließen. Das Array im `dataProvider`-Parameter ist jetzt mit den Elementen ausgefüllt, die Sie erstellt haben.

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

`dataProvider`-Parameter mit Daten

Sie können auf die Namen- und Daten-Werte, die Sie erstellt haben, zugreifen, indem Sie mit ActionScript-Code auf die Eigenschaft `dataProvider` der Komponente zugreifen.

Verwenden von ActionScript

Sie können einen `Datenprovider` erstellen, indem Sie die Daten in einem Array- oder XML-Objekt erstellen und das Objekt als `value`-Parameter an den `DataProvider`-Konstruktor übergeben.

HINWEIS

In ActionScript 3.0 können Sie ein Array- oder XML-Objekt nicht direkt einer `dataProvider`-Eigenschaft zuweisen, da die Eigenschaft als ein `DataProvider`-Objekt definiert ist und nur ein Objekt vom `DataProvider`-Typ erhalten kann.

Im folgenden Beispiel wird eine List-Komponente mit Daten versehen. Es handelt sich hierbei um eine einzelne Spalte mit Reihen, die die Namen verschiedener Kinder und ihre Geburtstage enthalten. 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 besteht aus Paaren aus Namen- und Wert-Feldern. Die Namen-Felder sind `label` und `data`, und die Wert-Felder sind die Namen und Geburtstage der Kinder. Das Feld `label` gibt den Inhalt an, der in der Liste angezeigt wird, in diesem Fall die Namen der Kinder. Die fertig gestellte List-Instanz sieht wie folgt aus:

David
Colleen
Sharon
Ronnie
James

Eine List-Komponente mit Daten vom Datenprovider

Der Wert des Datenfelds ist verfügbar, wenn der Benutzer ein Element in der Liste auswählt, indem er darauf klickt, und damit ein `change`-Ereignis auslöst. Im folgenden Beispiel wird dem vorigen Beispiel eine `TextArea`-Komponente (`aTa`) und eine Ereignisprozedur (`changeHandler`) hinzugefügt, um den Geburtstag des Kindes anzuzeigen, wenn ein Benutzer einen Namen in der Liste auswählt.

```
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);

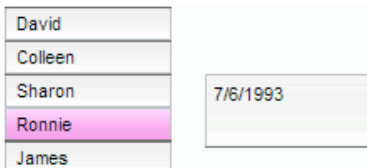
aList.move(150,150);
aTa.move(150, 260);

aList.addEventListener(Event.CHANGE, changeHandler);

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

```

Wenn ein Benutzer den Namen eines Kindes in der Liste auswählt, wird der Geburtstag des Kindes in der TextArea-Instanz angezeigt wie in der folgenden 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.



Anzeige des Datenfelds aus dem Datenprovider einer List-Komponente

In einem Datenprovider können Sie auch andere Daten als Text verwenden. Im folgenden Beispiel sind in einem Datenprovider, der Daten an eine TileListe-Komponente liefert, Movieclips enthalten. Der Datenprovider wird erstellt, indem `addItem()` aufgerufen wird, um jedes Element hinzuzufügen, nachdem der Movieclip, ein farbiges Feld, erstellt wurde.

```

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();
}

```

Sie können auch XML-Daten (anstelle eines Arrays) verwenden, um ein DataProvider-Objekt mit Daten zu versehen. Mit dem folgenden Code werden beispielsweise Daten in einem XML-Objekt mit dem Namen `employeesXML` gespeichert. Das Objekt wird dann als `value`-Parameter der `DataProvider()`-Konstruktorfunktion 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 Daten als Attribute der XML-Daten verfügbar machen, wie im vorigen Codebeispiel, oder als Eigenschaften der XML-Daten, wie im folgenden Beispiel:

```

var employeesXML:XML =
    <employees>
        <employee>
            <Name>Edna</Name>
            <ID>22</ID>
        </employee>
    </employees>

```

```

        <Name>Stu</Name>
        <ID>23</ID>
    </employee>
</employees>;

```

Der Datenprovider verfügt auch über einen Satz Methoden und Eigenschaften, die Ihnen den Zugriff und die Bearbeitung ermöglichen. Sie können mithilfe der DataProvider-API Elemente zu einem Datenprovider hinzufügen, daraus entfernen bzw. sie ersetzen, sortieren und verbinden.

Bearbeiten von Daten Providern

Mit den Methoden `addItem()` und `addItemAt()` können Sie einem Datenprovider Elemente hinzufügen. Im folgenden Beispiel werden Elemente hinzugefügt, die ein Benutzer in das Textfeld einer bearbeitbaren ComboBox-Komponente eingibt. Es setzt voraus, dass eine ComboBox-Komponente auf die Bühne gezogen wurde und `aCb` genannt wurde.

```

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 mithilfe deren Datenproviders auch ersetzen und entfernen. Im folgenden Beispiel werden zwei separate List-Komponenten implementiert, `listA` und `listB`, und es wird eine Schaltfläche mit dem Namen „Sync“ hinzugefügt. Wenn ein Benutzer auf die Schaltfläche klickt, werden die Elemente in `listB` durch die Elemente in `listA` ersetzt. Dies geschieht durch das Aufrufen der Methode `replaceItemAt()`. Wenn `listA` länger ist als `listB`, ruft der Beispielcode die Methode `addItem()` auf, um `listB` weitere Elemente hinzuzufügen. Wenn `listB` länger ist als `listA`, ruft der Beispielcode die Methode `removeItemAt()` auf, um die überflüssigen 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 Verbinden mit einem Datenprovider und das Sortieren der Elemente möglich. Im folgenden Beispiel werden zwei `DataGrid`-Instanzen (`aDg` und `bDg`) mit unvollständigen Aufstellungen für zwei Softball-Teams versehen. Außerdem wird die Schaltfläche mit der Beschriftung „Merge“ (Verbinden) hinzugefügt; wenn der Benutzer darauf klickt, verbindet die Ereignisprozedur (`mrgHandler`) die Aufstellung aus `bDg` mit der Aufstellung 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 über die DataProvider-Klasse im *[Komponenten-Referenzhandbuch für ActionScript 3.0](#)*.

Verwenden von CellRenderern

CellRenderer ist eine Klasse, die List-basierte Komponenten wie List, DataGrid, TileList und ComboBox verwenden, um benutzerdefinierte Zelleninhalte für ihre Zeilen zu bearbeiten und anzuzeigen. Eine benutzerdefinierte Zelle kann Text, eine vorgefertigte Komponente, wie beispielsweise eine CheckBox-Komponente, oder eine von Ihnen erstellte Anzeigeobjektklasse enthalten. Um Daten mit einem benutzerdefinierten CellRenderer wiederzugeben, können Sie entweder die CellRenderer-Klasse erweitern oder die ICellRenderer-Schnittstelle implementieren, um eigene CellRenderer-Klassen zu erstellen.

Die List-, DataGrid-, TileList- und ComboBox-Klassen sind Unterklassen der SelectableList-Klasse. Die SelectableList-Klasse beinhaltet einen `cellRenderer`-Stil. Dieser Stil definiert das Anzeigeobjekt, das die Komponente verwendet, um Zellen wiederzugeben.

Sie können die Formatierung der vom CellRenderer verwendeten Stile ändern, indem Sie die Methode `setRendererStyle()` des List-Objekts aufrufen (siehe nächster Abschnitt, [„Formatieren von Zellen“](#)). Alternativ dazu können Sie auch eine benutzerdefinierte Klasse definieren, die als CellRenderer verwendet wird (siehe [„Benutzerdefinierte CellRenderer-Klassen“](#) auf Seite 72).

Formatieren von Zellen

Die CellRenderer-Klasse enthält mehrere Stile, mit denen Sie das Format der Zelle steuern können.

Mit den folgenden Stilen können Sie die Skins definieren, die für die verschiedenen Zustände der Zelle verwendet werden (disabled, down, over und up):

- `disabledSkin` **und** `selectedDisabledSkin`
- `downSkin` **und** `selectedDownSkin`
- `overSkin` **und** `selectedOverSkin`
- `upSkin` **und** `selectedUpSkin`

Die folgenden Stile gelten für die Textformatierung:

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

Sie können diese Stile festlegen, indem Sie die Methode `setRendererStyle()` des List-Objekts oder die Methode `setStyle()` des `CellRenderer`-Objekts aufrufen. Sie können diese Stile 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`-Stile als benannte Eigenschaften des Objekts definiert.

Sie können die `clearRendererStyle()`-Methode aufrufen, um einen Stil auf seinen Standardwert zurückzusetzen.

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

Benutzerdefinierte CellRenderer-Klassen

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

So definieren Sie mit einer Klasse, die die `CellRenderer`-Klasse erweitert, einen benutzerdefinierten `CellRenderer`:

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“ anzuzeigen.
4. Ziehen Sie eine List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
5. Wenn der Eigenschafteninspektor nicht angezeigt wird, wählen Sie „Fenster“ > „Eigenschaften“ > „Eigenschaften“ aus.
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 im Skriptfenster den folgenden Code ein, um die CustomCellRenderer-Klasse zu definieren:


```
package {
    import fl.controls.listClasses.CellRenderer;
    import flash.text.TextFormat;
    import flash.filters.BevelFilter;
    public class CustomCellRenderer2 extends CellRenderer {
        public function CustomCellRenderer2() {
            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“.

Sie können einen `CellRenderer` auch mithilfe einer beliebigen Klasse, die von der `DisplayObject`-Klasse übernimmt und die `ICellRenderer`-Schnittstelle implementiert, definieren. Im folgenden Codebeispiel werden zwei Klassen definiert. Die `ListSample2`-Klasse fügt der Anzeigeliste ein `List`-Objekt hinzu und definiert dessen `CellRenderer` für die Verwendung der `CustomRenderer`-Klasse. Die `CustomRenderer`-Klasse erweitert die `CheckBox`-Klasse (die ihrerseits die `DisplayObject`-Klasse erweitert) und implementiert die `ICellRenderer`-Schnittstelle. Beachten Sie, dass die `CustomRenderer`-Klasse `get`- oder `set`-Methoden für die Eigenschaften `data` und `listData` definiert. Diese Eigenschaften werden in der `ICellRenderer`-Schnittstelle definiert. Andere Eigenschaften und Methoden, die in der `ICellRenderer`-Schnittstelle definiert werden (die Eigenschaft `selected` und die Methode `setSize()`) werden bereits in der `CheckBox`-Klasse definiert:

So definieren Sie mit einer Klasse, die die `ICellRenderer`-Schnittstelle implementiert, einen benutzerdefinierten `CellRenderer`:

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“ anzuzeigen.
4. Ziehen Sie eine `List`-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
5. Wenn der Eigenschafteninspektor nicht angezeigt wird, wählen Sie „Fenster“ > „Eigenschaften“ > „Eigenschaften“ aus.
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 im Skriptfenster den folgenden Code 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“.

Sie können auch ein Symbol in der Bibliothek verwenden, um einen CellRenderer zu definieren. Das Symbol muss für ActionScript exportiert werden, und der Klassenname für das Bibliothekssymbol muss über eine zugeordnete Klassendatei verfügen, die entweder die ICellRenderer-Schnittstelle implementiert oder die CellRenderer-Klasse (bzw. eine ihrer Unterklassen) erweitert.

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

So definieren Sie einen CellRenderer mit einem Bibliothekssymbol:

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“ anzuzeigen.
4. Ziehen Sie eine List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
5. Wenn der Eigenschafteninspektor nicht angezeigt wird, wählen Sie „Fenster“ > „Eigenschaften“ > „Eigenschaften“ aus.
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 dataProvider-Reihe.
8. Klicken Sie im Dialogfeld „Werte“ zwei Mal auf die Schaltfläche das Pluszeichen, um zwei Datenelemente (deren Beschriftungen auf label0 und label1 eingestellt sind) hinzuzufügen, 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
- In erstes Bild exportieren: Ausgewählt
- Klasse: MyCellRenderer
- Basisklasse: flash.display.SimpleButton

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 im 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 {
    }
    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“.

CellRenderer-Eigenschaften

Die Eigenschaft `data` ist ein Objekt, das alle Eigenschaften enthält, die für den `CellRenderer` festgelegt wurden. Im folgenden Beispiel einer Klasse, die einen benutzerdefinierten `CellRenderer` definiert, der die `CheckBox`-Klasse erweitert, übergibt die `set`-Funktion für die Eigenschaft `data` den Wert von `data.label` an die Eigenschaft `label`, die von der `CheckBox`-Klasse übernommen wird:

```

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 in der Liste eine Zelle ausgewählt ist oder nicht.

CellRenderer für eine Spalte eines DataGrid-Objekts

Ein `DataGrid`-Objekt kann mehrere Spalten haben, und Sie können für jede Spalte einen anderen `CellRenderer` angeben. Jede Spalte eines `DataGrid`-Objekts wird durch ein `DataGridColumn`-Objekt repräsentiert, und die `DataGridColumn`-Klasse enthält die Eigenschaft `cellRenderer`, für die Sie den `CellRenderer` für die Spalte definieren können.

CellRenderer für eine bearbeitbare Zelle

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 wird und der Benutzer auf die zu bearbeitende Zelle klickt. Um einen `CellRenderer` für die bearbeitbare Zelle zu definieren, legen Sie die Eigenschaft `itemEditor` für jedes Element des `columns`-Arrays des `DataGrid`-Objekts fest.

Verwenden von Bildern, SWF-Dateien oder Movieclips als CellRenderer

Die ImageCell-Klasse, eine Unterklasse der CellRenderer-Klasse, definiert ein Objekt, das zum Rendern von Zellen verwendet wird, deren Hauptinhalt ein Bild, eine SWF-Datei oder ein Movieclip ist. Die ImageCell-Klasse enthält die folgenden Stile, um das Erscheinungsbild der Zelle festzulegen:

- `imagePadding` – Die Zellauffüllung, die den Rand der Zelle vom Rand des Bilds trennt (in Pixel)
- `selectedSkin` – Die Skin, mit der der ausgewählte Zustand angezeigt wird
- `textOverlayAlpha` – Die Transparenz der Überlagerung hinter der Zellenbeschriftung
- `textPadding` – Die Zellauffüllung, die den Rand der Zelle vom Rand des Textes trennt (in Pixel)

Die ImageCell-Klasse ist der Standard-CellRenderer für die TileList-Klasse.

Barrierefreiheit von Komponenten

Sie können die grafischen Inhalte Ihrer Flash-Anwendungen für Benutzer mit eingeschränktem Sehvermögen zugänglich machen, indem Sie die Verwendung eines Bildschirmlesegeräts ermöglichen, welches eine gesprochene Beschreibung des Bildschirminhalts bereitstellt. Wie Sie Flash-Anwendungen für ein Bildschirmlesegerät verfügbar machen, lesen Sie in [Kapitel 18, „Barrierefreie Inhalte erstellen“](#) in *Flash verwenden*.

Damit ActionScript 3.0-Komponenten für ein Bildschirmlesegerät erkennbar sind, müssen Sie auch die jeweilige Accessibility-Klasse importieren und die Methode `enableAccessibility()` dieser Klasse aufrufen. Die folgenden ActionScript 3.0-Komponenten sind für Bildschirmlesegeräte geeignet:

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

Die Accessibility-Klassen der Komponenten befinden sich im Paket `fl.accessibility`. Damit zum Beispiel eine CheckBox-Komponente für ein Bildschirmlesegerät erkennbar ist, müssen Sie Ihrer Anwendung die folgenden Anweisungen hinzufügen:

```
import fl.accessibility.CheckBoxAccImpl;
```

```
CheckBoxAccImpl.enableAccessibility();
```

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

HINWEIS

Durch die Implementierung der Barrierefreiheit nimmt die Dateigröße in geringem Maße zu, da die erforderlichen Klassen beim Kompilieren einbezogen werden.

Bei den meisten Komponenten ist auch die Navigation per Tastatur möglich. Weitere Informationen zum Verwenden zugänglicher Komponenten und zum Navigieren mithilfe der Tastatur finden Sie in den Abschnitten zur Benutzerinteraktion in „[Verwenden von UI-Komponenten](#)“ und zu den [Accessibility](#)-Klassen im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Verwenden von UI-Komponenten

In diesem Kapitel wird die Verwendung der folgenden ActionScript 3.0-UI-Komponenten, die in Flash enthalten sind, beschrieben:

Button-Komponente	83
CheckBox-Komponente	88
ColorPicker-Komponente	92
ComboBox-Komponente	96
DataGrid-Komponente	100
Label-Komponente	108
List-Komponente	111
NumericStepper-Komponente	117
ProgressBar-Komponente	121
RadioButton-Komponente	129
ScrollPane-Komponente	133
Slider-Komponente	136
TextArea-Komponente	140
TextInput-Komponente	144
TileList-Komponente	149
U Loader-Komponente	153
UIScrollBar-Komponente	155

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. Ein Schaltfläche mit Umschaltfunktion (Toggle-Schaltfläche) bleibt nach dem Klicken so lange aktiviert, bis der Benutzer erneut darauf klickt.

Die Schaltfläche (Button) gehört zu den grundlegenden Bestandteilen von vielen Formularen und Web-Anwendungen. 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 der Button-Komponente

Sie können eine Schaltfläche in einer Anwendung 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:

Taste	Beschreibung
Umschalt+Tab	Der Fokus wird auf das vorherige Objekt verschoben.
Leertaste	Die Schaltfläche wird gedrückt bzw. losgelassen, und das Ereignis <code>click</code> wird ausgelöst.
Tabulator	Der Fokus wird auf das nächste Objekt verschoben.
Eingabetaste	Der Fokus wird auf das nächste Objekt verschoben, falls eine Schaltfläche als Standardschaltfläche des FocusManager festgelegt wurde.

Weitere Informationen zur Steuerung des Fokus finden Sie in den Abschnitten über die Schnittstelle [IFocusManager](#) und über [FocusManager](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0* sowie unter „[Verwenden des FocusManagers](#)“ auf Seite 57.

Die Änderungen an den Parametern der einzelnen Button-Instanzen, die beim Authoring im Eigenschafteninspektor oder im Komponenteninspektor 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 als Standardschaltfläche anzugeben (also die Schaltfläche, für die beim Drücken der Eingabetaste das click-Ereignis ausgelöst wird), legen Sie `FocusManager.defaultButton` fest. Im folgenden Codebeispiel wird als Standardschaltfläche eine `Button`-Instanz mit dem Namen `submitButton` festgelegt.

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

Wenn Sie die `Button`-Komponente einer Anwendung hinzufügen, können Sie sie für ein Bildschirmlesegerät zugänglich machen, indem Sie die folgenden `ActionScript`-Codezeilen hinzufügen:

```
import fl.accessibility.ButtonAccImpl;
```

```
ButtonAccImpl.enableAccessibility();
```

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

Button-Parameter

Im **Eigenschaftenspektor** („Fenster“ > „Eigenschaften“ > „Eigenschaften“) oder im **Komponenten-Spektor** („Fenster“ > „Komponenten-Spektor“) der einzelnen `Button`-Instanzen können Sie die folgenden **Authoring-Parameter** festlegen: `emphasized`, `label`, `labelPlacement`, `selected` und `toggle`. Jeder dieser Parameter verfügt über eine entsprechende `ActionScript`-Eigenschaft desselben Namens. Wenn Sie diesen Parametern einen Wert zu weisen, legen Sie den anfänglichen Zustand der Eigenschaft in der Anwendung fest. Durch das Einstellen der Eigenschaft in `ActionScript` wird der Wert, den Sie für den Parameter festlegen, außer Kraft gesetzt. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur `Button`-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Erstellen von Anwendungen mit der Button-Komponente

Im Folgenden wird erläutert, wie Sie einer Anwendung beim **Authoring** eine `Button`-Komponente hinzufügen. In diesem Beispiel ändert die `Button`-Komponente den Status der Komponente `ColorPicker`, wenn auf die Schaltfläche geklickt wird.

So erstellen Sie eine Anwendung mit der Button-Komponente:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine Button-Komponente aus dem 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. Ziehen Sie eine ColorPicker-Komponente auf die 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;
    }
}
```

Mit der zweiten Codezeile wird die Funktion `clickHandler()` als Ereignisverarbeitungsfunktion für das `MouseEvent.CLICK`-Ereignis registriert. Das Ereignis wird ausgelöst, wenn der Benutzer auf die Schaltfläche klickt, sodass die `clickHandler()`-Funktion je nach Wert der Button-Komponente eine der folgenden Aktionen ausführt:

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

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

Mit dem folgenden Verfahren wird eine Schaltfläche mit Umschaltfunktion mithilfe von ActionScript erstellt und der Ereignistyp im Bedienfeld „Ausgabe“ angezeigt, wenn auf die Schaltfläche geklickt wird. Im Beispiel wird die Button-Instanz erstellt, indem der Konstruktor der Klasse aufgerufen wird, und sie wird mit der `addChild()`-Methode auf der Bühne platziert.

So erstellen Sie mithilfe von ActionScript-Code eine Schaltfläche:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die Button-Komponente aus dem 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);
```

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

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 die Meldung „Ereignistyp: click“ im Bedienfeld „Ausgabe“ an.

CheckBox-Komponente

Eine [CheckBox](#)-Komponente ist ein Kontrollkästchen, also ein quadratisches Feld, das ausgewählt werden kann oder dessen Auswahl aufgehoben werden kann. Ausgewählte Kontrollkästchen werden durch Häkchen gekennzeichnet. Kontrollkästchen können mit einem Beschriftungstext versehen werden, der wahlweise links, rechts, oberhalb oder unterhalb vom Kontrollkästchen positioniert werden kann.

Sie können Kontrollkästchen verwenden, wenn mehrere Boolesche Werte (`true` oder `false`) abgefragt werden sollen, die sich nicht gegenseitig ausschließen. In einer Anwendung, in der Benutzer danach gefragt werden, welches Auto sie kaufen würden, könnten Sie die Benutzer Merkmale des Autos anhand von Kontrollkästchen auswählen lassen.

Benutzerinteraktion mit der CheckBox-Komponente

Sie können eine CheckBox-Komponente in einer Anwendung aktivieren oder deaktivieren. Wenn ein Kontrollkästchen aktiviert ist und der Benutzer darauf oder auf dessen Beschriftung klickt, erhält das Kontrollkästchen 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 CheckBox-Komponente losgelassen wird. Außerdem verfügt die CheckBox-Komponente über zwei „disabled“-Zustände, „selected“ und „deselected“, die die Skins `selectedDisabledSkin` bzw. `disabledSkin` verwenden, in denen keine Interaktion mit der Maus oder der Tastatur möglich ist.

Deaktivierte Kontrollkästchen werden unabhängig von den Bedienvorgängen des Benutzers immer im Disabled-Erscheinungsbild 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:

Taste	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.
Tabulator	Verschiebt den Fokus auf das nächste Element.

Weitere Informationen zur Steuerung des Fokus finden Sie unter „[Verwenden des FocusManagers](#)“ auf Seite 57 und im Abschnitt über die [FocusManager](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3*.

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

Wenn Sie die CheckBox-Komponente einer Anwendung hinzufügen, können Sie sie für ein Bildschirmlesegerät zugänglich machen, indem Sie die folgenden ActionScript-Codezeilen hinzufügen:

```
import fl.accessibility.CheckBoxAccImpl;
```

```
CheckBoxAccImpl.enableAccessibility();
```

Sie aktivieren die Eingabehilfen für eine Komponente nur einmal, unabhängig davon, wie viele Instanzen der Komponente Sie verwenden.

CheckBox-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer [CheckBox](#)-Komponente im Eigenschafteninspektor oder im Komponenteninspektor festgelegt werden: `label`, `labelPlacement` und `selected`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur CheckBox-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Erstellen von Anwendungen mit der CheckBox-Komponente

Im folgenden Verfahren wird eine CheckBox-Komponente beim Authoring einer Anwendung hinzugefügt. Hierbei wird ein Auszug aus einem Kreditantrag verwendet. Im Formular wird gefragt, ob der Antragsteller ein Eigenheim besitzt. Um mit „Ja“ zu antworten, kann der Benutzer ein Kontrollkästchen auswählen. In diesem Fall zeigt das Formular zwei Optionsfelder an, mit denen der Benutzer des ungefähren Wert des Hauses angeben kann.

So erstellen Sie eine Anwendung mit der CheckBox-Komponente:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine CheckBox-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
3. Führen Sie im Eigenschafteninspektor die folgenden Schritte durch:
 - Geben Sie als Instanznamen **homeCh** ein.
 - Geben Sie **140** als Wert für die Breite (B) ein.
 - Geben Sie „**Own your home?**“ für den Parameter „label“ ein.
4. Ziehen Sie zwei **RadioButton**-Komponenten aus dem 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 als Wert für den Parameter „W“ (Breite) der beiden RadioButton-Komponenten **120** ein.
 - Geben Sie für den Parameter „label“ der Instanz **underRb** den Wert **Under \$500,000?** ein.
 - Geben Sie für den Parameter „label“ der Instanz **overRb** den Wert **Over \$500,000?** ein.
 - Geben Sie für beide RadioButton-Komponenten 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;
}
```

Mit diesem Code wird eine Ereignisprozedur für ein `CLICK`-Ereignis erstellt, das die `RadioButton`-Komponenten `underRb` und `overRb` aktiviert, falls das Kontrollkästchen `homeCh` ausgewählt wurde, und sie deaktiviert, falls `homeCh` nicht ausgewählt wurde.

Weitere Informationen finden Sie im Abschnitt über die `MouseEvent`-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

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

Im folgenden Beispiel wird die oben stehende Anwendung dupliziert, allerdings werden die `CheckBox`-Komponente und die `RadioButton`-Komponenten mit ActionScript erstellt.

So erstellen Sie mithilfe von ActionScript-Code ein Kontrollkästchen:

1. Erstellen Sie eine neue Flash-Datei (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, drücken Sie <Strg>+<L> oder wählen Sie „Fenster“ > „Bibliothek“, um es zu öffnen.

Damit stehen die Komponenten für die Anwendung zur Verfügung, befinden sich jedoch noch nicht auf der Bühne.

3. Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein, um Instanzen 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?";

```

In diesem Codebeispiel werden die `CheckBox()`- und `RadioButton()`-Konstruktoren verwendet, um die Komponenten zu erstellen, und die `addChild()`-Methode, um sie auf der Bühne zu platzieren. Zur Positionierung der Komponenten auf der Bühne wird die `move()`-Methode verwendet.

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;
}

```

Mit diesem Code wird eine Ereignisprozedur für das `CLICK`-Ereignis erstellt, das die `RadioButton`-Komponenten `underRb` und `overRb` aktiviert, falls das Kontrollkästchen `homeCh` ausgewählt wurde, und sie deaktiviert, falls `homeCh` nicht ausgewählt wurde.

Weitere Informationen finden Sie im Abschnitt über die `MouseEvent`-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

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

ColorPicker-Komponente

Die `ColorPicker`-Komponente ermöglicht einem Benutzer die Auswahl einer Farbe aus einer Farbfeldliste. Im Standardmodus der `ColorPicker`-Komponente wird eine einzelne Farbe in einer quadratischen Schaltfläche angezeigt. Wenn ein Benutzer auf diese Schaltfläche klickt, wird in einem Farbfeld eine Liste der verfügbaren Farben zusammen mit einem Textfeld, in dem der Hexadezimalwert der zurzeit ausgewählten Farbe angezeigt wird, eingeblendet.

Sie können die Farben, die in der `ColorPicker`-Komponente angezeigt werden, festlegen, indem Sie die `colors`-Eigenschaft der Komponente auf die gewünschten Farbwerte einstellen.

Benutzerinteraktion mit der ColorPicker-Komponente

Eine ColorPicker-Komponente (eine Farbauswahl) ermöglicht es dem Benutzer, eine Farbe auszuwählen und auf ein anderes Objekt in der Anwendung anzuwenden. Wenn Sie zum Beispiel möchten, dass der Benutzer bestimmte Elemente der Anwendung, wie Hintergrundfarbe oder Textfarbe, anpassen kann, können Sie eine ColorPicker-Komponente einbinden und die vom Benutzer ausgewählte Farbe anwenden.

Ein Benutzer wählt eine Farbe, indem er auf das entsprechende Farbfeld klickt oder den Hexadezimalwert der Farbe in das Textfeld eingibt. Nachdem der Benutzer eine Farbe ausgewählt hat, können Sie diese mithilfe der `selectedColor`-Eigenschaft der ColorPicker-Komponente auf den Text oder ein anderes Objekt der Anwendung anwenden.

Eine ColorPicker-Instanz erhält den Eingabefokus, wenn der Benutzer den Mauszeiger darauf bewegt oder die Instanz mit der Tabulatortaste ansteuert. Wenn das Farbfeld einer ColorPicker-Komponente geöffnet ist, können Sie die folgenden Tasten zur Steuerung verwenden:

Taste	Beschreibung
Pos 1	Bewegt die Auswahl zur ersten Farbe im Farbfeld.
Pfeil nach oben	Bewegt die Auswahl im Farbfeld eine Zeile nach oben.
Pfeil nach unten	Bewegt die Auswahl im Farbfeld eine Zeile nach unten.
Pfeil nach rechts	Bewegt die Auswahl im Farbfeld eine Farbe nach rechts.
Pfeil nach links	Bewegt die Auswahl im Farbfeld eine Farbe nach links.
Ende	Bewegt die Auswahl zur letzten Farbe im Farbfeld.

ColorPicker-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer ColorPicker-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden:

`selectedColor` und `showTextField`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur ColorPicker-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Erstellen von Anwendungen mit der ColorPicker-Komponente

Im folgenden Beispiel wird einer Anwendung beim Authoring eine ColorPicker-Komponente hinzugefügt. In diesem Beispiel wird jedes Mal, wenn die Farbe in der Farbauswahl geändert wird die Funktion `changeHandler()` aufgerufen, um die Funktion `drawBox()` aufzurufen, die ein neues Feld mit der ausgewählten Farbe zeichnet.

So erstellen Sie eine Anwendung mit der ColorPicker-Komponente:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine ColorPicker-Komponente aus dem 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 die ColorPicker-Komponente und wählen Sie eine Farbe aus, um das Feld mit einer Farbe zu versehen.

Im folgenden Beispiel werden der `ColorPicker()`-Konstruktor und `addChild()` verwendet, um eine ColorPicker-Komponente auf der Bühne zu platzieren. Die `colors`-Eigenschaft wird auf die Farbwerte für Rot (0xFF0000), Grün (0x00FF00) und Blau (0x0000FF) eingestellt, um die Farben festzulegen, die in der Farbauswahl angezeigt werden. Außerdem wird eine TextArea-Komponente erstellt, und jedes Mal, wenn in der Farbauswahl eine andere Farbe ausgewählt wird, ändert sich die Farbe des Textes in diesem Textfeld entsprechend.

So erstellen Sie mithilfe von ActionScript-Code eine Farbauswahl:

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

ComboBox-Komponente

Eine **ComboBox**-Komponente (ein Kombinationsfeld) ermöglicht dem Benutzer die Auswahl einer einzelnen Option aus einer Dropdownliste. Ein Kombinationsfeld kann statisch oder bearbeitbar sein. Bei einem bearbeitbaren Kombinationsfeld kann der Benutzer Text direkt in das Textfeld am oberen Rand der Liste eingeben. Wenn die Dropdownliste über den unteren Dokumentrand hinausreicht, wird sie nach oben statt nach unten geöffnet. Die ComboBox-Komponente besteht aus drei Unterkomponenten: **BaseButton**-, **TextInput**- und **List**-Komponente.

Bei bearbeitbaren Kombinationsfeldern umfasst der Hit-Bereich (Kollisionsbereich) nur die Schaltfläche; das Textfeld gehört nicht zum Hit-Bereich. Bei statischen Kombinationsfeldern umfasst der Kollisionsbereich die Schaltfläche und das Textfeld. Der Kollisionsbereich reagiert durch Öffnen oder Schließen der Dropdownliste.

Wenn der Benutzer mit der Maus oder über die Tastatur ein Element in der Liste auswählt, wird die Bezeichnung des ausgewählten Elements in das Textfeld am oberen Rand 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.

Wenn die ComboBox-Komponente bearbeitbar ist, die Eigenschaft `editable` also den Wert `true` hat, wird der Fokus mit den folgenden Tasten vom Texteingabefeld entfernt, und der vorherige Wert bleibt im Feld erhalten. Eine Ausnahme stellt die Eingabetaste dar, mit der zuerst der neue Wert übernommen wird, sofern der Benutzer Text eingeben hat.

Taste	Beschreibung
Umschalt+Tab	Verschiebt den Fokus auf das vorherige Element. Wenn ein neues Element ausgewählt wird, wird ein <code>change</code> -Ereignis ausgelöst.
Tabulator	Der Fokus wird auf das nächste Element verschoben. Wenn ein neues Element ausgewählt wird, wird ein <code>change</code> -Ereignis ausgelöst.
Pfeil nach unten	Verschiebt die Auswahl um ein Element nach unten.
Ende	Verschiebt die Auswahl an das Ende 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 eingeben hat, wird mit der Eingabetaste der Wert auf den eingegebenen Text festgelegt.
Pos 1	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.

Wenn Sie die ComboBox-Komponente einer Anwendung hinzufügen, können Sie sie für ein Bildschirmlesegerät zugänglich machen, indem Sie die folgenden ActionScript-Codezeilen hinzufügen:

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

Sie aktivieren die Eingabehilfen für eine Komponente nur einmal, unabhängig davon, wie viele Instanzen der Komponente Sie verwenden.

ComboBox-Parameter

Die folgenden Parameter können für jede Instanz einer ComboBox-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden: `dataProvider`, `editable`, `prompt` und `rowCount`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur ComboBox-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*. Informationen zur Verwendung des `dataProvider`-Parameters finden Sie unter „Verwenden des Parameters „`dataProvider`““ auf Seite 61.

Erstellen von Anwendungen mit der ComboBox-Komponente

Im Folgenden wird beschrieben, wie Sie einer Anwendung beim Authoring eine ComboBox-Komponente hinzufügen. Das Kombinationsfeld ist bearbeitbar, und wenn Sie **Add** in das Textfeld eingeben, wird im Beispiel ein Element zur Dropdownliste hinzugefügt.

So erstellen Sie eine Anwendung mit der ComboBox-Komponente:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine ComboBox-Komponente auf die Bühne und geben Sie ihr den Instanznamen `aCb`. Setzen Sie 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“.

Im folgenden Beispiel wird eine ComboBox-Komponente mithilfe von ActionScript erstellt und mit einer Liste der Universitäten im Gebiet San Francisco versehen. Die Eigenschaft `width` der ComboBox-Komponente wird so eingestellt, dass sie der Breite der Eingabeaufforderung entspricht. Die Eigenschaft `dropdownWidth` wird so eingestellt, dass sie etwas breiter als der längste Universitätsname ist.

Im Beispiel wird die Liste der Universitäten in einer Array-Instanz erstellt, wobei mit der Eigenschaft `label` die Namen der Universitäten und mit der Eigenschaft `data` die URLs der Uni-Websites gespeichert werden. Das Array wird der ComboBox-Komponente zugewiesen, indem deren Eigenschaft `dataProvider` festgelegt wird.

Wenn ein Benutzer eine Universität aus der Liste auswählt, wird ein `Event.CHANGE`-Ereignis ausgelöst und die Funktion `changeHandler()` aufgerufen, welche die Eigenschaft `data` in eine URL-Anforderung zum Zugriff auf die Uni-Website lädt.

Beachten Sie, dass mit der letzten Zeile die Eigenschaft `selectedIndex` der ComboBox-Komponente auf -1 gesetzt wird, um wieder die Eingabeaufforderung anzuzeigen, wenn die Liste geschlossen wird. Andernfalls wird die Eingabeaufforderung durch den Namen der ausgewählten Universität ersetzt.

So erstellen Sie mithilfe von ActionScript-Code ein Kombinationsfeld:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die ComboBox-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.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 der Flash-Authoring-Umgebung implementieren und ausführen, es wird jedoch eine Warnmeldung angezeigt, wenn Sie versuchen, die Website einer Universität durch Klicken auf die Elemente im Kombinationsfeld aufzurufen. Das ComboBox-Beispiel mit vollem Funktionsumfang finden Sie im Internet unter:

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

DataGrid-Komponente

Mit der [DataGrid](#)-Komponente (einem Datenraster) können Sie Daten in Zeilen und Spalten anzeigen, wobei die Daten aus einem Array oder einer externen XML-Datei bezogen werden, die Sie in ein Array für den Datenprovider leiten. Die DataGrid-Komponente enthält vertikale und horizontale Bildlaufleisten, Ereignisunterstützung (auch Unterstützung bearbeitbarer Zellen) und Sortiermöglichkeiten.

Eigenschaften wie Schriftart, Farbe und Ränder von Spalten in einem Raster können angepasst und in der Größe geändert werden. Ein benutzerdefinierter Movieclip kann als „CellRenderer“ für alle Spalten eines Rasters verwendet werden. (Ein CellRenderer zeigt den Inhalt einer Zelle an.) Sie können die Bildlaufleisten deaktivieren und die DataGrid-Methoden verwenden, um eine seitenweise Anzeige zu erstellen. Weitere Informationen zur Anpassung finden Sie im Abschnitt über die [DataGridColumn](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Benutzerinteraktion mit der DataGrid-Komponente

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

Wenn die `sortableColumns`-Eigenschaft und die `sortable`-Eigenschaft der Spalte beide auf `true` gesetzt sind, werden die Daten im Datenraster beim Klicken auf eine Spaltenüberschrift anhand der Werte in dieser Spalte sortiert. Sie können die Sortierung für eine einzelne Spalte deaktivieren, indem Sie deren `sortable`-Eigenschaft auf `false` setzen.

Wenn die `resizableColumns`-Eigenschaft 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 auf `true` gesetzt sind.

Weitere Informationen finden Sie in den Abschnitten über die [DataGrid](#)- und [DataGridColumn](#)-Klassen im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

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

Taste	Beschreibung
Pfeil nach unten	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.
Pfeil nach oben	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+Pfeil nach oben/unten	Wenn das Datenraster nicht bearbeitbar ist und <code>allowMultipleSelection</code> den Wert <code>true</code> hat, werden aneinandergrenzende Zeilen ausgewählt. Durch Drücken der Pfeiltaste in die entgegengesetzte Richtung wird die Auswahl der ausgewählten Zeilen aufgehoben, bis Sie wieder am Ausgangspunkt ankommen, wonach die Zeilen in dieser Richtung ausgewählt werden.
Umschalt+Klicken	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+Klicken	Wenn <code>allowMultipleSelection</code> den Wert <code>true</code> hat, werden weitere Zeilen ausgewählt, die nicht benachbart sein müssen.

Taste	Beschreibung
Pfeil nach rechts	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.
Pfeil nach links	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.
Pos 1	Wählt die erste Zeile des Datenrasters aus.
Ende	Wählt die letzte Zeile des Datenrasters aus.
Bild auf	Wählt die erste Zeile auf der aktuellen Seite des Datenrasters aus. Eine Seite besteht aus der Anzahl von Zeilen, die das Datenraster ohne Bildlauf anzeigen kann.
Bild ab	Wählt die letzte Zeile auf der aktuellen Seite des Datenrasters aus. Eine Seite besteht aus der Anzahl von Zeilen, die das Datenraster ohne Bildlauf anzeigen kann.
Eingabetaste/ Umschalt+Eingabetaste	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	Wenn das Datenraster bearbeitbar ist, wird der Fokus auf das vorherige/nächste Element verschoben, bis das Ende der Spalte erreicht wird, und dann zur vorherigen/nächsten Zeile, bis die erste bzw. letzte Zelle erreicht wird. Wenn die erste Zelle ausgewählt ist, wird der Fokus mit Umschalt+Tab zum vorherigen Steuerelement verschoben. Wenn die letzte Zelle ausgewählt ist, wird der Fokus mit der Tabulatortaste zum nächsten Steuerelement verschoben. Wenn das Datenraster nicht bearbeitbar ist, wird der Fokus zum vorherigen/nächsten Steuerelement verschoben.

Die DataGrid-Komponente lässt sich als Grundlage für zahlreiche Typen datengesteuerter Anwendungen einsetzen. Sie können auf einfache Weise eine formatierte Tabellenansicht von Daten anzeigen oder mit den CellRenderer-Möglichkeiten 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

Wenn Sie eine Anwendung mit der DataGrid-Komponente erstellen, ist es hilfreich, wenn Sie die Konstruktion der List-Komponente verstehen, da die DataGrid-Klasse die SelectList-Klasse erweitert. Weitere Informationen zur SelectList-Klasse und zur List-Komponente finden Sie in den Abschnitten zur [SelectableList](#)-Klasse und zur [List](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Wenn Sie die DataGrid-Komponente einer Anwendung hinzufügen, können Sie sie für ein Bildschirmlesegerät zugänglich machen, indem Sie die folgenden ActionScript-Codezeilen hinzufügen:

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

Diese Anweisung ist unabhängig von der Anzahl der Instanzen nur einmal pro Komponente erforderlich. Weitere Informationen finden Sie in [Kapitel 18](#), „*Barrierefreie Inhalte erstellen*,“ in *Flash verwenden*.

DataGrid-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer DataGrid-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden

`allowMultipleSelection`, `editable`, `headerHeight`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `resizableColumns`, `rowHeight`, `showHeaders`, `verticalLineScrollSize`, `verticalPageScrollSize` und `verticalScrollPolicy`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur DataGrid-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Erstellen von Anwendungen mit der DataGrid-Komponente

Um eine Anwendung mit der DataGrid-Komponenten zu erstellen, müssen Sie zuerst bestimmen, woher die Daten stammen. Normalerweise kommen die Daten aus einem Array, das Sie mithilfe der `dataProvider`-Eigenschaft in das Raster ziehen können. Sie können auch die Methoden der DataGrid- und DataGridColumn-Klassen verwenden, um dem Raster Daten hinzuzufügen.

Im folgenden Beispiel wird ein Datenraster erstellt, um die Aufstellungsliste eines Softball-Teams anzuzeigen. Das Raster wird in einem Array definiert (`aRoster`) und der `dataProvider`-Eigenschaft der DataGrid-Komponente zugewiesen.

So verwenden Sie einen lokalen Datenprovider mit der DataGrid-Komponente:

1. Wählen Sie in Flash „Datei“ > „Neu“ und dann „Flash-Datei (ActionScript 3.0)“.
2. Ziehen Sie die DataGrid-Komponente aus dem 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:

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

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

Sie können auf eine beliebige Spaltenüberschrift klicken, um den Inhalt des Datenrasters anhand der Werte in der entsprechenden Spalte in absteigender Reihenfolge zu sortieren.

Im folgenden Beispiel werden einer `DataGrid`-Komponente mithilfe der `addColumn()`-Methode `DataGridColumn`-Instanzen hinzugefügt. Die Spalten enthalten Namen von Spielern sowie deren Punkte. Im Beispiel wird auch die `sortOptions`-Eigenschaft festgelegt, um die Sortieroptionen für die einzelnen Spalten anzugeben:

`Array.CASEINSENSITIVE` für die Spalte mit den Namen und `Array.NUMERIC` für die Spalte mit den Punkten. Die Größe des Datenraster wird angegeben, indem die Länge auf die Anzahl der Zeilen und die Breite auf 200 festgelegt wird.

So geben Sie für eine `DataGrid`-Komponente in einer Anwendung Spalten an und sortieren sie:

1. Wählen Sie in Flash „Datei“ > „Neu“ und dann „Flash-Datei (ActionScript 3.0)“.
2. Ziehen Sie die `DataGrid`-Komponente aus dem 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:

```

import fl.controls.dataGridClasses.DataGridColumn;
import fl.events.DataGridEvent;
import fl.data.DataProvider;
// Create columns to enable sorting of data.
var namedDGC:DataGridColumn = new DataGridColumn("name");
namedDGC.sortOptions = Array.CASEINSENSITIVE;
var scoreDGC:DataGridColumn = new DataGridColumn("score");
scoreDGC.sortOptions = Array.NUMERIC;
aDg.addColumn(namedDGC);
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“.

Im folgenden Beispiel wird eine DataGrid-Komponente mithilfe von ActionScript erstellt und mit einem Array von Spielernamen und Punktzahlen versehen.

So erstellen Sie mithilfe von ActionScript-Code eine DataGrid-Komponenteninstanz:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die DataGrid-Komponente aus dem 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 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);
```

Mit diesem Code wird die DataGrid-Instanz erstellt, und die Größe und Position des Rasters werden festgelegt.

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“.

Im folgenden Beispiel wird die DataGridColumn-Klasse verwendet, um die Spalten des Datenrasters zu erstellen. Das Datenraster wird mit Daten versehen, indem ein XML-Objekt als value-Parameter des DataProvider()-Konstruktors übergeben wird.

So laden Sie eine XML-Datei in ein Datenraster:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Doppelklicken Sie im Bedienfeld „Komponenten“ auf die DataGrid-Komponente, um sie der Bühne hinzuzufügen.
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:

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

var teamXML:XML = <team>
    <player name="Player A" avg="0.293" />
    <player name="Player B" avg="0.214" />
    <player name="Player C" avg="0.317" />
</team>;

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;
```

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

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 Eigenschafteninspektor oder im Komponenteninspektor 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

Mit der Label-Komponente können Sie Textbezeichnungen für andere Komponenten eines Formulars erzeugen. So können Sie beispielsweise ein Texteingabefeld (TextInput), in das der Benutzer seinen Namen eingeben soll, mit der Bezeichnung „Name:“ versehen. Die Verwendung einer Label-Komponente hat gegenüber einem einfachen Textfeld den Vorteil, dass Sie Stile verwenden können, um ein einheitliches Erscheinungsbild beizubehalten.

Wenn Sie eine Label-Komponente drehen möchten, müssen Sie Schriftarten einbetten, andernfalls werden sie beim Testen des Films nicht angezeigt.

Label-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer [Label](#)-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden: `autoSize`, `condenseWhite`, `selectable`, `text` und `wordWrap`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur Label-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Erstellen von Anwendungen mit der Label-Komponente

Im Folgenden wird erläutert, wie Sie einer Anwendung beim Authoring eine Label-Komponente hinzufügen. Im folgenden Beispiel zeigt die Bezeichnung den Text „Expiration Date“ an.

So erstellen Sie eine Anwendung mit der Label-Komponente:

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

Im folgenden Beispiel wird eine Label-Komponente mithilfe von ActionScript erstellt. Die Bezeichnung wird verwendet, um den Zweck einer ColorPicker-Komponente anzugeben, und mit der `htmlText`-Eigenschaft wird der Text der Bezeichnung formatiert.

So erstellen Sie mithilfe von ActionScript-Code eine Label-Komponenteninstanz:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die Label-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ des aktuellen Dokuments.
3. Ziehen Sie die ColorPicker-Komponente aus dem 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:

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

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

addChild(aLabel);
addChild(aCp);

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

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

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

List-Komponente

Bei der [List](#)-Komponente handelt es sich um 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. Ein Beispiel: Das Sortiment eines Online-Shops umfasst 30 Artikel, die in einer Liste mit Bildlaufleiste ausgewählt werden können. Beim Bestellvorgang wählt der Kunde durch Klicken den gewünschten Artikel aus.

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:

Taste	Beschreibung
Alphanumerische Tasten	Steuert das nächste Element an, dessen Beschriftung mit <code>Key.getAscii()</code> beginnt.
Strg	Toggle-Taste, mit der die Auswahl mehrerer nicht aufeinander folgender Elemente vorgenommen oder aufgehoben werden kann.
Pfeil nach unten	Verschiebt die Auswahl um ein Element nach unten.
Pos 1	Verschiebt die Auswahl an den Anfang der Liste.
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.
Pfeil nach oben	Verschiebt die Auswahl um ein Element nach oben.

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.
Beachten Sie, dass der Bildlauf in Pixel und nicht in Zeilen angegeben wird.

Weitere Informationen zur Steuerung des Fokus finden Sie in den Abschnitten über die Schnittstelle [IFocusManager](#) und über [FocusManager](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0* sowie unter „Verwenden des FocusManagers“ auf Seite 57.

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

Wenn Sie die List-Komponente einer Anwendung hinzufügen, können Sie sie für ein Bildschirmlesegerät zugänglich machen, indem Sie die folgenden ActionScript-Codezeilen hinzufügen:

```
import fl.accessibility.ListAccImpl;

ListAccImpl.enableAccessibility();
```


Diese Anweisung ist unabhängig von der Anzahl der Instanzen nur einmal pro Komponente erforderlich. Weitere Informationen finden Sie in [Kapitel 18, „Barrierefreie Inhalte erstellen“](#), in *Flash verwenden*.

List-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer [List](#)-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden:

`allowMultipleSelection`, `dataProvider`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `multipleSelection`, `verticalLineScrollSize`, `verticalPageScrollSize` und `verticalScrollPolicy`.

Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur List-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Informationen zur Verwendung des `dataProvider`-Parameters finden Sie unter „[Verwenden des Parameters „dataProvider“](#)“ auf Seite 61.

Erstellen von Anwendungen mit der List-Komponente

Im folgenden Beispiel wird beschrieben, wie Sie einer Anwendung beim Authoring eine List-Komponente hinzufügen. In diesem Beispiel besteht die Liste aus Bezeichnungen, die Automodelle angeben, sowie aus Datenfelder, die die jeweiligen Preise enthalten.

So fügen Sie einer Anwendung eine einfache List-Komponente hinzu:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
3. Führen Sie im Eigenschafteninspektor die folgenden Schritte durch:
 - Geben Sie den Instanznamen `aList` ein.
 - Weisen Sie der Breite (B) den Wert `200` zu.
4. Erstellen Sie mit dem Textwerkzeug ein Textfeld unterhalb von `aList` und geben Sie ihm den Namen `aTf`.

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 (Xc11nt 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 `addItem()`-Methode verwendet, um `aList` mit drei Elementen zu versehen, denen jeweils ein `label`-Wert, der in der Liste angezeigt wird, und ein `data`-Wert zugewiesen wird. Wenn Sie ein Element in der Liste auswählen, ruft der Ereignis-Listener die `showData()`-Funktion auf, die den `data`-Wert des ausgewählten Elements anzeigt.

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

Im folgenden Beispiel wird ebenfalls eine Liste mit Automodellen und Preisen erstellt. Statt mit der `addItem()`-Methode wird die Liste mithilfe eines Datenproviders mit Daten versehen.

So füllen Sie eine List-Instanz mithilfe eines Datenproviders:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
3. Führen Sie im Eigenschafteninspektor die folgenden Schritte durch:
 - Geben Sie den Instanznamen `aList` ein.
 - Weisen Sie der Breite (B) den Wert `200` zu.
4. Erstellen Sie mit dem Textwerkzeug ein Textfeld unterhalb von `aList` und geben Sie ihm den Namen `aTf`.

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 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.

Im folgenden Beispiel wird eine Liste mit Farbnamen erstellt. Wenn Sie eine Farbe auswählen, wird sie auf einen Movieclip angewendet.

So verwenden Sie eine List-Komponente zur Steuerung einer MovieClip-Instanz:

1. Erstellen Sie eine Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
 - Geben Sie als Instanznamen **aList** ein.
 - Geben Sie **60** als H-Wert 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:

```
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“, um die Anwendung auszuführen.

5. Klicken Sie auf Farben in der Liste, um sie in einem Movieclip anzuzeigen.

Im folgenden Beispiel wird mithilfe von ActionScript eine einfache Liste erstellt, die mit der addItem()-Methode mit Daten versehen wird.

So erstellen Sie mithilfe von ActionScript-Code eine List-Komponenteninstanz:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die List-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.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“, um die Anwendung auszuführen.

NumericStepper-Komponente

Die [NumericStepper](#)-Komponente ermöglicht es dem Benutzer, eine geordnete Zahlenmenge schrittweise durchzugehen. Die Komponente besteht aus einer Zahl in einem Textfeld, das neben kleinen Pfeilschaltflächen nach oben und unten 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 Eigenschafteninspektor oder Komponenten-Inspektor wieder. In der Live-Vorschau können die `NumericStepper`-Pfeilschaltflächen jedoch nicht mit der Maus oder der Tastatur bedient werden.

Benutzerinteraktion mit der NumericStepper-Komponente

Sie können die `NumericStepper`-Komponente überall verwendet, wo der Benutzer einen Zahlenwert auswählen soll. Beispielsweise können Sie die `NumericStepper`-Komponente in einem Formular einsetzen, damit ein Benutzer den Monat, den Tag und das Jahr des Ablaufdatums seiner Kreditkarte eingeben 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 die Größe des Steppers beim Authoring anpassen, wenn Sie mehr als zwei Ziffern anzeigen möchten (z. B. die Zahlen 5246 oder 1,34).

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

Taste	Beschreibung
Pfeil nach unten	Der Wert ändert sich um eine Einheit.
Pfeil nach links	Verschiebt die Einfügemarke im Textfeld nach links.
Pfeil nach rechts	Verschiebt die Einfügemarke im Textfeld nach rechts.
Umschalt+Tab	Der Fokus wird auf das vorherige Objekt verschoben.
Tabulator	Der Fokus wird auf das nächste Objekt verschoben.
Pfeil nach oben	Der Wert ändert sich um eine Einheit.

Weitere Informationen zur Steuerung des Fokus finden Sie im Abschnitt über die [FocusManager](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3* und unter „Verwenden des FocusManagers“ auf Seite 57.

NumericStepper-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer [NumericStepper](#)-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden: `maximum`, `minimum`, `stepSize` und `value`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur NumericStepper-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Erstellen von Anwendungen mit der NumericStepper-Komponente

Im folgenden Verfahren wird erklärt, wie Sie einer Anwendung beim Authoring die NumericStepper-Komponente hinzufügen. 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 des NumericSteppers ändert, wird der neue Wert in der `text`-Eigenschaft der Label-Instanz angezeigt.

So erstellen Sie eine Anwendung mit der NumericStepper-Komponente:

1. Ziehen Sie die NumericStepper-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
2. Geben Sie im Eigenschafteninspektor den Instanznamen `aNs` ein.
3. Ziehen Sie eine Label-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
4. Geben Sie im Eigenschafteninspektor 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 die `text`-Eigenschaft der Bezeichnung auf den Wert des NumericStepper gesetzt. Die `changeHandler()`-Funktion aktualisiert die `text` der Label-Komponente, wenn der Wert in der NumericStepper-Instanz geändert wird.

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

Im folgenden Beispiel werden drei NumericStepper-Komponenten mithilfe von ActionScript-Code erstellt: je eine für die Eingabe von Monat, Tag und Jahr des Geburtsdatums. Außerdem werden Bezeichnungen für eine Eingabeaufforderung und für Titel der einzelnen NumericStepper-Komponenten hinzugefügt.

So erstellen Sie mithilfe von ActionScript-Code eine NumericStepper-Komponente:

1. Erstellen Sie eine neue Flash-Datei (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“, um die Anwendung auszuführen.

ProgressBar-Komponente

Die [ProgressBar](#)-Komponente (eine Fortschrittleiste) zeigt den Ablauf des Ladevorgangs von Inhalten an, um den Benutzer über den Fortschritt zu informieren, wenn die Inhalte umfangreich sind und die Ausführung der Anwendung verzögern kann. 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* Fortschrittleiste ist eine lineare Darstellung des Fortschritts einer Aufgabe im Zeitverlauf; sie wird verwendet, wenn bekannt ist, wie groß der geladene Inhalt ist. Eine *unbestimmte* Fortschrittleiste wird verwendet, wenn die Größe des geladenen Inhalts nicht bekannt ist. Sie können auch eine Label-Komponente hinzufügen, um den Fortschritt des Ladevorgangs als Prozentwert anzuzeigen.

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

Die ProgressBar-Komponente kann in drei Modi verwendet werden. Am häufigsten werden der Ereignismodus („event“) und der Abfragemodus („polled“) verwendet. Diese Modi geben einen Ladevorgang an, der entweder die Ereignisse `progress` und `complete` erzeugt (Ereignismodus und Abfragemodus) oder die Eigenschaften `bytesLoaded` und `bytesTotal` übergibt (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. Sie können die `indeterminate`-Eigenschaft festlegen, um anzugeben, ob die Fortschrittleiste eine gestreifte Füllung und eine Quelle unbekannter Größe hat (`true`) oder eine einfarbige Füllung und eine Quelle bekannter Größe (`false`).

Sie legen den Modus der Fortschrittleiste fest, in dem Sie deren `mode`-Eigenschaft angeben, entweder mithilfe des `mode`-Parameters im Eigenschafteninspektor oder im Komponenteninspektor, oder mithilfe von `ActionScript`.

Wenn Sie die ProgressBar-Komponente verwenden, um den Status einer Verarbeitung anzuzeigen (zum Beispiel bei der Analyse von 100.000 Elementen), gibt es keine sichtbaren Aktualisierungen der Fortschrittleiste, wenn sich die Komponente in einer Einzelbildschleife befindet, da der Bildschirm nicht neu aufgebaut wird.

ProgressBar-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer [ProgressBar](#)-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden: `direction`, `mode` und `source`. Jeder dieser Parameter verfügt über 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 über die ProgressBar-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Erstellen von Anwendungen mit der ProgressBar-Komponente

Im folgenden Verfahren wird beschrieben, wie Sie einer Anwendung beim Authoring eine ProgressBar-Komponente hinzufügen. In diesem Beispiel verwendet die Fortschrittleiste den Ereignismodus. Im Ereignismodus muss der geladene Inhalt die Ereignisse `progress` und `complete` erzeugen, mit denen die Fortschrittleiste den Fortschritt anzeigt. Wenn das `progress`-Ereignis eintritt, wird die Bezeichnung aktualisiert, um anzuzeigen, wie viel Prozent des Inhalts geladen wurde. Wenn das `complete`-Ereignis eintritt, wird im Beispiel „Loading complete“ sowie der Wert der `bytesTotal`-Eigenschaft, wobei es sich um die Größe der Datei handelt, angezeigt.

So erstellen Sie eine Anwendung mit der ProgressBar-Komponente im Ereignismodus:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die ProgressBar-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
 - Geben Sie im Eigenschafteninspektor den Instanznamen **aPb** ein.
 - Geben Sie auf der Registerkarte „Parameter“ für den X-Wert **200** ein.
 - Geben Sie **260** als Y-Wert ein.
 - Wählen Sie „event“ für den Parameter `mode`.
3. Ziehen Sie die Button-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
 - Geben Sie im Eigenschafteninspektor den Instanznamen **loadButton** ein.
 - Geben Sie **220** als X-Parameter ein.
 - Geben Sie **290** als Y-Parameter ein.
 - Geben Sie **Load Sound** als Wert für den Parameter `label` ein.
4. Ziehen Sie eine Label-Komponente auf die Bühne und geben Sie ihr den Instanznamen **progLabel**.
 - Geben Sie **150** als Breite (B) ein.
 - Geben Sie **200** als X-Parameter ein.
 - Geben Sie **230** als Y-Parameter ein.
 - Löschen Sie im Bereich „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“.

Im folgenden Beispiel befindet sich die ProgressBar-Komponente im Abfragemodus. Im Abfragemodus wird der Fortschritt bestimmt, indem progress-Ereignisse für den Inhalt, der geladen wird, abgefragt werden und mithilfe der Eigenschaften `bytesLoaded` und `bytesTotal` der Fortschritt berechnet wird. Im Beispiel wird ein Sound-Objekt geladen, das Eintreten von progress-Ereignissen abgefragt und der Prozentwert des geladenen Inhalts anhand der Eigenschaften `bytesLoaded` und `bytesTotal` berechnet. Der bereits geladene Inhalt wird als Prozentwert in einer Bezeichnung und im Bedienfeld „Ausgabe“ angezeigt.

So erstellen Sie eine Anwendung mit der ProgressBar-Komponente im Abfragemodus:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine ProgressBar-Komponente aus dem 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 im Bereich „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“, um die Anwendung auszuführen.

Im folgenden Beispiel befindet sich die `ProgressBar`-Komponente im manuellen Modus. Im manuellen Modus müssen Sie den Fortschritt manuell festlegen, indem Sie die Methode `setProgress()` aufrufen und die aktuellen Werte und Höchstwerte eingeben, um den Fortschritt festzustellen. Die Eigenschaft `source` stellen Sie im manuellen Modus nicht ein. Im Beispiel wird eine `NumericStepper`-Komponente mit dem Höchstwert 250 verwendet, um die Fortschrittleiste zu inkrementieren. Wenn der Wert der `NumericStepper`-Komponente geändert wird und ein `CHANGE`-Ereignis auslöst, ruft die Ereignisprozedur (`nsChangeHandler`) die `setProgress()`-Methode auf, um die Fortschrittleiste weiterzubewegen. Außerdem wird der Prozentwert des Fortschritts angezeigt, der anhand des Höchstwerts berechnet wird.

So erstellen Sie eine Anwendung mit der `ProgressBar`-Komponente im manuellen Modus:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine `ProgressBar`-Komponente aus dem 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.
 - Geben Sie im Bereich Parameter **250** für den maximum-Parameter, **0** für den minimum-Parameter, **1** für den `stepSize`-Parameter und **0** für den `value`-Parameter ein.
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 **150** als Breite (B) 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“, um die Anwendung auszuführen.
7. Klicken Sie auf den Pfeil nach oben der NumericStepper-Komponente, um die Fortschrittleiste weiterzubewegen.

Im folgenden Beispiel wird eine ProgressBar-Komponente mithilfe von ActionScript erstellt. Außerdem wird die Funktion des vorigen Beispiels wiederholt, in dem eine Fortschrittleiste im manuellen Modus erstellt wurde.

So erstellen Sie mithilfe von ActionScript-Code eine Fortschrittleiste:

1. Erstellen Sie eine neue Flash-Datei (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“, um die Anwendung auszuführen.
7. Klicken Sie auf den Pfeil nach oben der NumericStepper-Komponente, um die Fortschrittleiste weiterzubewegen.

RadioButton-Komponente

Mit der [RadioButton](#)-Komponente (einem 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 gehören zu den grundlegenden Bestandteilen von vielen Formularen im Internet. 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, erhält nur das ausgewählte Optionsfeld den Fokus. Der Benutzer kann die Auswahl dann mit den folgenden Tasten steuern:

Taste	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.
Tabulator	Der Fokus wird von der Optionsfeldgruppe auf die nächste Komponente verschoben.

Weitere Informationen zur Steuerung des Fokus finden Sie in den Abschnitten über die Schnittstelle [IFocusManager](#) und über [FocusManager](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0* sowie unter „Verwenden des FocusManagers“ auf Seite 57.

Eine Live-Vorschau der einzelnen RadioButton-Instanzen auf der Bühne spiegelt die Änderungen wieder, die im Eigenschafteninspektor oder Komponenteninspektor 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 unter [„RadioButton-Parameter“ auf Seite 130](#).

Wenn Sie die RadioButton-Komponente einer Anwendung hinzufügen, können Sie sie für ein Bildschirmlesegerät zugänglich machen, indem Sie die folgenden Codezeilen hinzufügen:

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

Sie aktivieren die Eingabehilfen für eine Komponente nur einmal, unabhängig davon, wie viele Instanzen der Komponente Sie verwenden. Weitere Informationen finden Sie in [Kapitel 18, „Barrierefreie Inhalte erstellen“ in *Flash verwenden*](#).

RadioButton-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer [RadioButton](#)-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden: `groupName`, `label`, `labelPlacement`, `selected` und `value`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur RadioButton-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

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

Erstellen von Anwendungen mit der RadioButton-Komponente

Im folgenden Verfahren wird beschrieben, wie Sie einer Anwendung beim Authoring RadioButton-Komponenten hinzufügen. In diesem Beispiel werden Optionsfelder für eine Ja-Nein-Frage verwendet. Die Daten der RadioButton-Komponente werden in einer TextArea-Komponente angezeigt.

So erstellen Sie eine Anwendung mit der RadioButton-Komponente:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie zwei RadioButton-Komponenten aus dem 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 aus dem 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“, um die Anwendung auszuführen.

Im folgenden Beispiel werden mithilfe von ActionScript drei Optionsfelder für die Farben Rot, Blau und Grün erstellt, und es wird ein graues Feld gezeichnet. Die `value`-Eigenschaft jeder RadioButton-Komponente legt den Hexadezimalwert für die mit dem jeweiligen Optionsfeld verknüpfte Farbe fest. Wenn ein Benutzer auf eines der drei Optionsfelder klickt, ruft die `clickHandler()`-Funktion die Methode `drawBox()` auf und übergibt die Farbe der `value`-Eigenschaft der entsprechenden RadioButton-Komponente, um das Feld farbig darzustellen.

So erstellen Sie mithilfe von ActionScript-Code ein Optionsfeld:

1. Erstellen Sie eine neue Flash-Datei (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“, um die Anwendung auszuführen.

Weitere Informationen finden Sie im Abschnitt über die RadioButton-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

ScrollPane-Komponente

Mit einer [ScrollPane](#)-Komponente (einem Bildlaufraster) kann Inhalt angezeigt werden, der für den Bereich, in den er geladen wird, zu groß ist. Wenn beispielsweise in einer Anwendung nur ein kleiner Bereich für ein großes Bild zur Verfügung steht, können Sie es in ein Bildlaufraster laden. Das Bildlaufraster akzeptiert Movieclips sowie JPEG-, PNG-, GIF- und SWF-Dateien.

Komponenten wie die ScrollPane- und die UILoader-Komponente verfügen über `complete`-Ereignisse, anhand derer Sie feststellen können, wenn Inhalte vollständig geladen wurden. Wenn Sie die Eigenschaften für den Inhalt einer ScrollPane- oder UILoader-Komponente festlegen möchten, fragen Sie das Ereignis `complete` ab und legen Sie die Eigenschaft in der Ereignisprozedur fest. 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 Bildlaufraster 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 über die ScrollPane-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

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:

Taste	Beschreibung
Pfeil nach unten	Der Inhalt wird vertikal um eine Zeile nach oben verschoben.
Pfeil nach oben	Der Inhalt wird vertikal um eine Zeile nach unten verschoben.
Ende	Der Inhalt wird an den unteren Rand des Bildlauffensters verschoben.
Pfeil nach links	Der Inhalt wird horizontal um eine Zeile nach rechts verschoben.
Pfeil nach rechts	Der Inhalt wird horizontal um eine Zeile nach links verschoben.
Pos 1	Der Inhalt wird an den oberen Rand des Bildlauffensters verschoben.
Ende	Der Inhalt wird an den unteren Rand des Bildlauffensters verschoben.
Bild ab	Der Inhalt wird vertikal um eine Seite nach oben verschoben.
Bild auf	Der Inhalt wird vertikal um eine Seite 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 Inhalt mit der Maus ziehen, wenn die Eigenschaft `scrollDrag` den Wert `true` aufweist. Das Einblenden des handförmigen Zeigers über dem Inhalt zeigt an, dass der Benutzer den Inhalt ziehen kann. Anders als bei den meisten anderen Steuerelementen tritt eine Aktion ein, wenn die Maustaste gedrückt wird, und wird fortgesetzt, bis die Maustaste losgelassen wird. Wenn der Inhalt über gültige Tabstopps enthält, müssen Sie `scrollDrag` auf `false` setzen. Andernfalls wird mit jeder Mausbetätigung der Inhalt verschoben.

ScrollPane-Parameter

Die folgenden Parameter können für jede Instanz einer [ScrollPane-Komponente](#) im [Eigenschafteninspektor](#) oder im [Komponenten-Inspektor](#) festgelegt werden `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `scrollDrag`, `source`, `verticalLineScrollSize`, `verticalPageScrollSize` und `verticalScrollPolicy`. Jeder dieser Parameter verfügt über eine entsprechende [ActionScript-Eigenschaft](#) desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur ScrollPane-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

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

Erstellen von Anwendungen 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.

So erstellen Sie eine Anwendung mit der ScrollPane-Komponente:

1. Erstellen Sie eine neue Flash-Datei (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“, um die Anwendung auszuführen.

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 ein Bildlauffenster erstellt, und es wird ein Movieclip (ein rotes Feld) in das Bildlauffenster geladen, das eine Größe von 150 Pixel mal 300 Pixel (B x H) hat.

So erstellen Sie mithilfe von ActionScript-Code ein Bildlauffenster:

1. Erstellen Sie eine neue Flash-Datei (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“, um die Anwendung auszuführen.

Slider-Komponente

Mit einer [Slider](#)-Komponente (einem Schieberegler) kann der Benutzer einen Wert auswählen, indem er einen grafisch dargestellten *Schieber* zwischen den beiden Endpunkten einer Spur, die einem Wertebereich entspricht, verstellt. Mit einem Schieberegler können Sie einen Benutzer zum Beispiel die Angabe eines Prozentwerts ermöglichen. Sie können auch ActionScript verwenden, damit der Wert des Schiebereglers das Verhalten eines zweiten Objekts beeinflusst. So könnten Sie den Schieberegler zum Beispiel einem Bild zuordnen und diese je nach der relativen Position des Schiebers verkleinern oder vergrößern.

Der aktuelle Wert des Schiebereglers wird durch die relative Position des Schiebers zwischen den Endpunkten der Spur bzw. zwischen dem Mindest- und dem Höchstwert des Schiebereglers bestimmt.

Der Schieberegler ermöglicht die Verwendung eines durchgehenden Wertebereichs zwischen dem Mindest- und dem Höchstwert, Sie können jedoch auch den Parameter `snapInterval` einstellen, um Intervalle zwischen dem Mindest- und dem Höchstwert anzugeben. Ein Schieberegler kann in bestimmten Abständen entlang der Spur Markierungen anzeigen, die unabhängig von den zugewiesenen Werten des Schiebereglers sind.

Der Schieberegler ist standardmäßig horizontal ausgerichtet; wenn Sie den Parameter `direction` jedoch auf **vertical** stellen, wird der Regler senkrecht angezeigt. Die Spur des Schiebereglers verläuft von einem Ende zum anderen, und die Markierungen werden von links nach rechts direkt oberhalb der Spur angezeigt.

Benutzerinteraktion mit der Slider-Komponente

Slider-Instanzen mit Eingabefokus können mit den folgenden Tasten gesteuert werden:

Taste	Beschreibung
Pfeil nach rechts	Erhöht den zugeordneten Wert für einen horizontalen Schieberegler.
Pfeil nach oben	Erhöht den zugeordneten Wert für einen vertikalen Schieberegler.
Pfeil nach links	Verringert den zugeordneten Wert für einen horizontalen Schieberegler.
Pfeil nach unten	Verringert den zugeordneten Wert für einen vertikalen Schieberegler.
Umschalt+Tab	Der Fokus wird auf das vorherige Objekt verschoben.
Tabulator	Der Fokus wird auf das nächste Objekt verschoben.

Weitere Informationen zur Steuerung des Fokus finden Sie in den Abschnitten über die Schnittstelle [IFocusManager](#) und über [FocusManager](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0* sowie unter „Verwenden des FocusManagers“ auf Seite 57.

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

Slider-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer [Slider](#)-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden: `direction`, `liveDragging`, `maximum`, `minimum`, `snapInterval`, `tickInterval` und `value`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur Slider-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Erstellen von Anwendungen mit der Slider-Komponente

Im folgenden Beispiel wird eine Slider-Instanz erstellt, mit der ein Benutzer angeben kann, wie zufrieden er mit einem hypothetischen Ereignis ist. Der Benutzer verschiebt den Schieberegler nach rechts oder links, um größere oder geringere Zufriedenheit auszudrücken.

So erstellen Sie eine Anwendung mit der Slider-Komponente:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine Label-Komponente aus dem Bedienfeld „Komponenten“ auf die Mitte der Bühne.
 - Geben Sie ihr den Instanznamen **valueLabel**.
 - Weisen Sie dem `text`-ParameterAssign den Wert **0 Prozent** zu.
3. Ziehen Sie eine Slider-Komponente aus dem Bedienfeld „Komponenten“ und platzieren Sie sie unter `value_lbl`.
 - Geben Sie ihr den Instanznamen **aSlider**.
 - Weisen Sie ihr eine Breite (B) von **200** zu.
 - Weisen Sie ihr eine Höhe (H) von **10** zu.
 - Weisen Sie dem `maximum`-Parameter den Wert **100** zu.
 - Weisen Sie den Parametern `snapInterval` und `tickInterval` jeweils den Wert **10** zu.
4. Ziehen Sie eine weitere Label-Instanz aus dem Bedienfeld „Bibliothek“ und platzieren Sie sie unterhalb von `aSlider`.
 - Geben Sie ihr den Instanznamen **promptLabel**.
 - Weisen Sie ihr eine Breite (B) von **250** zu.
 - Weisen Sie ihr eine Höhe (H) von **22** zu.
 - Geben Sie für den `text`-Parameter den Text **Please indicate your level of satisfaction** 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“.

In diesem Beispiel aktualisiert ein Listener für das `SliderEvent.CHANGE`-Ereignis die Eigenschaft `text` von `valueLabel`, um den Prozentwert, der der Position des Schiebers entspricht, anzuzeigen, wenn der Schieber von einem Intervall zum nächsten bewegt wird.

Im folgenden Beispiel wird ein Schieberegler mithilfe von ActionScript erstellt. Im Beispiel wird das Bild einer Blume heruntergeladen, und mit dem Schieberegler kann der Benutzer das Bild blasser oder heller anzeigen. Dies wird durch Ändern der `alpha`-Eigenschaft erreicht, die dem Wert des Schiebereglers entspricht.

So erstellen Sie mithilfe von ActionScript eine Anwendung mit der Slider-Komponente:

1. Erstellen Sie eine neue Flash-Datei (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 Instanzen 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“, um die Anwendung auszuführen.
5. Bewegen Sie den Schieberegler nach links, um das Bild blasser werden zu lassen, oder nach rechts, um es aufzuhellen.

TextArea-Komponente

Die [TextArea](#)-Komponente (ein Textfeld) ist ein Wrapper für das native ActionScript-[TextField](#)-Objekt. Mit der TextArea-Komponente können Sie Text anzeigen und Texteingaben verarbeiten, falls die Eigenschaft `editable` auf den Wert `true` gesetzt wurde. Die Komponente kann mehrere Textzeilen anzeigen oder entgegennehmen. Falls die Eigenschaft `wordWrap` auf den Wert `true` gesetzt wurde, werden lange Textzeilen umgebrochen. Mit der Eigenschaft `restrict` können Sie verhindern, dass der Benutzer bestimmte Zeichen eingibt, und mit `maxChars` können Sie die Höchstzahl der Zeichen, die der Benutzer eingeben kann, festlegen. Wenn der Text über die horizontalen oder vertikalen Grenzen des Textfelds hinausreicht, werden automatisch horizontale und vertikale Bildlaufleisten angezeigt, sofern die dazugehörigen Eigenschaften, `horizontalScrollPolicy` und `verticalScrollPolicy`, nicht auf `off` gesetzt wurden.

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.

Sie können den `textFormat`-Stil mit der Methode `setStyle()` festlegen, um den Stil des Textes in der `TextArea`-Instanz zu ändern. Sie können eine `TextArea`-Komponente auch mit HTML formatieren, indem Sie die `htmlText`-Eigenschaft in ActionScript verwenden, und Sie können die Eigenschaft `displayAsPassword` auf den Wert `true` setzen, um den Text als Sternchen anzuzeigen. Wenn Sie die Eigenschaft `condenseWhite` auf den Wert `true` setzen, entfernt Flash überflüssigen weißen Leerraum im neuen Text, der durch Leerzeichen, Zeilenumbrüche usw. verursacht wird. Dies hat keinen Einfluss auf Text, der sich bereits in dem Steuerelement befindet.

Benutzerinteraktion mit der `TextArea`-Komponente

`TextArea`-Komponenten können in einer Anwendung aktiviert und deaktiviert werden. In deaktiviertem Zustand können Eingaben über die Maus oder die Tastatur nicht empfangen werden. 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:

Taste	Beschreibung
Pfeiltasten	Bewegt die Einfügemarke innerhalb des Texts nach oben, unten, links oder rechts, falls der Text bearbeitbar ist.
Bild ab	Bewegt die Einfügemarke an das Ende des Texts, falls der Text bearbeitbar ist.
Bild auf	Bewegt die Einfügemarke an den Anfang des Texts, falls der Text bearbeitbar ist.
Umschalt+Tab	Bewegt den Fokus zum vorigen Objekt in der Tabulatorschleife.
Tabulator	Bewegt den Fokus zum nächsten Objekt in der Tabulatorschleife.

Weitere Informationen zur Steuerung des Fokus finden Sie im Abschnitt über die [FocusManager](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3* und unter „[Verwenden des FocusManagers](#)“ auf Seite 57.

TextArea-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer [TextArea](#)-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden: `condenseWhite`, `editable`, `horizontalScrollPolicy`, `maxChars`, `restrict`, `text`, `verticalScrollPolicy` und `wordwrap`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur TextArea-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Die Änderungen an den Parametern der einzelnen TextArea-Instanzen, die beim Authoring im Eigenschafteninspektor oder im 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 über die TextArea-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Erstellen von Anwendungen mit der TextArea-Komponente

In der folgenden Anleitung wird erläutert, wie während des Authorings eine TextArea-Komponente einer Anwendung hinzugefügt wird. Im Beispiel wird eine 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.

So erstellen Sie eine Anwendung mit der TextArea-Komponente:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine TextArea-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz **aTa**. Ändern Sie die Standardeinstellungen der Parameter nicht.
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**. Ändern Sie die Standardeinstellungen der Parameter nicht.

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 Zeichen, die in das Textfeld aTa eingegeben werden können, auf Kleinbuchstaben, Komma, Apostroph und Leerzeichen beschränkt. Außerdem werden die Ereignisprozeduren für die Ereignisse change, KEY_FOCUS_CHANGE und MOUSE_FOCUS_CHANGE für die TextArea-Komponente aTa eingestellt. 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. Mit der k_m_fHandler()-Funktion für die Ereignisse KEY_FOCUS_CHANGE und MOUSE_FOCUS_CHANGE wird verhindert, dass Sie mit der Tabulatortaste zum nächsten Feld gehen können, wenn Sie noch keinen Text eingegeben haben. Um dies zu erreichen, wird das Standardverhalten unterdrückt.

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

Wenn Sie die Tabulatortaste drücken, um den Fokus zum zweiten Textfeld zu bewegen, ohne dass Sie Text eingegeben haben, sollte eine Fehlermeldung angezeigt werden, und der Fokus bleibt beim ersten Textfeld. Wenn Sie Text in das erste Textfeld eingeben, wird er ebenfalls im zweiten Textfeld angezeigt.

Im folgenden Beispiel wird eine TextArea-Komponente mithilfe von ActionScript 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.

So erstellen Sie mithilfe von ActionScript-Code ein Textfeld:

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

In diesem Beispiel wird die Eigenschaft `htmlText` verwendet, um die HTML-Attribute für Fett-Formatierung und Unterstreichen auf einen Textblock anzuwenden und den Text im Textfeld `a_ta` anzuzeigen. Im Beispiel wird außerdem die Eigenschaft `condenseWhite` auf `true` gesetzt, um Leerräume innerhalb des Textblocks zu verhindern. Mit der Methode `setSize()` werden Höhe und Breite des Textfelds eingestellt, und die Methode `move()` legt die Position fest. Mit der Methode `addChild()` wird die TextArea-Instanz der Bühne hinzugefügt.

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

TextInput-Komponente

Die [TextInput](#)-Komponente (ein Texteingabefeld) ist eine einzeilige Textkomponente, die das native ActionScript-Objekt „TextField“ erweitert. Informationen zu Textfeldern mit mehreren Zeilen finden Sie unter [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 die Fehlermeldung anzeigen, dass die richtige Anzahl an Zeichen in dem Feld eingegeben werden muss.

Sie können die `textFormat`-Eigenschaft mit der Methode `setStyle()` festlegen, um den Stil des Textes in der `TextInput`-Instanz zu 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:

Taste	Beschreibung
Pfeiltasten	Verschieben die Einfügemarke um ein Zeichen nach links oder rechts.
Umschalt+Tab	Der Fokus wird auf das vorherige Objekt verschoben.
Tabulator	Der Fokus wird auf das nächste Objekt verschoben.

Weitere Informationen zur Steuerung des Fokus finden Sie im Abschnitt über die [FocusManager](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3* und unter „Verwenden des `FocusManagers`“ auf Seite 57.

Die Änderungen an den Parametern der einzelnen `TextInput`-Instanzen, die beim Authoring im Eigenschafteninspektor oder im Komponenteninspektor 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 verfügbar machen.

TextInput-Parameter

Sie können die folgenden Authoring-Parameter für jede Instanz der `TextInput`-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festlegen: `editable`, `displayAsPassword`, `maxChars`, `restrict` und `text`. Jeder dieser Parameter verfügt über eine entsprechende `ActionScript`-Eigenschaft desselben Namens. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur `TextInput`-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

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 über die TextInput-Klasse im *[Komponenten-Referenzhandbuch für ActionScript 3.0](#)*.

Erstellen von Anwendungen mit der TextInput-Komponente

In der folgenden Anleitung wird erläutert, wie eine TextInput-Komponente einer Anwendung hinzugefügt wird. Im Beispiel werden zwei Texteingabefelder für die Eingabe und Bestätigung eines Kennworts verwendet. Es wird ein Ereignis-Listener verwendet, der feststellt, ob mindestens acht Zeichen eingegeben wurden, und ob der Text in beiden Feldern übereinstimmt.

So erstellen Sie eine Anwendung mit der TextInput-Komponente:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine Label-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und geben Sie im Eigenschafteninspektor die folgenden Werte ein:
 - Geben Sie den Instanznamen **pwdLabel** ein.
 - Geben Sie für die Breite (B) den Wert **100** ein.
 - Geben Sie für „X“ den Wert **50** ein.
 - Geben Sie für „Y“ den Wert **150** ein.
 - Geben Sie im Bereich „Parameter“ den Wert **Password:** für den Parameter „text“ 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 für die Breite (B) den Wert **100** ein.
 - Geben Sie für „X“ den Wert **50** ein.
 - Geben Sie für „Y“ den Wert **200** ein.
 - Geben Sie im Bereich „Parameter“ den Wert **Confirm Password** für den Parameter „text“ 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 für die 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. Dies bewirkt, dass der in das Textfeld eingegebene Wert mit Sternchen maskiert wird.
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 für die Breite (B) den Wert **150** ein.
 - Geben Sie für „X“ den Wert **190** ein.
 - Geben Sie für „Y“ den Wert **200** ein.
 - Doppelklicken Sie im Bereich „Parameter“ auf den Wert für den Parameter `displayAsPassword` und wählen Sie **true** aus. Dies bewirkt, dass der in das Textfeld eingegebene Wert mit Sternchen maskiert wird.
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 die beiden Eingaben für das Kennwort nicht übereinstimmen oder der Benutzer weniger als acht Zeichen eingibt, wird im Beispiel die folgende Meldung angezeigt: „Password is incorrect. Please reenter it.“ 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“.

Im folgenden Beispiel wird eine TextInput-Komponente mithilfe von ActionScript erstellt. Im Beispiel wird auch eine Label-Komponente (eine Bezeichnung) erstellt, die den Benutzer auffordert, seinen Namen einzugeben. Im Beispiel wird die Eigenschaft `restrict` der Komponente festgelegt, sodass nur Großbuchstaben und Kleinbuchstaben, ein Punkt und ein Leerzeichen eingegeben werden können. Außerdem wird ein `TextFormat`-Objekt erstellt, das verwendet wird, um den Text in den Label- und TextInput-Komponenten zu formatieren.

So erstellen Sie mithilfe von ActionScript-Code ein Texteingabefeld:

1. Erstellen Sie eine neue Flash-Datei (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:

```
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“, um die Anwendung auszuführen.

TileList-Komponente

Die [TileList](#)-Komponente besteht aus einer Liste, deren Zeilen und Spalten von einem Datenprovider mit Daten versehen werden. Als *Element* wird eine Dateneinheit in einer Zelle der TileList bezeichnet. Ein Element, das aus dem Datenprovider stammt, verfügt normalerweise über eine `label`-Eigenschaft und eine `source`-Eigenschaft.

Die `label`-Eigenschaft identifiziert den Inhalt, der in der Zelle angezeigt wird, und die `source`-Eigenschaft gibt einen Wert dafür an.

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. `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 TileList-Komponente stellt jede Zelle mithilfe eines Sprite-Objekts dar, welches das `ICellRenderer`-Interface implementiert. Sie können diesen Renderer mit der Eigenschaft `cellRenderer` der TileList-Komponente darstellen. Der Standard-CellRenderer der TileList-Komponente ist `ImageCell`, womit ein Bild (Klasse, Bitmap, Instanz oder URL) und eine optionale Bezeichnung angezeigt werden. Die Bezeichnung ist eine einzelne Zeile, die immer am unteren Rand der Zelle ausgerichtet ist. Sie können in eine TileList-Komponente nur in einer Richtung einen Bildlauf ausführen.

Wenn eine TileList-Instanz den Fokus hat, können Sie die Elemente der Liste auch mithilfe der folgenden Tasten ansteuern:

Taste	Beschreibung
Pfeil nach oben und Pfeil nach unten	Ermöglicht es Ihnen sich, innerhalb einer Spalte nach oben und unten zu bewegen. Wenn die Eigenschaft <code>allowMultipleSelection</code> den Wert <code>true</code> aufweist, können Sie diese Tasten in Kombination mit der Umschalttaste verwenden, um mehrere Tasten auszuwählen.
Pfeil nach links und Pfeil nach rechts	Ermöglicht es Ihnen sich, innerhalb einer Zeile nach links und rechts zu bewegen. Wenn die Eigenschaft <code>allowMultipleSelection</code> den Wert <code>true</code> aufweist, können Sie diese Tasten in Kombination mit der Umschalttaste verwenden, um mehrere Tasten auszuwählen.

Taste	Beschreibung
Pos 1	Wählt die erste Zelle in der 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 Zelle in der 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 die Eigenschaft <code>allowMultipleSelection</code> auf <code>true</code> gesetzt ist, können Sie diese Taste gedrückt halten, um durch Klicken mehrere Zellen in beliebiger Reihenfolge auszuwählen.

Wenn Sie die `TileList`-Komponente einer Anwendung hinzufügen, können Sie sie für ein Bildschirmlesegerät zugänglich machen, indem Sie die folgenden `ActionScript`-Codezeilen hinzufügen:

```
import fl.accessibility.TileListAccImpl;

TileListAccImpl.enableAccessibility();
```

Diese Anweisung ist unabhängig von der Anzahl der Instanzen nur einmal pro Komponente erforderlich. Weitere Informationen finden Sie in [Kapitel 18, „Barrierefreie Inhalte erstellen“](#), in *Flash verwenden*.

TileList-Parameter

Die folgenden `Authoring`-Parameter können für jede Instanz einer `TileList`-Komponente im `Eigenschafteninspektor` oder im `Komponenten-Inspektor` festgelegt werden:

`allowMultipleSelection`, `columnCount`, `columnWidth`, `dataProvider`, `direction`, `horizontalScrollLineSize`, `horizontalScrollPageSize`, `labels`, `rowCount`, `rowHeight` und `ScrollPolicy`, `verticalScrollLineSize`, `verticalScrollPageSize`. Jeder dieser Parameter verfügt über eine entsprechende `ActionScript`-Eigenschaft desselben Namens. Informationen zur Verwendung des `dataProvider`-Parameters finden Sie unter „[Verwenden des Parameters „dataProvider“](#)“ auf Seite 61.

Sie können zusätzliche Optionen für `TileList`-Instanzen mit den Methoden, Eigenschaften und Ereignissen von `ActionScript` festlegen. Weitere Informationen finden Sie im Abschnitt über die `TileList`-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Erstellen von Anwendungen mit der TileList-Komponente

In diesem Beispiel werden Movieclips verwendet, um eine TileList-Komponente mit einem Array von Farben zu füllen.

So erstellen Sie eine Anwendung mit der TileList-Komponente:

1. Erstellen Sie eine neue Flash-Datei (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.

Im nächsten Beispiel wird eine [TileList](#)-Instanz dynamisch erstellt, und es werden Instanzen der [ColorPicker](#)-, [ComboBox](#)-, [NumericStepper](#)- und [CheckBox](#)-Komponenten hinzugefügt. Es wird ein Array erstellt, das Bezeichnungen und Namen der anzuzeigenden Komponenten enthält, und dieses Array (dp) wird dann der Eigenschaft `dataProvider` der [TileList](#)-Komponente zugewiesen. Die Eigenschaften `columnWidth` und `rowHeight` sowie die Methode `setSize()` werden für das Layout der [TileList](#)-Komponente verwendet, die Methode `move()` für die Positionierung auf der Bühne und der Stil `contentPadding` für das Festlegen des Abstands zwischen den Rändern der [TileList](#)-Instanz und dem Inhalt, und mit der Methode `sortItemsOn()` wird der Inhalt anhand der Bezeichnungen sortiert.

So erstellen Sie mithilfe von ActionScript-Code eine [TileList](#)-Komponente:

1. Erstellen Sie eine neue Flash-Datei (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:

```
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.

UILoader-Komponente

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

Sie können den Inhalt des UILoaders skalieren oder die Größe des UILoaders selbst ändern, um sie an die Größe des Inhalts anzupassen. Als Standard wird der Inhalt so skaliert, dass er in den UILoader 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 für das Laden des Inhalts in den UILoader eine Position festlegen, müssen Sie die X- und Y-Koordinaten als 0,0 angeben.

Benutzerinteraktion mit der UILoader-Komponente

Die 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 zur Steuerung des Fokus finden Sie im Abschnitt über die [FocusManager](#)-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3* und unter „Verwenden des FocusManagers“ auf Seite 57.

UILoader-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer [UILoader](#)-Komponente im Eigenschafteninspektor oder im Komponenteninspektor festgelegt werden: `autoLoad`, `maintainAspectRatio`, `source` und `scaleContent`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens.

Eine Live-Vorschau der einzelnen UILoader-Instanzen spiegelt die Änderungen wieder, die im Eigenschafteninspektor 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 über die UILoader-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Erstellen von Anwendungen mit der UILoader-Komponente

Im folgenden Verfahren wird erklärt, wie Sie einer Anwendung beim Authoring die UILoader-Komponente hinzufügen. In diesem Beispiel lädt der Loader eine GIF-Bild eines Logos.

So erstellen Sie eine Anwendung mit der UILoader-Komponente:

1. Erstellen Sie eine neue Flash-Datei (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 Komponenten-Inspektor aus und geben Sie für den `source`-Parameter den Wert <http://www.helpexamples.com/images/logo.gif> ein.

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

So erstellen Sie mithilfe von ActionScript-Code eine UILoader-Komponenteninstanz:

1. Erstellen Sie eine neue Flash-Datei (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/imagel.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“.

UIScrollBar-Komponente

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

Wenn die Bildlaufleiste kleiner als die Größe der beiden Bildlaufpfeile ist, wird sie nicht korrekt angezeigt. Einer der Pfeile wird von dem anderen Pfeil 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.

Weitere Informationen zu Textfeldern finden Sie im Abschnitt über die TextField-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

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.

UIScrollBar-Parameter

Die folgenden Authoring-Parameter können für jede Instanz einer **UIScrollBar**-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden: `direction` und `scrollTargetName`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens.

Sie können zusätzliche Optionen für UIScrollBar-Instanzen mit den Methoden, Eigenschaften und Ereignissen der Klasse mithilfe von ActionScript festlegen. Weitere Informationen finden Sie im Abschnitt über die UIScrollBar-Klasse im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Erstellen von Anwendungen mit der UIScrollBar-Komponente

Im Folgenden wird beschrieben, wie Sie einer Anwendung beim Authoring eine UIScrollBar-Komponente hinzufügen.

So erstellen Sie eine Anwendung mit der UIScrollBar-Komponente:

1. Erstellen Sie eine neue Flash-Datei (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 ihr den Instanznamen **mySb**.

Die Eigenschaft `scrollTargetName` der Komponente wird automatisch mit dem Instanznamen des Textfelds im Eigenschafteninspektor und Komponenten-Inspektor 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“.

Sie können eine `UIScrollBar`-Instanz auch mithilfe von `ActionScript` erstellen und zur Laufzeit mit einem Textfeld verknüpfen. Im folgenden Beispiel wird eine horizontal ausgerichtete `UIScrollBar`-Instanz erstellt und am unteren Rand eines Textfelds mit dem Namen `myTxt`, das über eine URL mit Text gefüllt wird, verankert. Im Beispiel wird auch die Größe der Bildlaufleiste festgelegt, um die an die Größe des Textfelds anzupassen:

So erstellen Sie mithilfe von `ActionScript`-Code eine `UIScrollBar`-Komponenteninstanz:

1. Erstellen Sie eine neue Flash-Datei (`ActionScript 3.0`).
2. Ziehen Sie eine `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“.

Anpassen von UI-Komponenten

In diesem Kapitel wird die benutzerdefinierte Anpassung der Flash ActionScript 3.0 UI-Komponenten beschrieben. Es enthält die folgenden Abschnitte:

Anpassung von UI-Komponenten	160
Stile	160
Erstellen neuer Skins	168
Anpassen der Button-Komponente	170
Anpassen der CheckBox-Komponente	174
ColorPicker-Komponente anpassen	176
ComboBox-Komponente anpassen	178
DataGrid-Komponente anpassen	181
Label-Komponente anpassen	187
List-Komponente anpassen	189
NumericStepper-Komponente anpassen	192
ProgressBar-Komponente anpassen	195
RadioButton-Komponente anpassen	197
ScrollPane-Komponente anpassen	200
Slider-Komponente anpassen	202
TextArea-Komponente anpassen	204
TextInput-Komponente anpassen	207
TileList-Komponente anpassen	210
UILoader-Komponente anpassen	212
UIScrollBar-Komponente anpassen	213

Informationen zum Anpassen der FLVPlayback-Komponente finden Sie in [Kapitel 5](#), „Die FLVPlayback-Komponente“.

Anpassung von UI-Komponenten

Sie können das Erscheinungsbild der Komponenten in Ihren Anwendungen anpassen, indem Sie eines oder mehrere der folgenden Elemente ändern:

Stile Jede Komponente verfügt über eine Reihe von Stilen, die Sie einstellen können, um anzugeben, welche Werte Flash verwendet, um das Erscheinungsbild der Komponente darzustellen. Stile geben im Allgemeinen Skins und Symbole an, die für die verschiedenen Zustände einer Komponente verwendet werden, und legen auch fest, welche Werte für die Textformatierung und Zellauffüllung verwendet werden.

Skins Eine *Skin* besteht aus mehreren Symbolen, die das grafische Erscheinungsbild einer Komponente in einem bestimmten Zustand ausmachen. Während ein Stil festlegt, welche Skin zu verwenden ist, ist eine Skin ein grafisches Element, das Flash verwendet, um die Komponente zu zeichnen. *Skinning* bezeichnet das Verfahren, mit dem das Aussehen einer Komponente geändert wird, indem die ursprünglichen Grafiken geändert oder ersetzt werden.

HINWEIS

Das Standardaussehen der ActionScript 3.0-Komponenten kann man sich als Schema (Aeon Halo) vorstellen, diese Skins sind jedoch in die Komponenten integriert. Die ActionScript 3.0-Komponenten unterstützen nicht die externen Schemadateien, die von ActionScript 2.0-Komponenten unterstützt wurden.

Stile

Die Stile einer Komponente legen im Allgemeinen Werte für die Skins, die Symbole, die Textformatierung und die Zellauffüllung fest, wenn Flash die Komponente in verschiedenen Zuständen zeichnet. So zeichnet Flash eine Schaltfläche mit einer bestimmten Skin, um den Drücken-Zustand darzustellen, nachdem mit der Maus auf die Schaltfläche geklickt wurde, die sich von der Skin, die den normalen Zustand darstellt, unterscheidet. Für den deaktivierten Zustand (d. h. wenn die Eigenschaft `enabled` den Wert `false` hat), wird ebenfalls eine andere Skin verwendet.

Sie können Stile für Komponenten auf Dokument-, Klassen- und Instanzebene festlegen. Zusätzlich können einige Stileigenschaften auch von einer übergeordneten Komponente übernommen werden. Die List-Komponente übernimmt zum Beispiel ScrollBar-Stile von der BaseScrollPane-Komponente.

Sie können Stile auf verschiedene Weisen festlegen, um eine Komponente anzupassen:

- Festlegen von Stilen für eine Komponenteninstanz.
Sie können die Farb- und Texteingenschaften 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 Sie allen Komponenten eines bestimmten Typ, zum Beispiel allen Kontrollkästchen oder allen Schaltflächen, in einem Dokument ein einheitliches Erscheinungsbild geben möchten, können Sie Stile auf 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.

Arbeiten mit Stileinstellungen

Nachstehend sind einige wichtige Aspekte für die Verwendung von Stilen aufgeführt:

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

Vorrang Wenn ein Stil einer Komponente auf mehrere Weisen festgelegt wurde, verwendet Flash den ersten Stil in der Reihenfolge, die den Vorrang bestimmt. 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 nach dem Stil in der Komponenteninstanz.
4. Flash sucht nach einer globalen Einstellung im StyleManager.
5. Wenn die Eigenschaft immer noch nicht definiert ist, lautet der Eigenschaftswert `undefined`.

Zugriff auf die Standardstile einer Komponente

Mit der statischen Methode `getDefaultStyles()` für die Komponentenklasse haben Sie Zugriff auf die Standardstile einer Komponente. 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.getDefaultStyles();
trace(styleObj.buttonWidth); // 21
trace(styleObj.downArrowDownSkin); // downArrowDownSkin
```

Festlegen und Abrufen von Stilen für eine Komponenteninstanz

Jede UI-Komponenteninstanz ruft die Methoden `setStyle()` und `getStyle()` direkt auf, 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 einen `Object`-Typ zurückgibt, da sie mehrere Stile mit verschiedenen Datentypen zurückgeben kann. Mit dem folgenden Code wird zum Beispiel der Schriftstil für eine `TextArea`-Instanz (`aTa`) festgelegt und dann mit der `getStyle()`-Methode abgerufen. Im Beispiel wird der zurückgegebene Wert in ein `TextFormat`-Objekt umgewandelt, damit es einer `TextFormat`-Variablen zugewiesen werden kann. Ohne die Umwandlung würde der Compiler einen Fehler zurückgeben, weil versucht wurde, eine `Object`-Variable in eine `TextFormat`-Variable umzuwandeln.

```
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 mit dem TextFormat-Objekt

Verwenden Sie das TextFormat-Objekt, um Text für eine Komponenteninstanz zu formatieren. Das TextFormat-Objekt verfügt über Eigenschaften, die es ermöglichen, verschiedene Textmerkmale festzulegen, darunter `bold`, `bullet`, `color`, `font`, `italic` und `size`. Sie können diese Eigenschaften im TextFormat-Objekt festlegen und dann die Methode `setStyle()` aufrufen, um sie auf eine Komponenteninstanz anzuwenden. Mit dem folgenden Code werden zum Beispiel die Eigenschaften `font`, `size` und `bold` eines TextFormat-Objekts festgelegt und dann 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 Auswirkung dieser Einstellungen auf eine Schaltfläche mit der Bezeichnung „Senden“:



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 einstellen, desto langsamer wird die Komponente zur Laufzeit wiedergegeben.

Festlegen eines Stils für alle Instanzen einer Komponente

Sie können einen Stil für alle Instanzen einer Komponentenkategorie festlegen, indem Sie die Methode `setComponentStyle()` der `StyleManager`-Klasse verwenden. 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 Button-Instanzen, die Sie danach auf die Bühne ziehen, haben rote Bezeichnungen.

Festlegen eines Stils für alle Komponenten

Sie können einen Stil für alle Komponenten festlegen, indem Sie die Methode `setStyle()` der `StyleManager`-Klasse verwenden.

So legen Sie einen Stil für alle Komponenten fest:

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, und 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 (Xc11nt 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 sollte auf der Schaltfläche und in der Liste in Rot angezeigt werden.

Skins

Das Erscheinungsbild einer Komponenten setzt sich aus grafischen Elementen wie einem Umriss, einer Füllfarbe, Symbolen und sogar anderen Komponenten zusammen. Eine ComboBox-Komponente enthält zum Beispiel eine List-Komponente, und eine List-Komponente enthält eine ScrollBar-Komponente. Zusammen bilden die grafischen Elemente das Erscheinungsbild der ComboBox-Komponente. Das Erscheinungsbild einer Komponente basiert jedoch auf ihrem aktuellen Zustand. Eine CheckBox-Komponente (ein Kontrollkästchen) sieht in Ihrer Anwendung ohne ihre Bezeichnung ähnlich wie hier abgebildet aus:



Ein Kontrollkästchen in seinem normalen Up-Status

Wenn Sie mit der Maus auf das Kontrollkästchen klicken und die Maustaste gedrückt halten, ändert sich das Erscheinungsbild:



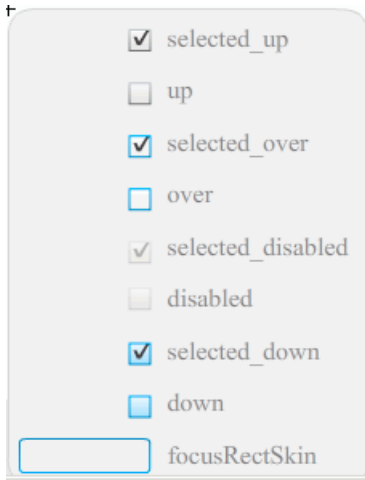
Ein Kontrollkästchen in seinem normalen Down-Status

Wenn Sie die Maustaste loslassen, wird das Kontrollkästchen in seinem ursprünglichen Aussehen angezeigt, enthält jetzt jedoch ein Prüfhäkchen, um anzuzeigen, dass es ausgewählt wurde.



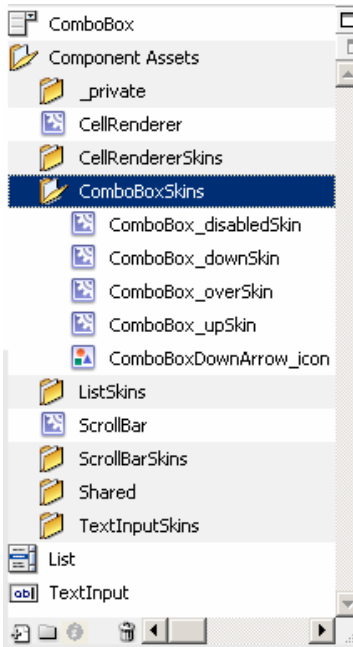
Ein Kontrollkästchen im ausgewählten Zustand

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



Die Skins einer CheckBox-Komponente

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 zum Beispiel eine ComboBox-Komponente auf die Bühne ziehen, finden Sie im Bedienfeld „Bibliothek“ die folgenden Elemente:



Bedienfeld „Bibliothek“ für die ComboBox-Komponente

Neben der ComboBox-Komponente 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 „Shared Assets“, der Elemente enthält, die diese Komponenten gemeinsam haben. 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 dargestellt:



ComboBox_downSkin

Erstellen neuer Skins

Wenn Sie ein neues Erscheinungsbild für eine Komponente in Ihrem Dokument erstellen möchten, bearbeiten Sie die Skins der Komponente, um ihr Aussehen zu ändern. 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 deren Elemente im Symbolbearbeitungsmodus zu öffnen. Stellen Sie die Vergrößerung auf 400 % oder höher und bearbeiten Sie das Symbol, um sein Erscheinungsbild zu ändern. Wenn Sie damit fertig sind, gelten die Änderungen für alle Instanzen der Komponente im Dokument. Alternativ dazu können Sie im Bedienfeld „Bibliothek“ auf eine bestimmte Skin doppelklicken, um diese auf der Bühne im Symbolbearbeitungsmodus zu öffnen.

Zum Ändern von Komponentenskins stehen die folgenden Möglichkeiten zur Verfügung:

- Erstellen Sie eine neue Skin für alle Instanzen
- Erstellen Sie neue Skins für einige Instanzen

Erstellen einer Skin für alle Instanzen

Wenn Sie die Skin einer Komponente bearbeiten, wird standardmäßig das Erscheinungsbild aller Instanzen dieser Komponente im Dokument geändert. Wenn Sie verschiedene Erscheinungsbilder für dieselbe Komponente erstellen möchten, müssen Sie die Skins, die Sie ändern möchten, duplizieren und ihnen andere Namen geben, sie bearbeiten und dann die gewünschten Stile einstellen, um sie anzuwenden. Weitere Informationen finden Sie unter [„Erstellen von Skins für einige Instanzen“ auf Seite 169](#).

In diesem Kapitel wird beschrieben, wie Sie eine oder mehrere Skins für die einzelnen UI-Komponenten ändern können. 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 in das Bedienfeld „Bibliothek“ aus.
- Duplizieren Sie die Skin und geben Sie ihr einen eindeutigen Klassennamen.
- Bearbeiten Sie die Skin, bis sie das gewünschte Erscheinungsbild hat.
- Rufen Sie die `setStyle()`-Methode für die Komponenteninstanz auf, um die neue Skin dem Skinstil zuzuweisen.

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

So erstellen Sie eine neue Skin „selectedDownSkin“ für eine Schaltfläche:

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. Öffnen Sie durch Klicken mit der rechten Maustaste das Kontextmenü und wählen Sie „Duplizieren“.
6. Geben Sie im Dialogfeld „Symbol duplizieren“ einen eindeutigen Namen für die Skin ein, zum Beispiel **Button_mySelectedDownSkin**. Klicken Sie anschließend 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. Wenn die Warnung, dass keine Klassendefinition gefunden wurde, angezeigt wird, klicken Sie auf „OK“ und dann 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 in den Dokumentbearbeitungsmodus zurückzukehren.
12. Klicken Sie im Eigenschafteninspektor für jede Schaltfläche auf die Registerkarte „Parameter“ und setzen Sie den Parameter „toggle“ 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 die einzelnen Schaltflächen. Sie sehen, dass die für den gedrückten Zustand (ausgewählt und nicht ausgewählt) des Objekts „bButton“ das neue Skin-Symbol 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`, `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 Rand der Schaltfläche und definiert auch den Hit-Bereich (Kollisionsbereich) für die 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.

Stile für die Button-Komponente

Die Stile einer Schaltfläche legen im Allgemeinen Werte für die Skins, die Symbole, die Textformatierung und die Zellauffüllung der Schaltfläche fest, wenn die Komponente in verschiedenen Zuständen gezeichnet wird.

Im folgenden Verfahren werden zwei Button-Komponenten auf der Bühne platziert, und die Eigenschaft `emphasized` wird für beide Schaltflächen auf `true` gesetzt, wenn der Benutzer auf eine der Schaltflächen klickt. Außerdem wird der Stil `emphasizedSkin` für die zweite Schaltfläche auf den Stil `selectedOverSkin` gesetzt, wenn der Benutzer darauf klickt, sodass die beiden Schaltflächen unterschiedliche Skins für denselben Zustand aufweisen.

So ändern Sie den Stil „`emphasizedSkin`“ für eine Button-Komponente:

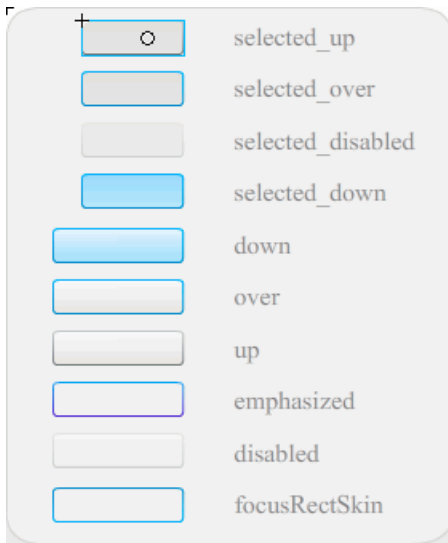
1. Erstellen Sie eine Flash-Datei (ActionScript 3.0).
2. Ziehen Sie nacheinander zwei Button-Komponenten auf die Bühne und geben Sie den Instanzen die Namen **aBtn** und **bBtn**. Geben Sie den Schaltflächen auf der Registerkarte „Parameter“ des Eigenschafteninspektors die Bezeichnungen **Button A** 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 Schaltfläche, um den Effekt des Stils `emphasizedSkin` auf beiden Schaltflächen zu sehen.

Skins für die Button-Komponente

Die Button-Komponente verwendet die folgenden Skins, die den unterschiedlichen Zuständen der Button-Komponente entsprechen. 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:

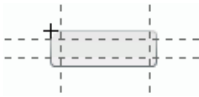


Skins der Button-Komponente

Aktivierte Schaltflächen werden im Over-Status angezeigt, wenn sich der Mauszeiger darüber befindet. Beim Klicken auf die Schaltfläche erhält diese den Eingabefokus und sie wird im Down-Status angezeigt. Beim Loslassen der Maustaste wird die Schaltfläche wieder im Over-Status angezeigt. Wenn der Zeiger von der Schaltfläche weg bewegt wird, während die Maustaste noch gedrückt ist, kehrt die Schaltfläche zu ihrem ursprünglichen Zustand 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 den Bedienvorgängen des Benutzers 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:



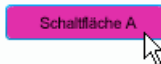
Jetzt können Sie die Skin mit den Flash-Authoring-Werkzeugen nach Ihren Vorstellungen beliebig verändern.

Mit dem folgenden Verfahren wird die Farbe der Skin „selected_over“ der Schaltfläche geändert.

So ändern Sie Farbe der Skin „selected_over“ einer Schaltfläche:

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 Parameter „toggle“ 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. Vergrößern Sie das Symbol auf 400 %, um es 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 in den 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“ angezeigt werden wie in der folgenden Abbildung.



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. Sie können zum Beispiel die Größe eines Kontrollkästchens ändern, indem Sie die Eigenschaften `height` und `width` sowie `scaleX` und `scaleY` der Komponente ändern. Wenn die Größe einer CheckBox-Komponente 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.

Stile für die CheckBox-Komponente

Durch Zuweisen von Stileigenschaften können Sie die Gestaltung einer CheckBox-Instanz ändern. Mit dem folgenden Verfahren ändern Sie zum Beispiel die Größe und die Farbe einer CheckBox-Bezeichnung.

So ändern Sie die Größe und die Farbe der Bezeichnung eines Kontrollkästchens:

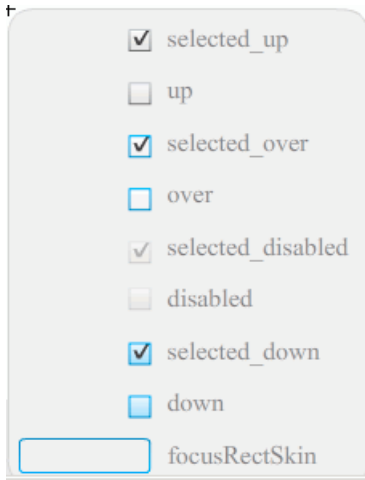
1. Ziehen Sie die CheckBox-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz **myCb**.
2. Klicken Sie im Eigenschafteninspektor auf die Registerkarte „Parameter“ und geben Sie den folgenden Wert für den Parameter „label“ ein: **Less than \$500?**
3. Klicken Sie auf 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 „[Stile](#)“ auf Seite 160. Informationen zum Einstellen von Stileigenschaften, um die Symbole und Skins der Komponente zu ändern, finden Sie unter „[Erstellen neuer Skins](#)“ auf Seite 168 und „[Skins für die CheckBox-Komponente](#)“ auf Seite 175.

Skins für die CheckBox-Komponente

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



Skins der CheckBox-Komponente

In diesem Beispiel wird die Konturfarbe und die Hintergrundfarbe der Komponenten in den Zuständen `up` und `selectedUp` geändert. Das Ändern der Skins für andere Zustände verläuft nach einem ähnlichen Muster.

So passen Sie CheckBox-Skins an:

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 die Skin „selected_up“, um sie im Symbolbearbeitungsmodus zu öffnen.
5. Vergrößern Sie das Symbol auf 800 %, um es besser bearbeiten zu können.
6. Klicken Sie auf den Rand des Kontrollkästchens, um diesen 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“.

ColorPicker-Komponente anpassen

Die einzige Größenänderung, die Sie an einer ColorPicker-Komponente vornehmen können, erfolgt über eine Bearbeitung der Stile: `swatchWidth`, `swatchHeight`, `backgroundPadding`, `textFieldWidth` und `textFieldHeight`. Wenn Sie versuchen, die Größe einer ColorPicker-Komponente mit dem Transformationswerkzeug oder mithilfe von ActionScript mit der Methode `setSize()` oder den Eigenschaften `width`, `height`, `scaleX` oder `scaleY` zu ändern, werden diese Werte beim Erstellen der SWF-Datei ignoriert, und die ColorPicker-Komponente wird mit der Standardgröße angezeigt. Der Hintergrund der Palette wird in seiner Größe an die Anzahl der Spalten angepasst, die mit `setStyle()` für den Stil `columnCount` festgelegt wurde. Die Anzahl der Spalten beträgt standardmäßig 18. Sie können eine anderen Anzahl (bis 1024) einstellen, und die Palette wird vertikal Richtung an die Anzahl der Farbfelder angepasst.

Stile für die ColorPicker-Komponente

Sie können verschiedene Stile festlegen, um das Erscheinungsbild der ColorPicker-Komponente zu ändern. Im folgenden Beispiel wird die Anzahl der Spalten (`columnCount`) im Farbauswahlfeld auf 12 gesetzt, die Höhe (`swatchHeight`) und die Breite (`swatchWidth`) der Farbfelder wird geändert und die Zellauffüllung für das Textfeld (`textPadding`) und den Hintergrund (`backgroundPadding`) wird ebenfalls geändert.

So ändern Sie das Erscheinungsbild der ColorPicker-Komponente mithilfe von Stilen:

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`.

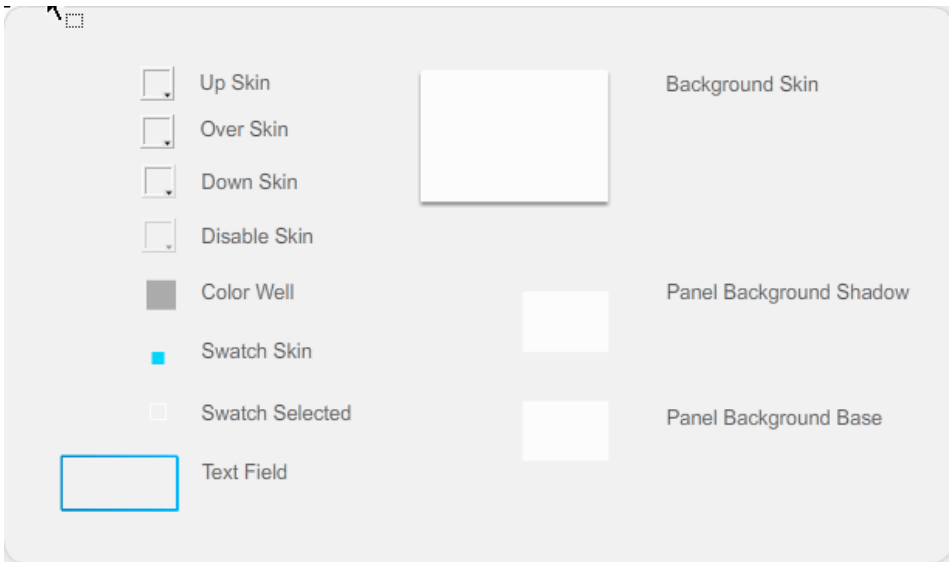
3. Öffnen Sie das Bedienfeld „Aktionen“, wählen Sie in der Hauptzeitleiste Bild 1 aus und geben Sie den folgenden Code ein:

```
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 Farbauswahl, um sie zu öffnen und zu sehen, wie sich die Änderungen auf das Erscheinungsbild auswirken.

Skins für die ColorPicker-Komponente

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



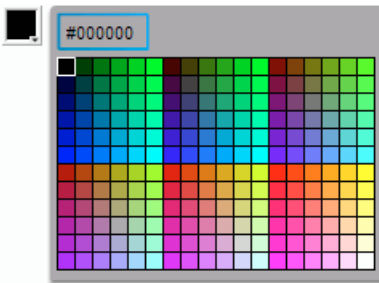
Skins der ColorPicker-Komponente

Sie können die Farbe der Hintergrundskin ändern, um die Farbe des Palettenhintergrunds zu ändern.

So ändern Sie die Farbe der Hintergrundskin:

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 den Hintergrund aus.
6. Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um in den 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.



ComboBox-Komponente anpassen

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 Liste wird an die Breite der Komponente angepasst, sofern nicht die Eigenschaft `dropdownWidth` festgelegt wurde.

Falls der Text nicht in das Kombinationsfeld passt, wird er unvollständig angezeigt. In diesem Fall müssen Sie die Größe der ComboBox ändern und die Eigenschaft `dropdownWidth` an die Textlänge anpassen.

Stile für die ComboBox-Komponente

Durch Zuweisen von Stileigenschaften können Sie die Gestaltung einer ComboBox-Komponente ändern. Die Stile geben Werte für die Skins, den CellRenderer, die Zellauffüllung 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 Stil „`textPadding`“ bestimmt den Abstand zwischen dem Rand eines Textfelds und dem darin enthaltenen Text. Dies ist hilfreich, um den Text vertikal im Textfeld zu zentrieren, wenn Sie die ComboBox-Komponente vergrößern. Andernfalls könnte der Text am oberen Rand des Textfelds angezeigt werden.

So legen Sie Stile für die ComboBox-Komponente fest:

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;

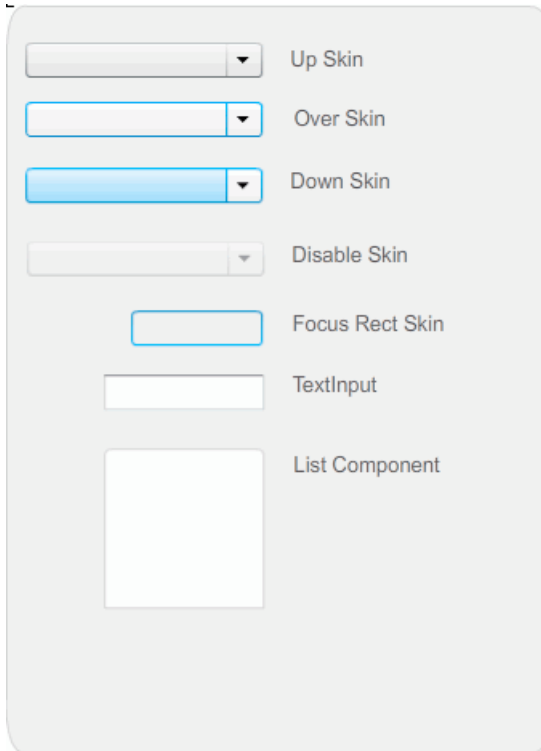
var items:Array = [
    {label:"San Francisco", data:"601 Townsend St."},
    {label:"San Jose", data:"345 Park Ave."},
    {label:"San Diego", data:"10590 West Ocean Air Drive, Suite 100"},
    {label:"Santa Rosa", data:"2235 Mercury Way, Suite 105"},
    {label:"San Luis Obispo", data:"3220 South Higuera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

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

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

Skins für die ComboBox-Komponente

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



Skins der ComboBox-Komponente

Sie können die Farbe der Up-Skin ändern, um die Farbe der Komponente in ihrem inaktiven Zustand auf der Bühne zu ändern.

So ändern Sie die Farbe der Hintergrundskin:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine 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 ist, und öffnen Sie sie zum Bearbeiten.
5. Stellen Sie Vergrößerung auf 400 % ein.

6. Klicken Sie auf den mittleren Bereich der Skin, bis seine 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 in den Dokumentbearbeitungsmodus zurückzukehren.
 9. Wählen Sie „Steuerung“ > „Film testen“.
- Die ComboBox-Komponente sollte wie in der folgenden Abbildung auf der Bühne angezeigt werden.



DataGrid-Komponente anpassen

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.

Stile für die 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 [„Stile mit der List-Komponente verwenden“ auf Seite 766.](#))

Festlegen von Stilen für einzelne Spalten

Eine DataGrid-Komponente kann mehrere Spalten haben, und Sie können für jede Spalte einen anderen CellRenderer angeben. Jede Spalte eines Datenrasters wird durch ein DataGridColumn-Objekt repräsentiert, und die DataGridColumn-Klasse enthält die Eigenschaft `cellRenderer`, für die Sie den CellRenderer für die Spalte definieren können.

So erstellen Sie eine mehrzeilige Spalte in einer DataGrid-Komponente:

1. Erstellen Sie eine neue Flash-Datei (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: Mit diesem Code wird ein Datenraster erstellt, das in der dritten Spalte einen langen Textstring enthält. 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 als `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 als `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“.

Das Datenraster 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

Festlegen von Überschriftenstilen

Sie können den Textstil für eine Überschriftenzeile mit dem Stil `headerTextFormat` festlegen. Im folgenden Beispiel wird das `TextFormat`-Objekt verwendet, um den Stil `headerTextFormat` auf die Schriftart Arial, die Farbe Rot, die Schriftgröße 14 und den Schriftstil Kursiv einzustellen.

So legen Sie den Stil „headerTextFormat“ für ein Datenraster fest:

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.
```



```

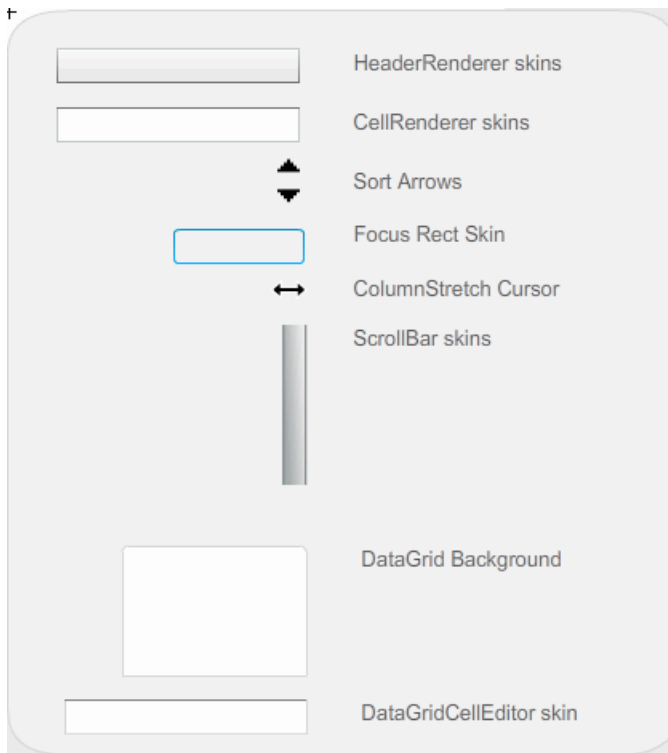
aDg.dataProvider = new DataProvider(myDP);
aDg.setSize(160,190);
aDg.move(40,40);
aDg.columns[0].width = 80;
aDg.columns[1].width = 80;
var tf:TextFormat = new TextFormat();
tf.size = 14;
tf.color = 0xff0000;
tf.italic = true;
tf.font = "Arial"
aDg.setStyle("headerTextFormat", tf);

```

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

Skins für die DataGrid-Komponente

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



Skins der DataGrid-Komponente

Die CellRenderer-Skin ist die Skin, die für die Hauptzellen des Datenrasters verwendet wird, während die HeaderRenderer-Skin für die Überschriftenzeile verwendet wird. In den folgenden Schritten wird die Hintergrundfarbe der Überschriftenzeile geändert; Sie könnten auf die gleiche Weise vorgehen, um die Hintergrundfarbe der Hauptzellen des Datenraster zu ändern, indem Sie die CellRenderer-Skin bearbeiten.

So ändern Sie die Hintergrundfarbe der Überschriftenzeile des Datenrasters:

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. 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 in den 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.

Das Datenraster sollte wie in der folgenden Abbildung aussehen, mit dem Hintergrund der Überschriftenzeile in Grün.

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

Label-Komponente anpassen

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 wird je nach Einstellung des Parameters `wordwrap` geändert. Falls der Parameter den Wert `true` hat, wird die Bezeichnung vertikal an den Text angepasst. Wenn der Parameter den Wert `false` hat, wird die Bezeichnung horizontal angepasst. Zur Laufzeit verwenden Sie die Methode `setSize()`. Weitere Informationen finden Sie in den Abschnitten zur Methode `Label.setSize()` und zur Eigenschaft `Label.autoSize` im *Komponenten-Referenzhandbuch für ActionScript 3.0*. Siehe auch „Erstellen von Anwendungen mit der Label-Komponente“ auf Seite 109.

Stile für die 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 Stil `textFormat`, der dieselben Attribute enthält wie das `TextFormat`-Objekt. Er ermöglicht Ihnen, die selben Eigenschaften für den Inhalt von `Label.text` festzulegen wie bei einem normalen Textfeld in Flash. Im folgenden Beispiel wird die Textfarbe der Bezeichnung auf Rot festgelegt.

So ändern Sie die Textfarbe einer Bezeichnung:

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 Bild 1 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 über Stile der Label-Komponente finden Sie im Abschnitt über die Label-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Skins für die Label-Komponente

Die Label-Komponente enthält keine grafischen Elemente, die mit Skins versehen werden können.

List-Komponente anpassen

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 bei Bedarf automatisch positioniert.

Stile für die List-Komponente

Sie können Stileigenschaften festlegen, um das Erscheinungsbild der List-Komponente zu ändern. Die Stile legen die Werte für die Skins und Zellauffüllungen beim Zeichnen der Komponente fest.

Die verschiedenen Skinstile ermöglichen es, verschiedene Klassen festzulegen, die für die Skin verwendet werden sollen. Weitere Informationen zum Verwenden der Skinstile finden Sie unter „Skins“ auf Seite 165.

In den folgenden Schritten wird der Wert des Stils `contentPadding` für die List-Komponente festgelegt. Beachten Sie, dass der Wert dieser Einstellung von der Größe der Liste subtrahiert wird, um die Zellauffüllung um den Inhalt zu bestimmen, deshalb müssen Sie möglicherweise die Größe der Liste ändern, damit der Text in der Liste nicht abgeschnitten wird.

So legen Sie den Stil „contentPadding“ für die Liste fest:

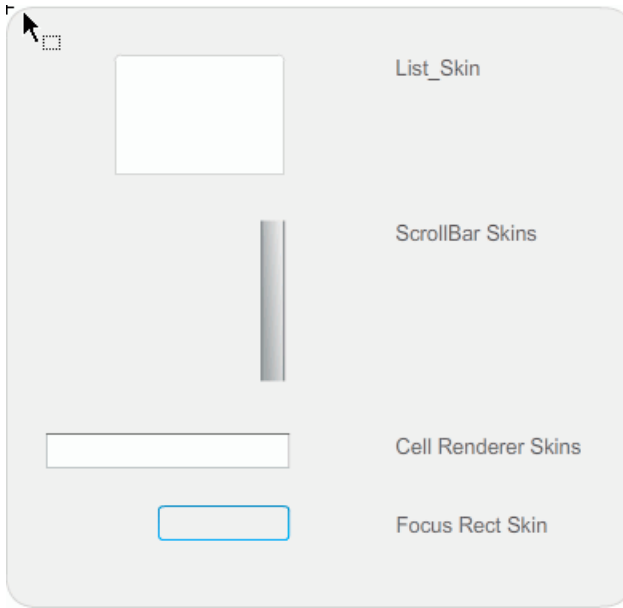
1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz `aList`.
3. Wählen Sie in der Hauptzeitleiste Bild 1 aus, öffnen Sie das Bedienfeld „Aktionen“ und geben Sie den folgenden Code ein, mit 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 (Xc11nt Cond)", data:17000});
aList.rowCount = aList.length;
```

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

Skins für die List-Komponente

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



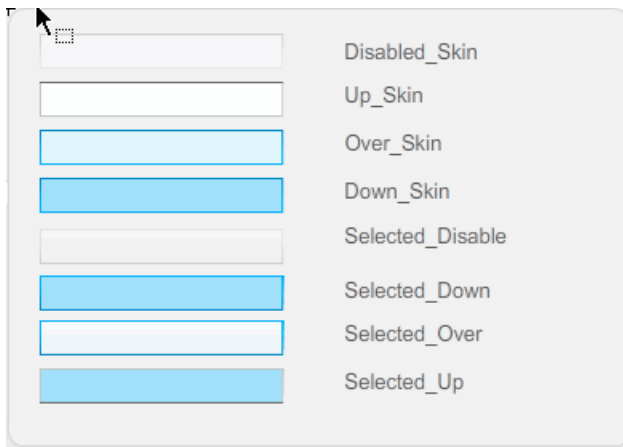
Skins der List-Komponente

Weitere Informationen zum Skinning von Bildlaufleisten finden Sie unter [„UIScrollBar-Komponente anpassen“ auf Seite 213](#). Informationen zum Skinning der „Focus Rect“-Skin finden Sie unter [„TextArea-Komponente anpassen“ auf Seite 204](#).

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 CellRenderer-Skin, um eine zweite Palette mit den Skins für die verschiedenen Zustände einer Listenzelle zu öffnen.



Die CellRenderer-Skins des List-Komponente

Sie können das Erscheinungsbild 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.

So ändern Sie Farbe der CellRenderer-Skin „Up_Skin“ einer Liste:

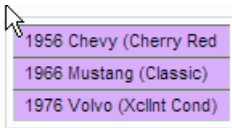
1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die List-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne und nennen Sie die Instanz **aList**.
3. Doppelklicken Sie auf die Liste, um die Palette ihrer Skins zu öffnen.
4. Doppelklicken Sie auf die CellRenderer-Skin, um 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 Füllungsbereich der Skin, um ihn auszuwählen. Im Eigenschafteninspektor wird ein Farbfeld „Füllung“ angezeigt, das die aktuelle Füllfarbe der Skin enthält.
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 in den 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 (Xc11nt Cond)", data:17000});  
aList.rowCount = aList.length;
```

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

Die Liste sollte jetzt wie in der folgenden Abbildung angezeigt werden:



Die Umrandung wird durch die Einstellung des Stils `contentPadding` bewirkt.

NumericStepper-Komponente anpassen

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 für die NumericStepper-Komponente

Sie können die Stileigenschaften der NumericStepper-Komponenten festlegen, um das Erscheinungsbild der Nummernauswahl zu ändern. Die Stile legen die Werte für die Skins, Zellauffüllungen und Textformate beim Zeichnen der Komponente fest. Der Stil `textFormat` ermöglicht Ihnen, die Größe und das Aussehen der Werts der Nummernauswahl zu ändern. Die verschiedenen Skinstile ermöglichen es, verschiedene Klassen festzulegen, die für die Skin verwendet werden sollen. Weitere Informationen zum Verwenden der Skinstile finden Sie unter „Skins“ auf Seite 165.

So ändern Sie das Erscheinungsbild des Wertes in der Nummernauswahl:

In den folgenden Schritten wird der Stil `textFormat` verwendet, um das Aussehen des in der NumericStepper-Komponente angezeigten Werts zu ändern.

1. Erstellen Sie eine neue Flash-Datei (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“.

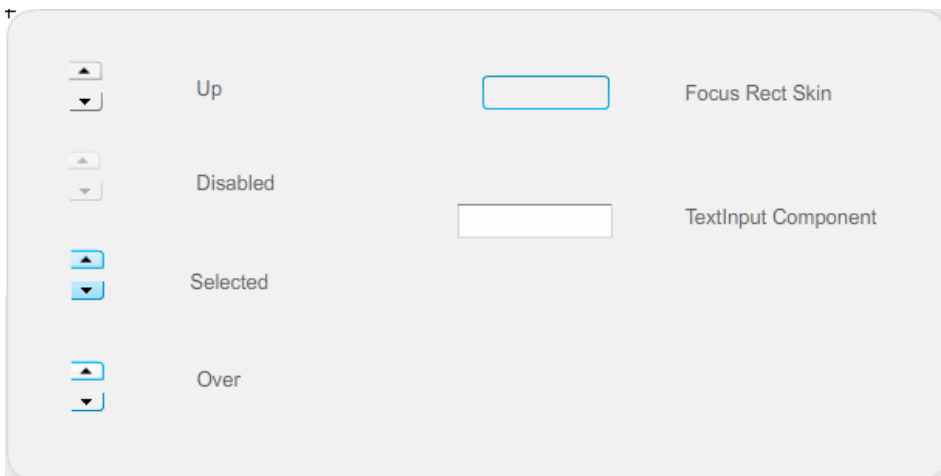
Skins für die NumericStepper-Komponente

Die NumericStepper-Komponente verfügt über Skins, um die verschiedenen Zustände ihrer Schaltflächen (Up, Down, Disabled, Selected) darzustellen.

Wenn ein Stepper aktiviert wird, werden die Schaltflächen nach oben und unten im Zustand „Over“ angezeigt, wenn der Zeiger über sie hinweg bewegt wird. Wenn auf die Schaltflächen geklickt wird, werden sie im Zustand „Down“ angezeigt. Wenn die Maustaste losgelassen wird, kehren die Schaltflächen zum Zustand „Over“ 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 Zustand „Disabled“ angezeigt.

Die NumericStepper-Komponente verfügt über die folgenden Skins:



Skins der NumericStepper-Komponente

So ändern Sie Farbe des Texthintergrunds und der Schaltflächen im Zustand „Up“ für die NumericStepper-Komponente:

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 in den Dokumentbearbeitungsmodus zurückzukehren.
7. Doppelklicken Sie erneut auf die NumericStepper-Komponente, um die Palette 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:



ProgressBar-Komponente anpassen

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`.

Die ProgressBar-Komponente verfügt über drei Skins: eine für die Leiste, eine für den Balken sowie eine unbestimmte Skin. Für die Skalierung der Elemente wird Scale-9 verwendet.

Stile für die ProgressBar-Komponente

Durch Zuweisen von Stileigenschaften können Sie die Gestaltung einer ProgressBar-Instanz ändern. Die Stile der ProgressBar-Komponente legen die Werte für die Skins und Zellauffüllungen beim Zeichnen der Komponente fest. Im folgenden Beispiel wird eine ProgressBar-Instanz vergrößert, und der Stil `barPadding` wird festgelegt.

So legen Sie die Stile für die Zellauffüllung einer ProgressBar-Instanz fest:

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:

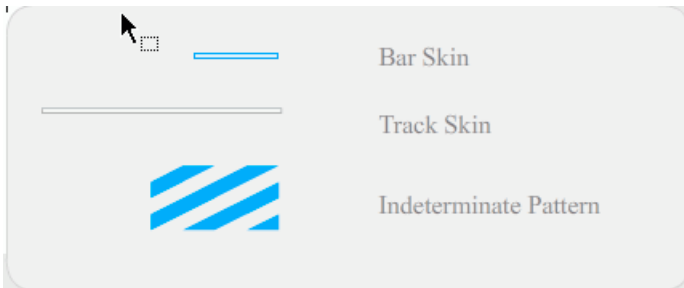
```
myPb.width = 300;  
myPb.height = 30;  
  
myPb.setStyle("barPadding", 3);
```

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

Einzelheiten zum Festlegen von Skinstilen finden Sie unter „Skins“ auf Seite 165.

Skins für die ProgressBar-Komponente

Die ProgressBar-Komponente stellt den Balken der Fortschrittleiste, die vollständige Leiste und die unbestimmte Leiste mithilfe von Skins dar, wie in der folgenden Abbildung dargestellt.



Skins der ProgressBar-Komponente

Die Leiste wird über der Skin für den Balken platziert, dabei wird die Position mithilfe der Einstellung für die Zellauffüllung festgelegt. 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 Skin wird vertikal und horizontal in der Größe an die ProgressBar-Instanz angepasst.

Sie können diese Skins bearbeiten, um das Erscheinungsbild der Fortschrittsleiste zu ändern. Im folgenden Beispiel wird die Farbe der unbestimmten Leiste geändert.

So ändern Sie die Farbe der unbestimmten Leiste durch Bearbeiten ihrer Skin:

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. Vergrößern Sie das Symbol auf 400 %, um es 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 auf die Farbauswahl „Füllung“ im Eigenschafteninspektor, 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 in den Dokumentbearbeitungsmodus zurückzukehren.
8. Wählen Sie „Steuerung“ > „Film testen“.

Die Fortschrittleiste sollte jetzt wie in der folgenden Abbildung aussehen.



RadioButton-Komponente anpassen

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 Begrenzungsrahmen der RadioButton-Komponente ist unsichtbar und definiert außerdem den Kollisionsbereich der Komponente. Wenn Sie die Komponente vergrößern, vergrößert sich auch die Größe des Kollisionsbereichs.

Wenn die Begrenzungsbox der Komponente zu klein ist, um die Komponentenbezeichnung aufzunehmen, wird die Bezeichnung abgeschnitten.

Stile für die RadioButton-Komponente

Sie können Stileigenschaften festlegen, um das Erscheinungsbild eines Optionsfelds zu ändern. Die Stile der ScrollPane-Komponente legen die Werte für die Skins, Symbole, Textformatierung und Zellauffüllungen beim Zeichnen der Komponente fest. Die Stile der ScrollPane-Komponente legen die Werte für die Skins und Zellauffüllungen beim Zeichnen der Komponente fest.

Im folgenden Beispiel wird der Stil `textFormat` von einer CheckBox-Komponente abgerufen und auf eine RadioButton-Komponente angewendet, damit die Bezeichnungen der beiden Elemente ein einheitliches Aussehen haben.

So wenden Sie den Stil „textFormat“ einer CheckBox-Komponente auf eine RadioButton-Komponente an:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie eine CheckBox-Komponente auf die Bühne und geben Sie dieser im Eigenschafteninspektor den Instanznamen **myCh**.
3. Ziehen Sie eine RadioButton-Komponente auf die Bühne und geben Sie dieser im Eigenschafteninspektor den Instanznamen **myRb**.
4. Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Zeitleiste den folgenden Code hinzu:

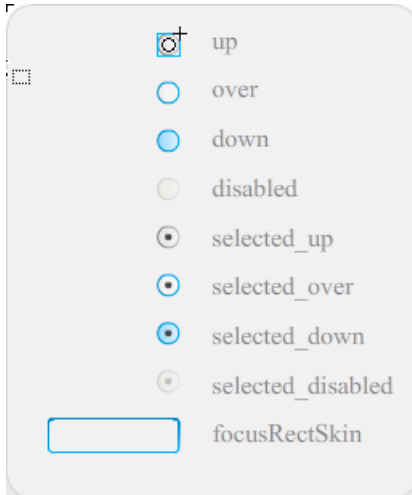
```
var tf:TextFormat = new TextFormat();
tf.color = 0x00FF00;
tf.font = "Georgia";
tf.size = 18;
myCh.setStyle("textFormat", tf);
myRb.setStyle("textFormat", myCh.getStyle("textFormat"));
```

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 für die RadioButton-Komponente

Die RadioButton-Komponente verfügt über die folgenden Skins, die Sie bearbeiten können, um das Erscheinungsbild der Komponente zu ändern:



Skins der RadioButton-Komponente

Wenn ein Optionsfeld aktiviert, aber nicht ausgewählt ist, zeigt es den Zustand „Over“ an, wenn ein Benutzer den Mauszeiger darauf setzt. 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 Skin „Up“ angezeigt.

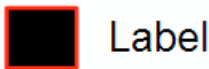
Deaktivierte Optionsfelder werden unabhängig von den Bedienvorgängen des Benutzers immer im Disabled-Status angezeigt.

Im folgenden Beispiel wird die Skin „selected_up“, die den ausgewählten Zustand darstellt, ersetzt.

So erstellen Sie eine neue Skin `selected_up` für eine `RadioButton`-Komponente:

1. Erstellen Sie eine neue Flash-Datei (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. Vergrößern Sie das Symbol auf 800 %, um es besser bearbeiten zu können.
4. Doppelklicken Sie auf die Skin „`selected_up`“, um sie auszuwählen, und drücken Sie die Taste <Entf>, 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 in den Dokumentbearbeitungsmodus zurückzukehren.
9. Wählen Sie „Steuerung“ > „Film testen“.
10. Klicken Sie auf das Optionsfeld, um es auszuwählen.

Das Optionsfeld im ausgewählten Zustand sollte jetzt ähnlich wie in der folgenden Abbildung aussehen.



ScrollPane-Komponente anpassen

`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 verfügt über 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 der ScrollPane-Komponente geändert wird, werden die Spur der Bildlaufleiste und das Bildlauffeld vergrößert oder verkleinert und die Größe der Kollisionsbereiche wird entsprechend angepasst. Die Schaltflächen behalten ihre Größe.

Stile für die ScrollPane-Komponente

Die Stileigenschaften der ScrollPane-Komponente legen die Werte für die Skins und Zellauffüllungen beim Zeichnen der Komponente fest. Die verschiedenen Skinstile ermöglichen es, verschiedene Klassen festzulegen, die für die Skins der Komponente verwendet werden sollen. Weitere Informationen zum Verwenden der Skinstile finden Sie unter „Skins“ auf Seite 165.

So legen Sie den Stil „contentPadding“ für eine ScrollPane-Komponente fest:

1. Erstellen Sie eine neue Flash-Datei.
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 Parameter „source“ 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);
```

Beachten Sie, dass die Zellauffüllung zwischen dem Rand der Komponente und ihrem Inhalt, außerhalb der Bildlaufleisten, angewendet wird.

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

Skins für die ScrollPane-Komponente

Die ScrollPane-Komponente verwendet einen Rand und Bildlaufleisten, um durch Bestände zu scrollen. Informationen zum Anwenden von Skins auf Bildlaufleisten finden Sie unter [„Skins für die UIScrollBar-Komponente“ auf Seite 214](#).

Slider-Komponente anpassen

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`.

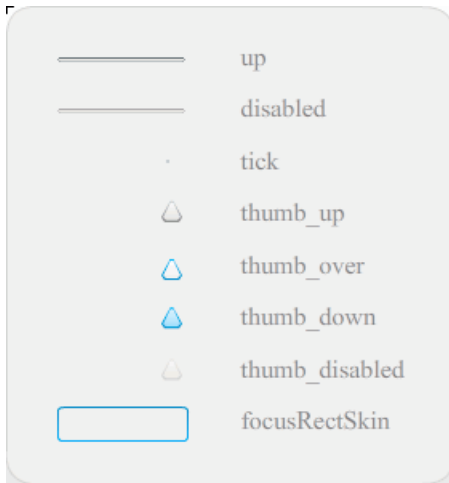
Sie können einen Schieberegler nur verlängern, jedoch nicht die Höhe vergrößern. Flash ignoriert die Eigenschaft `height` und den Parameter „height“ der Methode `setSize()`. Sie können jedoch einen vertikalen Schieberegler erstellen und in vertikaler Richtung verlängern.

Stile für die Slider-Komponente

Die Stile der Slider-Komponente legen nur die Klassen für die Skin sowie einen Wert für `FocusRectPadding` fest, der angibt, wie viele Pixel für die Zellauffüllung zwischen dem Begrenzungsrahmen der Komponente und deren äußeren Rand liegen. Weitere Informationen zum Verwenden der Skinstile finden Sie unter [„Skins“ auf Seite 165](#).

Skins für die Slider-Komponente

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



Skins der Slider-Komponente

Im folgenden Beispiel wird die Farbe der Up-Skin des Balkens zu Blau geändert.

So ändern Sie Farbe der Skin „Up“ eines Schiebereglers:

1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die Slider-Komponente aus dem Bedienfeld „Komponenten“ auf die Bühne.
3. Doppelklicken Sie auf die Slider-Komponente, ein Fenster mit ihren Skins zu öffnen.
4. Doppelklicken Sie an ihrem Registrierungspunkt auf die Up-Skin, um sie im Symbolbearbeitungsmodus zu öffnen.
5. Vergrößern Sie das Symbol auf 800 %, um es besser bearbeiten zu können. Beachten Sie, dass die Spur des Schiebereglers aus drei Balken besteht.
6. Klicken Sie auf den oberen Balken, um ihn auszuwählen. Wenn er ausgewählt ist, erscheint die entsprechende Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor.
7. Wählen Sie in der Farbauswahl „Füllung“ im Eigenschafteninspektor die Farbe „#000066“ aus, um sie für den oberen Balken der Schiebereglerspur zu verwenden.
8. Klicken Sie auf den mittleren Balken der Schiebereglerspur, um ihn auszuwählen. Wenn er ausgewählt ist, erscheint die entsprechende Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor.

9. Wählen Sie in der Farbauswahl „Füllung“ im Eigenschafteninspektor die Farbe „#0066FF“ aus, um sie für den mittleren Balken der Schiebereglerspur zu verwenden.
10. Klicken Sie auf den unteren Balken der Schiebereglerspur, um ihn auszuwählen. Wenn er ausgewählt ist, erscheint die entsprechende Farbe in der Farbauswahl „Füllung“ im Eigenschafteninspektor.
11. Wählen Sie in der Farbauswahl „Füllung“ im Eigenschafteninspektor die Farbe „#00CCFF“ aus, um sie für den unteren Balken der Schiebereglerspur zu verwenden.
12. Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um in den Dokumentbearbeitungsmodus zurückzukehren.
13. Wählen Sie „Steuerung“ > „Film testen“.

Der Schieberegler sollte jetzt wie in der folgenden Abbildung angezeigt werden



TextArea-Komponente anpassen

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 Feldrahmens entsprechend des neuen Begrenzungsrahmens geändert. 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. Falls die TextArea-Komponente zu schmal für den Text ist, wird dieser nicht vollständig angezeigt.

Stile für die TextArea-Komponente

Die Stile der TextArea-Komponente legen die Werte für die Skins, Zellauffüllungen und Textformate beim Zeichnen der Komponente fest. Die Stile `textFormat` und `disabledTextFormat` steuern den Stil des Textes, der im Textfeld angezeigt wird. Weitere Informationen zur Verwendung der Skins finden Sie unter „Skins für die TextArea-Komponente“ auf Seite 206.

Im folgenden Beispiel wird der Stil `disabledTextFormat` festgelegt, um das Erscheinungsbild des Textes zu ändern, wenn das Textfeld deaktiviert wird, dasselbe Verfahren gilt jedoch auch für die Verwendung des Stils `textFormat` für ein aktiviertes Textfeld.

So legen Sie den Stil „disabledTextFormat“ für eine TextArea-Komponente fest:

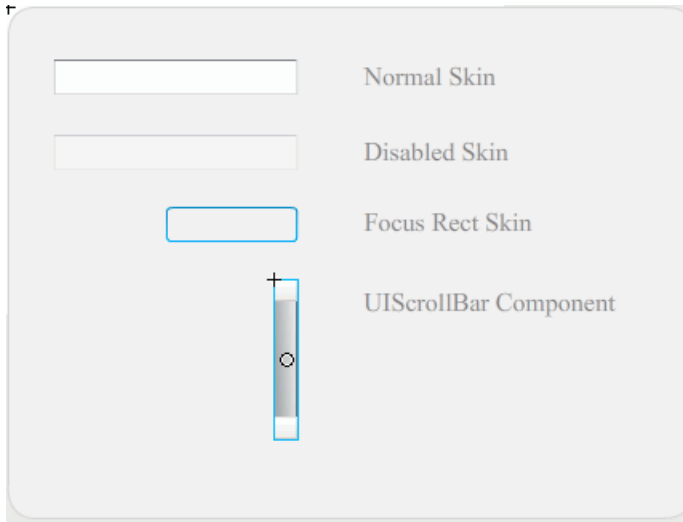
1. Erstellen Sie eine neue Flash-Datei.
2. Ziehen Sie eine TextArea-Komponente auf die Bühne und geben Sie ihr den Instanznamen `myTa`.
3. Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Hauptzeitleiste den folgenden Code hinzu:

```
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“.

Skins für die TextArea-Komponente

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



Skins der TextArea-Komponente

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 wird die Randfarbe der „Focus Rect Skin“, die angezeigt wird, wenn das Textfeld den Fokus erhält, und der „Normal Skin“ geändert.

So ändern Sie die Farbe der Textfeldränder:

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 Rand von „Focus Rect Skin“, um diesen 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 öffnen, und wählen Sie die Farbe „#CC0000“ aus, um sie für den Rand zu verwenden.
6. Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um in den 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 in den Dokumentbearbeitungsmodus zurückzukehren.
11. Wählen Sie „Steuerung“ > „Film testen“.

Wenn Sie das Textfeld auswählen, um mit der Texteingabe zu beginnen, sollte der Rand wie in der folgenden Abbildung aussehen:



Der äußere Rand ist die „Focus Rect Skin“, und der innere Rand ist der Rahmen der „Normal“-Skin.

Informationen zum Bearbeiten der UIScrollBar-Skin finden Sie unter [„UIScrollBar-Komponente anpassen“](#) auf Seite 213.

TextInput-Komponente anpassen

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 Feldrahmens entsprechend der neuen Begrenzungsbox geändert. 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 zum Anzeigen des Textes zu klein ist, wird der Text abgeschnitten.

Stile für die TextInput-Komponente

Die Stile der TextInput-Komponente legen die Werte für die Skins, Zellauffüllungen und Textformatierung beim Zeichnen der Komponente fest. Die Stile `textFormat` und `disabledTextFormat` steuern den Stil des Textes, der in der Komponente angezeigt wird. Weitere Informationen zur Verwendung der Skins finden Sie unter „Skins für die TextInput-Komponente“ auf Seite 209.

Im folgenden Beispiel wird der Stil `textFormat` verwendet, um die Schriftart, -größe und -farbe des Textes in der TextInput-Komponente festzulegen. Dasselbe Verfahren gilt auch für das Festlegen des Stils `disabledTextFormat`, der auf die deaktivierte Komponente angewendet wird.

So legen Sie den Stil „textFormat“ für eine TextInput-Instanz fest:

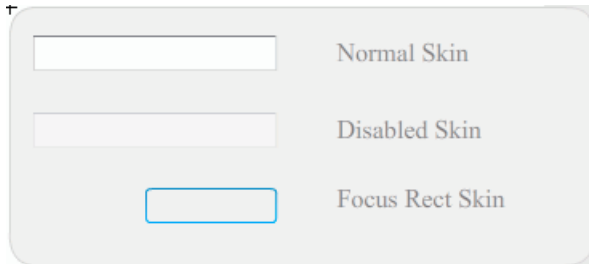
1. Erstellen Sie eine neue Flash-Datei.
2. Ziehen Sie eine TextInput-Komponente auf die Bühne und geben Sie ihr den Instanznamen `myTi`.
3. Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Hauptzeitleiste den folgenden Code hinzu:

```
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 für die TextInput-Komponente

Die TextInput-Komponente verfügt über die folgenden Skins, die Sie bearbeiten können, um das Erscheinungsbild der Komponente zu ändern:



Skins der TextInput-Komponente

Im folgenden Verfahren werden die Rahmen- und die Hintergrundfarbe einer TextInput-Komponente geändert:

So ändern Sie die Farbe für den Rahmen und den Hintergrund der TextInput-Komponente:

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. Vergrößern Sie das Symbol auf 800 %, um es 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 in den Dokumentbearbeitungsmodus zurückzukehren.
8. Wählen Sie „Steuerung“ > „Film testen“.

Das Texteingabefeld sollte jetzt wie in der folgenden Abbildung aussehen:



TileList-Komponente anpassen

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 wie zum Beispiel `width`, `height`, `columnCount`, `rowCount`, `scaleX` und `scaleY`. Die in der TileList-Komponente enthaltene Bildlaufleiste wird mit dem Listenfeld skaliert.

Stile für die TileList-Komponente

Die Stile der TileList-Komponente legen die Werte für die Skins, Zellauffüllungen und Textformatierung beim Zeichnen der Komponente fest. Die Stile `textFormat` und `disabledTextFormat` steuern den Stil des Textes, der in der Komponente angezeigt wird. Weitere Informationen zu den Skinstilen finden Sie unter „[Skins für die TileList-Komponente](#)“ auf Seite 211.

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. Dasselbe Verfahren gilt auch für das Festlegen des Stils `disabledTextFormat`, der angewendet wird, wenn die Eigenschaft `enabled` auf `false` gesetzt wurde.

So legen Sie den Stil „textFormat“ für eine TileList-Instanz fest:

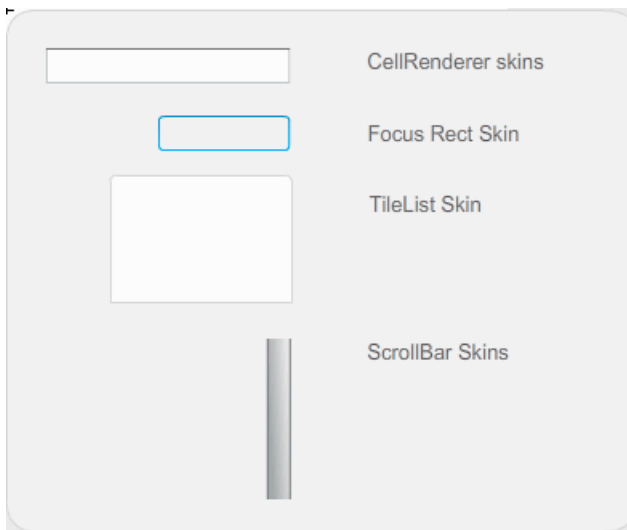
1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die TileList-Komponente auf die Bühne und geben Sie ihr den Instanznamen `myTl`.
3. Fügen Sie dem Bedienfeld „Aktionen“ von Bild 1 in der Zeitleiste den folgenden Code hinzu:

```
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);
```

Skins für die TileList-Komponente

Die TileList-Komponente verfügt über eine TileList-, eine CellRenderer- und eine ScrollBar-Skin. Sie können diese Skins bearbeiten, um das Erscheinungsbild der TileList-Instanz zu ändern.



Skins der TileList-Komponente

HINWEIS

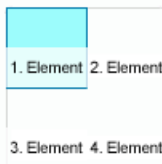
Wenn Sie die ScrollBar-Skin in einer Komponente ändern, wird sie auch für alle anderen Komponenten, die die ScrollBar-Komponente verwenden, geändert.

Mit dem folgenden Verfahren wird die Farbe der CellRenderer-Skin „Selected_Up“ der TileList-Komponente geändert.

So ändern Sie Farbe der CellRenderer-Skin einer TileList-Komponente:

1. Erstellen Sie eine Flash-Datei (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 Selected_Up-Skin aus.
5. Klicken Sie links neben der Bearbeitungsleiste über der Bühne auf „Zurück“, um in den 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 der Zellen in der TileList-Komponente, um sie auszuwählen, und bewegen Sie den Mauszeiger dann von der ausgewählten Zelle weg.

Die ausgewählte Zelle sollte jetzt wie in der folgenden Abbildung angezeigt werden:



UILoader-Komponente anpassen

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 durch die 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, und `setSize()` sowie die Größeneigenschaften bleiben wirkungslos.

Die UILoader-Komponente verfügt über keine UI-Elemente, auf die Sie Stile oder Skins anwenden können.

UIScrollBar-Komponente anpassen

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

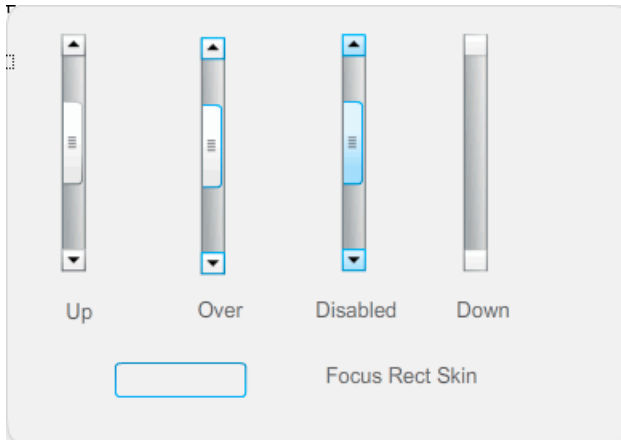
Mit der Methode `setSize()` können Sie nur die Breite einer horizontalen Bildlaufleiste oder die Höhe einer vertikalen Bildlaufleiste ändern. Beim Authoring können Sie die Höhe einer horizontalen Bildlaufleiste oder die Breite einer vertikalen Bildlaufleiste ändern, die Werte werden beim Veröffentlichen des Movies jedoch zurückgesetzt. Nur die Länge einer Bildlaufleiste kann geändert werden.

Stile für die UIScrollBar-Komponente

Die Stile der UIScrollBar-Komponente legen nur die Klassen für die Skin sowie einen Wert für „FocusRectPadding“ fest, der angibt, wie viele Pixel für die Zellauffüllung zwischen dem Begrenzungsrahmen der Komponente und deren äußeren Rand liegen. Weitere Informationen zum Verwenden der Skinstile finden Sie unter „Skins“ auf Seite 165.

Skins für die UIScrollBar-Komponente

Die UIScrollBar-Komponente verwendet die folgenden Skins.



Skins der UIScrollBar-Komponente

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 der UIScrollBar-Komponente und der Pfeilschaltflächen geändert.

So ändern Sie die Farbe der UIScrollBar-Skins:

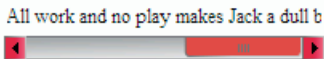
1. Erstellen Sie eine neue Flash-Datei (ActionScript 3.0).
2. Ziehen Sie die UIScrollBar-Komponente auf die Bühne und geben Sie ihr den Instanznamen **mySb**. Setzen Sie „direction“ auf der Registerkarte „Parameter“ auf den Wert „horizontal“.
3. Doppelklicken Sie auf die Bildlaufleiste, um die Palette ihrer Skins zu öffnen.
4. Klicken Sie auf die Skin „Up“, um sie auszuwählen.
5. Vergrößern Sie das Symbol auf 400 %, um es 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 in den Dokumentbearbeitungsmodus zurückzukehren.
9. Wiederholen Sie die Schritte 6 bis 8 für das Bildlauffeld und den Pfeil nach links (bzw. nach unten, falls es sich um eine vertikale Bildlaufleiste handelt).
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.



Die FLVPlayback-Komponente

Mit der FLVPlayback-Komponente können Sie unkompliziert einen Video-Player in Ihre Adobe Flash CS3 Professional-Anwendung einschließen, um progressiv heruntergeladene Adobe Flash Video-Dateien (FLV) über HTTP oder Streaming-FLV-Dateien von einem Macromedia Flash Media Server von Adobe oder von einem Flash Video Streaming Service (FVSS) abzuspielen.

Die benutzerfreundliche FLVPlayback-Komponente verfügt über die folgenden Merkmale und Vorteile:

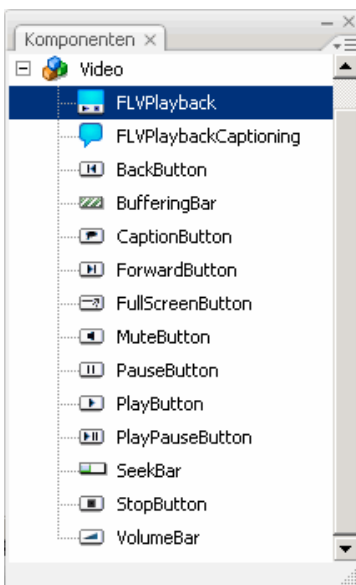
- Kann auf die Bühne gezogen und schnell und problemlos implementiert werden.
- Unterstützt den Vollbildmodus
- Bietet verschiedene vordefinierte *Skins* zum individuellen Gestalten der Abspielsteuerungen.
- Ermöglicht die Auswahl von Farb- und Alpha-Werten für vordefinierte Skins.
- Ermöglicht erfahrenen Benutzern das Erstellen eigener Skins.
- Bietet die Möglichkeit zur Live-Vorschau beim Authoring
- Bietet Layout-Eigenschaften, um die FLV-Datei bei Größenänderungen zu zentrieren.
- Ermöglicht den Abspielbeginn, wenn ein ausreichender Anteil der progressiv heruntergeladenen FLV-Datei 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

Die Verwendung der FLVPlayback-Komponente besteht im Wesentlichen darin, sie auf die Bühne zu ziehen und eine FLV-Datei zum Abspielen auszuwählen. Darüber hinaus können Sie auch noch verschiedene Parameter einstellen, die das Verhalten der Komponente regeln und die FLV-Datei beschreiben.

Die FLVPlayback-Komponente enthält auch eine ActionScript-API (API = application programming interface). Die API enthält die folgenden Klassen, deren vollständige Beschreibung Sie im *Komponenten-Referenzhandbuch für ActionScript 3.0* finden: [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 FLV-Datei 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 usw. 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 zu sehen ist:



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 auf die folgenden Arten ändern:

- Wählen Sie eine von mehreren vorgefertigten Skins aus
- Erstellen Sie eine benutzerdefinierte Skin und fügen Sie sie der Gruppe der vorgefertigten 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 „[Auswahl vordefinierter Skins](#)“ auf Seite 243.

Nachdem Sie eine andere Skin ausgewählt haben, wird die ausgewählte Skin zur neuen Standardskin.

Weitere Informationen zum Auswählen oder Erstellen von Skins für die [FLVPlayback-Komponente](#) finden Sie unter „[Anpassen der FLVPlayback-Komponente](#)“ auf Seite 242.

Erstellen von Anwendungen 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 eine Skin zuweisen, indem Sie 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.

So ziehen Sie die FLVPlayback-Komponente aus dem Bedienfeld „Komponenten“:

1. Klicken Sie im Bedienfeld „Komponenten“ auf das Pluszeichen (+), um das Video 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:

- einen lokalen Pfad zu einer FLV-Datei
- eine URL zu einer FLV-Datei
- eine URL zu einer SMIL-Datei, in der das Abspielen einer FLV-Datei beschrieben wird (SMIL = Synchronized Multimedia Integration Language)

Wie Sie eine SMIL-Datei erstellen, um eine oder mehrere FLV-Dateien zu beschreiben, wird unter [„Verwenden von SMIL-Dateien“ auf Seite 260](#) erläutert.

4. Während auf der Bühne die FLVPlayback-Komponente 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.

Wenn Sie die Farbe eines benutzerdefinierten UI-Steuerelements ändern möchten, müssen Sie es anpassen. Informationen zur Verwendung benutzerdefinierter UI-Steuerelemente finden Sie unter [„Zuweisen von Skins für einzelne FLV Playback Custom UI-Komponenten“ auf Seite 245](#).

7. Klicken Sie auf „OK“, um das Dialogfeld „Skin auswählen“ zu schließen.
8. Wählen Sie „Steuerung“ > „Film testen“, um die SWF-Datei auszuführen und das Video zu starten.

Im folgenden Verfahren wird mit dem Videoimportassistenten eine FLVPlayback-Komponente hinzugefügt:

So verwenden Sie den 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
 - Vom Flash Media Server streamen
 - Video in eine SWF-Datei einbetten und in der Zeitleiste abspielen

WARNUNG

Wählen Sie nicht die Option zum Einbetten des Videos. Die FLVPlayback-Komponente spielt nur externes Streaming-Video ab. Mit dieser Option wird eine FLVPlayback-Komponente nicht 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.

So erstellen Sie eine Instanz mithilfe von ActionScript dynamisch:

1. Ziehen Sie die FLVPlayback-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ („Fenster“ > „Bibliothek“).
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 Ordners „Skins“ in Ihrer Installation anzugeben:

Für Windows:

```
import fl.video.*;
var my_FLVPlaybk = new FLVPlayback();
my_FLVPlaybk.x = 100;
my_FLVPlaybk.y = 100;
addChild(my_FLVPlaybk);
my_FLVPlaybk.skin = "file:///install_drive|/Program Files/Adobe/Adobe
  Flash CS3/en/Configuration/FLVPlayback Skins/ActionScript 3.0/
  SkinOverPlaySeekMute.swf"
my_FLVPlaybk.source = "http://www.helpexamples.com/flash/video/
  water.flv";
```

Für Macintosh:

```
import fl.video.*;
var my_FLVPlaybk = new FLVPlayback();
my_FLVPlaybk.x = 100;
my_FLVPlaybk.y = 100;
addChild(my_FLVPlaybk);
```

```
my_FLVPlaybk.skin = "file:///Macintosh HD:Applications:Adobe Flash
  CS3:Configuration:FLVPlayback Skins:ActionScript
  3.0SkinOverPlaySeekMute.swf"
my_FLVPlaybk.source = "http://www.helpexamples.com/flash/video/
  water.flv";
```

HINWEIS

Wenn Sie die Eigenschaften `source` und `skin` nicht festlegen, ist der generierte Movieclip leer.

3. Wählen Sie „Steuerung“ > „Film testen“, um die SWF-Datei auszuführen und die FLV-Datei zu starten.

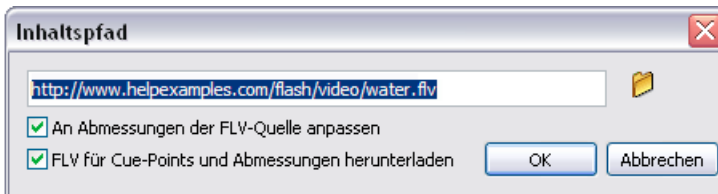
Parameter der FLVPlayback-Komponente

Die folgenden Authoring-Parameter können für jede Instanz der FLVPlayback-Komponente im Komponenten-Inspektor oder im Eigenschafteninspektor festgelegt werden: `align`, `autoPlay`, `cuePoints`, `preview`, `scaleMode`, `skin`, `skinAutoHide`, `skinBackgroundAlpha`, `skinBackgroundColor`, `source` und `volume`. Jeder dieser Parameter verfügt über eine entsprechende ActionScript-Eigenschaft desselben Namens. Wenn Sie diesen Parametern einen Wert zu weisen, legen Sie den anfänglichen Zustand der Eigenschaft in der Anwendung fest. Durch das Einstellen der Eigenschaft in ActionScript wird der Wert, den Sie für den Parameter festlegen, außer Kraft gesetzt. Informationen zu den möglichen Werten für diese Parameter finden Sie in den Abschnitten zur FLVPlayback-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Festlegen des „source“-Parameters

Mit dem Parameter `source` geben Sie den Namen und den Speicherort der FLV-Datei 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. Das Dialogfeld sieht wie folgt aus:



Dialogfeld „Inhaltspfad“ für FLVPlayback-Komponente

Das Dialogfeld enthält zwei Kontrollkästchen, mit denen Sie die Abmessungen der FLVPlayback-Instanz festlegen und angeben, ob die Informationen zu Abmessungen und Cue-Points der FLV-Datei entnommen werden sollen. Weitere Informationen finden Sie unter [„Die Optionen für die FLV-Datei“](#) auf Seite 226.

Die Quelle

Geben Sie die URL oder den lokalen Pfad für die FLV-Datei oder für eine XML-Datei an, worin festgelegt ist, wie die FLV-Datei abgespielt wird. Wenn Ihnen der genaue Speicherort der FLV- oder XML-Datei nicht bekannt ist, klicken Sie auf das Ordnersymbol, um ein Dialogfeld für die Dateiauswahl zu öffnen, in dem Sie die Datei auswählen können. Falls sich die gesuchte FLV-Datei am oder unter dem Speicherort der Ziel-SWF-Datei 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. Wenn Sie den Namen einer lokalen XML-Datei eingeben, müssen Sie den Pfad und den Namen angeben.

Wenn Sie eine HTTP-URL angeben, wird die FLV-Datei als progressiver Download abgespielt. Wenn Sie eine RTMP-URL angeben, wird die FLV-Datei von einem Flash Media Server oder FVSS gestreamt. Eine URL zu einer XML-Datei könnte auch eine Streaming-FLV-Datei von einem Flash Media Server oder FVSS sein.

VORSICHT

Wenn Sie im Dialogfeld „Inhaltspfad“ auf „OK“ klicken, aktualisiert die Komponente den Wert des Parameters `cuePoints`, da dieser möglicherweise nicht mehr gültig ist, wenn der Inhaltspfad geändert wird. Dabei könnten alle deaktivierten Cue-Points verloren gehen, nicht jedoch ActionScript-Cue-Points. (Sie verlieren die deaktivierten Cue-Points nicht, wenn die neue FLV-Datei dieselben Cue-Points enthält, was möglich ist, wenn Sie nur den Pfad ändern.) Aus diesem Grund sollten Sie ggf. Cue-Points, die keine ActionScript-Cue-Points sind, mithilfe von ActionScript deaktivieren statt über das Dialogfeld „Cue-Points“.

Sie können auch den Speicherort einer SMIL-Datei angeben, in der beschrieben wird, wie mehrere FLV-Dateistreams für verschiedene Bandbreiten abgespielt werden sollen. Die SMIL-Datei verwendet zur Beschreibung der FLV-Dateien Synchronized Multimedia Integration Language (SMIL). Eine Beschreibung der SMIL-Datei finden Sie unter [„Verwenden von SMIL-Dateien“](#) auf Seite 260.

Sie können den Namen und den Speicherort der FLV-Datei auch angeben, indem Sie die ActionScript-Eigenschaft `FLVPlayback.source` und die Methoden `FLVPlayback.play()` und `FLVPlayback.load()` verwenden. Diese drei Alternativen haben Vorrang vor dem Parameter `source` im Komponenten-Inspektor. Weitere Informationen finden Sie in den Abschnitten zu `FLVPlayback.source`, `FLVPlayback.play()` und `FLVPlayback.load()` der `FLVPlayback`-Klasse im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Die Optionen für die FLV-Datei

Das Dialogfeld „Inhaltspfad“ enthält zwei Kontrollkästchen. Mit dem ersten, „An Abmessungen der FLV-Quelle anpassen“, legen Sie fest, ob die FLVPlayback-Instanz auf der Bühne an die Abmessungen der FLV-Quelldatei angepasst werden soll. Die FLV-Quelldatei enthält bevorzugte Abmessungen (Höhe und Breite) für das Abspielen. Wenn Sie die erste Option aktivieren, werden die Abmessungen der FLVPlayback-Instanz an diese bevorzugten Abmessungen angepasst. Diese Option ist jedoch nur verfügbar, wenn auch die zweite Option aktiviert wurde.

Die zweite Option, „FLV für Cue-Points und Abmessungen herunterladen“, kann nur ausgewählt werden, wenn es sich bei dem Inhaltspfad um eine HTTP- oder RTMP-URL handelt, die FLV-Datei also keine lokale Datei ist. (Jeder Pfad, der nicht mit „.flv“ endet, wird ebenfalls als Netzwerkpfad betrachtet, da es sich um eine XML-Datei handeln muss und diese auf FLV-Dateien an anderen Speicherorten verweisen kann.) Diese Option legt fest, ob ein Teil der FLV-Datei heruntergeladen oder gestreamt werden soll, um die Abmessungen der FLV-Datei sowie ggf. eingebettete Cue-Point-Informationen zu ermitteln. Flash verwendet die Abmessungen, um die Größe der FLVPlayback-Instanz zu ändern, und lädt die Cue-Point-Definitionen im Komponenten-Inspektor in den Parameter `cuePoints`. Wenn diese Option nicht ausgewählt wurde, ist die erste Option deaktiviert.

Verwenden der Live-Vorschau

Der Parameter `preview` der FLVPlayback-Komponente ermöglicht das Anzeigen eines Bilds der Quell-FLV-Datei in der Komponente auf der Bühne. Auf diese Weise können Sie auch Änderungen sehen, die Sie an der Komponente vorgenommen haben. Durch das Klicken auf den `preview`-Parameter wird das folgenden Dialogfeld geöffnet, in dem eine SWF-Datei der Quell-FLV-Datei abgespielt wird.



Dialogfeld zur Auswahl eines Bilds für die Live-Vorschau

Klicken Sie auf „OK“, wenn die FLV-Datei die Szene erreicht, die Sie für die Vorschau der Komponente auf der Bühne erfassen möchten. Wenn Sie ein Bild der FLV-Datei in der Komponente auf der Bühne anzeigen, können Sie es auf der Bühne in Relation zu anderen Elementen der Anwendung sehen.

Sie können das ausgewählte Bild auch exportieren, um es als PNG-Datei an einem beliebigen Speicherort zu speichern.

Vollbildunterstützung

Die ActionScript 3.0-Version der FLVPlayback-Komponente unterstützt den Vollbildmodus. Hierfür ist der Flash Player 9.0.28.0 erforderlich; außerdem muss der HTML-Code für die Anzeige im Vollbild korrekt eingerichtet sein. Einige vordefinierte Skins enthalten eine Schaltfläche mit Umschaltfunktion, um den Vollbildmodus ein- und auszuschalten. In der folgenden Abbildung ist diese Schaltfläche auf der rechten Seite zu sehen.



Vollbildsymbol in der Steuerungsleiste

Die Vollbildunterstützung verfügt über die folgenden Eigenschaften:

`fullScreenBackgroundColor`, `fullScreenSkinDelay` und `fullScreenTakeOver`. Weitere Informationen dazu finden Sie im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Layout für das Abspielen mehrerer FLV-Dateien

Die ActionScript 3.0-Komponente FLVPlayback verfügt über die Eigenschaft `align`, die festlegt, ob die FLV-Datei bei der Größenänderung zentriert oder oben, unten, links oder rechts in der Komponente positioniert wird. Zusätzlich zu den Eigenschaften `x`, `y`, `width` und `height` der Komponente, verfügt die ActionScript 3.0-Komponente auch über die Eigenschaften `registrationX`, `registrationY`, `registrationWidth` und `registrationHeight`. Anfänglich stimmen diese mit den Eigenschaften `x`, `y`, `width` und `height` überein. Beim Laden nachfolgender FLV-Dateien werden diese durch das automatische Layout nicht geändert, sodass die neue FLV-Datei 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 FLV-Dateien

Beim Laden einer FLV-Datei, die progressiv heruntergeladen wird, beginnt FLVPlayback erst dann mit dem Abspielen der FLV-Datei, wenn ein ausreichender Anteil der Datei heruntergeladen wurde, sodass die FLV-Datei von Anfang bis Ende abgespielt werden kann.

Wenn die FLV-Datei schon abgespielt werden soll, bevor ausreichen davon heruntergeladen wurde, rufen Sie die Methode `play()` ohne Parameter auf.

Wenn Sie in den Wartezustand, bis ein ausreichender Teil der FLV-Datei heruntergeladen wurde, zurückkehren möchten, 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 FLV-Datei 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, um Punkte in der FLV-Datei mit anderen Aktionen auf der Webseite zu synchronisieren.

Es gibt drei Arten von Cue-Points: Cue-Points für Navigation, Cue-Points für Ereignisse und ActionScript-Cue-Points. Die Cue-Points für Navigation und Ereignisse werden auch als *eingebettete* Cue-Points bezeichnet, da sie in den FLV-Dateistream 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, dass sich zwischen den Einzelbild-Frames im FLV-Dateistream 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. Sie können Cue-Points für die Navigation und für Ereignisse mithilfe des Videoimportassistenten oder des Flash Video Encoders in eine FLV-Datei einbetten. Weitere Informationen zum Videoimportassistenten und zum Flash Video Encoder finden Sie in [Kapitel 16, „Mit Videos arbeiten“](#), in *Flash verwenden*.

Ein *ActionScript-Cue-Point* ist ein externer Cue-Point, den Sie über das 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 Eigenschaft

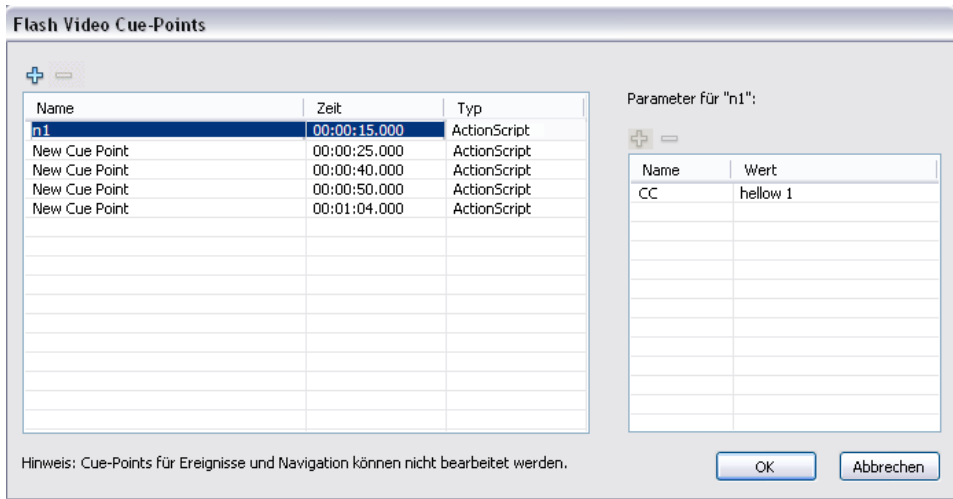
`FLVPlayback.playheadUpdateInterval` im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

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 Namen/Wert-Paaren.

Wenn ein Ereignis `cuePoint` eintritt, ist das Cue-Point-Objekt im Ereignisobjekt über die Eigenschaft `info` verfügbar. Weitere Informationen finden Sie unter „[Abfragen von cuePoint-Ereignissen](#)“ auf Seite 234.

Dialogfeld „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.

So fügen Sie einen ActionScript-Cue-Point hinzu:

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 „Neuer Cue-Point“ und bearbeiten Sie den Text, um einen Namen für den Cue-Point einzugeben.

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.

So löschen Sie einen 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.

So aktivieren oder deaktivieren Sie einen eingebetteten Cue-Point aus 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 ActionScript mit Cue-Points

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 bereit zum Abspielen 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.

```
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 Methode

`FLVPlayback.addASCuePoint()` im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Abfragen von cuePoint-Ereignissen

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.

```
my_FLVPlayback.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace("Elapsed time in seconds: " + my_FLVPlayback.playheadTime);
    trace("Cue point name is: " + eventObject.info.name);
    trace("Cue point type is: " + eventObject.info.type);
}
```

Weitere Informationen zum Ereignis `cuePoint` finden Sie im Abschnitt zum Ereignis `FLVPlayback.cuePoint` im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

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:

```
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_FLVPlybk.source = "http://www.helpexamples.com/flash/video/
  cuepoints.flv"
var rtn_obj:Object; //create cue point object
my_FLVPlybk.addASCuePoint(2.02, "ASpt"); //add AS cue point
my_FLVPlybk.addASCuePoint(3.4, "ASpt"); //add 2nd ASpt
my_FLVPlybk.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:VideoEvent):void {
    rtn_obj = my_FLVPlybk.findCuePoint("ASpt", CuePointType.ACTIONSCRIPT);
    traceit(rtn_obj);
    rtn_obj = my_FLVPlybk.findNextCuePointWithName(rtn_obj);
    traceit(rtn_obj);
}
function traceit(cuePoint:Object):void {
    trace("Cue point name is: " + cuePoint.name);
    trace("Cue point time is: " + cuePoint.time);
    trace("Cue point type is: " + cuePoint.type);
}
```

Weitere Informationen zum Suchen von Cue-Points finden Sie in den Abschnitten über die Methoden `FLVPlayback.findCuePoint()`, `FLVPlayback.findNearestCuePoint()` und `FLVPlayback.findNextCuePointWithName()` im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Vor- oder Rücklauf zu 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.

```
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 über die Methoden `FLVPlayback.seekToNavCuePoint()`, `FLVPlayback.seekToNextNavCuePoint()` und `FLVPlayback.seekToPrevNavCuePoint()` im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Aktivieren und Deaktivieren von eingebetteten FLV-Datei-Cue-Points

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 über die Methoden `FLVPlayback.isFLVCuePointEnabled()` und `FLVPlayback.setFLVCuePointEnabled()` im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Entfernen von ActionScript-Cue-Points

Sie können ActionScript-Cue-Points mithilfe der Methode `removeASCuePoint()` entfernen. Im folgenden Beispiel wird der Cue-Point `ASpt2` entfernt, wenn `ASpt1` eintritt.

```
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(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace("Cue point name is: " + eventObject.info.name);
    if (eventObject.info.name == "ASpt1") {
        my_FLVPlayback.removeASCuePoint("ASpt2");
        trace("Removed cue point ASpt2");
    }
}
```

Weitere Informationen finden Sie im Abschnitt zur Methode `FLVPlayback.removeASCuePoint()` im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Abspielen von mehreren FLV-Dateien

Sie können FLV-Dateien in einer `FLVPlayback`-Instanz nacheinander abspielen, indem Sie einfach eine neue URL in die Eigenschaft `source` laden, wenn eine FLV-Datei fertig abgespielt wurde. Im folgenden ActionScript-Code wird z. B. auf das Ereignis `complete` gewartet, das eintritt, wenn eine FLV-Datei fertig abgespielt wurde. Wenn dieses Ereignis eintritt, setzt der Code den Namen und den Speicherort einer neuen FLV-Datei 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 mehrerer Video-Player

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 über die Eigenschaften [FLVPlayback.activeVideoPlayerIndex](#) und [FLVPlayback.visibleVideoPlayerIndex](#) im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Mit dem folgenden ActionScript-Code wird die Eigenschaft `source` geladen, um eine FLV-Datei im Standard-Player abzuspielen, und es wird ein Cue-Point hinzugefügt. Wenn das Ereignis `ready` eintritt, öffnet die Ereignisprozedur einen zweiten Video-Player, indem die Eigenschaft `activeVideoPlayerIndex` auf die Nummer 1 gesetzt wird. Es werden eine FLV-Datei und ein Cue-Point für den zweiten Video-Player angegeben; dann wird der Standard-Player (0) wieder zum aktiven Video-Player.

```
/**
 * Requires:
 *   - FLVPlayback component on the Stage with an instance name of my_FLVPlayback
 */
// add a cue point to the default player
import fl.video.*;
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/clouds.flv";
my_FLVPlayback.addASCuePoint(3, "1st_switch");
my_FLVPlayback.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:VideoEvent):void {
    // add a second video player and create a cue point for it
    my_FLVPlayback.activeVideoPlayerIndex = 1;
    my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/
    water.flv";
    my_FLVPlayback.addASCuePoint(3, "2nd_switch");
    my_FLVPlayback.activeVideoPlayerIndex = 0;
};
```

Wenn Sie beim Abspielen einer FLV-Datei zu einer anderen wechseln möchten, müssen Sie in Ihrem ActionScript-Code den Wechsel 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_FLVPlayback.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_FLVPlayback.pause(); //pause the first FLV file
        my_FLVPlayback.activeVideoPlayerIndex = 1; // make the 2nd player active
        my_FLVPlayback.visibleVideoPlayerIndex = 1; // make the 2nd player
                                                // visible
        my_FLVPlayback.play(); // begin playing the new player/FLV
    } else if (eventObject.vp == 1) {
        my_FLVPlayback.pause(); // pause the 2nd FLV
        my_FLVPlayback.activeVideoPlayerIndex = 0; // make the 1st player active
        my_FLVPlayback.visibleVideoPlayerIndex = 0; // make the 1st player
        visible
        my_FLVPlayback.play(); // begin playing the 1st player
    }
}
my_FLVPlayback.addEventListener(VideoEvent.COMPLETE, complete_listener);
function complete_listener(eventObject:VideoEvent):void {
    trace("Hit complete event for player: " + eventObject.vp);
    if (eventObject.vp == 0) {
        my_FLVPlayback.activeVideoPlayerIndex = 1;
        my_FLVPlayback.visibleVideoPlayerIndex = 1;
        my_FLVPlayback.play();
    } else {
        my_FLVPlayback.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: leerer String, 0 und `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 die Lautstärke, die Position, die Abmessungen, die Sichtbarkeit und Benutzeroberflächen-Steuerelemente gesteuert werden, sind immer global, und ihr Verhalten wird durch das Einstellen der Eigenschaft `activeVideoPlayerIndex` nicht beeinflusst. Weitere Informationen zu diesen Methoden und Eigenschaften sowie die Auswirkungen beim Festlegen der Eigenschaft `activeVideoPlayerIndex` finden Sie im Abschnitt über die Eigenschaft `FLVPlayback.activeVideoPlayerIndex` im *Komponenten-Referenzhandbuch für ActionScript 3.0*. 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 Eigenschaft `FLVPlayback.visibleVideoPlayerIndex` im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Streaming 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-Provider verfügbar ist. „Native Bandbreitenerkennung“ bedeutet, dass die Bandbreitenerkennung in den Streaming-Server integriert ist, wodurch eine bessere Leistung gewährleistet ist. Fragen Sie Ihren Provider, ob die native Bandbreitenerkennung verfügbar ist. Verwenden Sie eine URL, z. B. „`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` mit dem Wert `true` belegen. Weitere Informationen finden Sie im Abschnitt zur Eigenschaft `FLVPlayback.isLive` im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Weitere Informationen zum Verwalten des 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/.

Bei nativer Bandbreitenerkennung oder nicht vorhandener Bandbreitenerkennung

Die NCManagerNative-Klasse ist eine Unterklasse von NCManager, die die native Bandbreitenerkennung unterstützt. Einige Flash Video Streaming Service-Provider unterstützen möglicherweise die native Bandbreitenerkennung. Wenn Sie NCManagerNative verwenden, werden auf dem Flash Media Server keine besonderen Dateien benötigt. 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 standardmäßigen NCManager-Klasse zu verwenden, fügen Sie im ersten Bild Ihrer FLA-Datei die folgenden Codezeilen hinzu:

```
import fl.video*;  
VideoPlayer.INCManagerClass = fl.video.NCManagerNative;
```

Bei nicht-nativer Bandbreitenerkennung

Falls Ihr Flash Video Streaming Service-Provider keine native Bandbreitenerkennung zur Verfügung stelle, Sie jedoch Bandbreitenerkennung benötigen, müssen Sie Ihrer Flash Media Server FLV-Anwendung die Datei „main.asc“ hinzufügen. Die Datei „main.asc“ finden Sie im Flash-Anwendungsordner unter „Adobe Flash CS3/Samples and Tutorials/Samples/Components/FLVPlayback/main.asc“.

So richten Sie Ihren Flash Media Server für das Streaming von FLV-Dateien ein:

1. Erstellen Sie in Ihrem Flash Media Server-Anwendungsordner einen Ordner und benennen 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 Vorgehensweisen individuell anpassen.

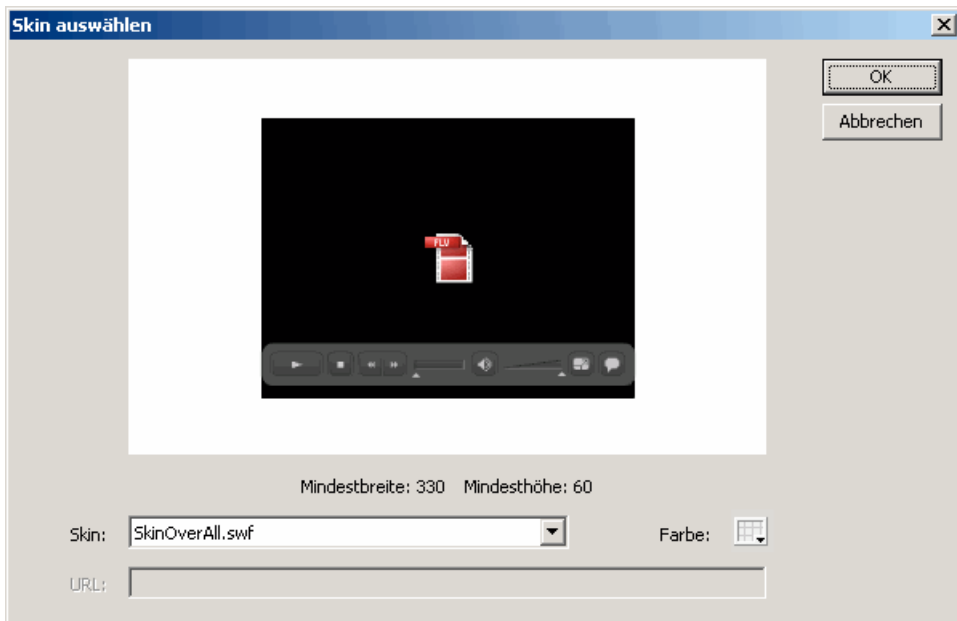
Sie haben die folgenden Möglichkeiten zum Anpassen der FLVPlayback-Komponente: Wählen Sie eine vordefinierte Skin aus, verwenden Sie Skins für einzelne FLV Playback Custom UI-Komponenten, oder erstellen Sie eine neue Skin. Das Verhalten von Skins lässt sich mit den FLVPlayback-Eigenschaften modifizieren.

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.

Auswahl vordefinierter Skins

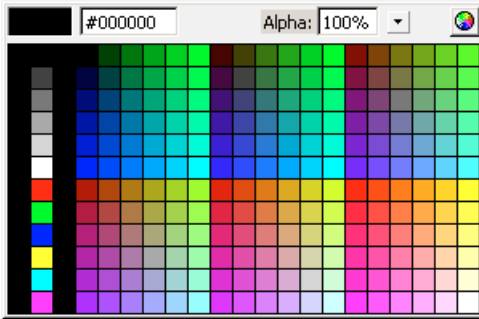
Sie können eine Skin für die FLVPlayback-Komponente auswählen, indem Sie im Komponenten-Inspektor auf die Zelle Wert für den Parameter `skin` klicken. Klicken Sie dann auf das Lupensymbol, um das Dialogfeld „Skin auswählen“ zu öffnen, in dem Sie eine Skin auswählen oder eine URL eingeben können, die den Speicherort einer Skin-SWF-Datei angibt.



FLVPlayback-Dialogfeld „Skin auswählen“

Skins, die im Popupmenü Skin aufgelistet sind, befinden sich im Flash-Ordner „Configuration/FLVPlayback Skins/ActionScript 3.0/“. Wenn neue Skins in diesem Dialogfeld verfügbar sein sollen, müssen Sie die entsprechenden SWF-Dateien nach dem Erstellen in diesem Ordner ablegen. Der Name wird im Popupmenü mit der Dateierweiterung „.swf“ angezeigt. Informationen zum Erstellen eines neuen Skinsatzes finden Sie unter [„Erstellen neuer Skins“ auf Seite 253](#).

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.



Klicken Sie auf ein Farbfeld oder geben Sie dessen numerischen Wert in das Textfeld ein, um eine Farbe auszuwählen. 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 für die Eigenschaft `skinBackgroundColor` einen `0xRRGGBB`-Wert (Rot, Grün, Blau) ein. Legen Sie für die Eigenschaft „`skinBackgroundAlpha`“ eine Zahl zwischen 0,0 und 1,0 fest. Im folgenden Beispiel wird `skinBackgroundColor` auf „`0xFF0000`“ (rot) und `skinBackgroundAlpha` auf „`0,5`“ festgelegt.

```
my_FLVPlayback.skinBackgroundColor = 0xFF0000;
my_FLVPlayback.skinBackgroundAlpha = .5;
```

Die Standardwerte sind die zuletzt vom Benutzer ausgewählten Werte.

Wenn Sie der `FLVPlayback`-Komponente Skins zuweisen möchten, indem Sie die `FLVPlayback Custom UI`-Komponenten verwenden, wählen Sie im Popupmenü „Aus“.

Zuweisen von Skins für einzelne 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` mit dem Wert `exactFit` belegt 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 verwendet werden. Wenn Sie sie auf dieselbe Zeitleiste und dasselbe Bild wie die FLVPlayback-Komponente platzieren und in der Komponente keine Skin festgelegt ist, stellt die FLVPlayback-Komponente automatisch eine Verbindung zu ihnen her. Wenn Sie mehrere FLVPlayback-Komponenten auf der Bühne haben, oder wenn sich das benutzerdefinierte Steuerelement und die FLVPlayback-Instanz 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 für die Anzeigestatus auf der Bühne: „normal“, „over“ (Mauszeiger ist über der Schaltfläche), „down“ (Schaltfläche wurde gedrückt) und „disabled“ (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“ müssen Sie allerdings trotzdem auswählen.)

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.

Die 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 Schaltflächen „Abspielen“ und „Pause“, für MuteButton die Schaltflächen „Ton aus“ und „Ton ein“, für FullScreenButton die Schaltflächen „Vollbild ein“ und „Vollbild aus“, für CaptionButton die Schaltflächen „Untertitel ein“ und „Untertitel aus“. Für das Skinning dieser Schaltflächen weisen Sie jeweils den beiden internen Schaltflächen Skins zu wie unter [„Zuweisen von Skins für einzelne FLV Playback Custom UI-Komponenten“ auf Seite 245](#) beschrieben; weitere Schritte sind nicht erforderlich.

CaptionButton ist für die FLVPlaybackCaptioning-Komponente gedacht und muss dieser Komponente (nicht der FLVPlayback-Komponente) zugewiesen werden.

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 der 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 Zwischenspeichern-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“ müssen Sie allerdings trotzdem auswählen.

Vielleicht ist Ihnen im Hintergrund des Handle-Movieclips ein Rechteck mit dem Alphawert 0 aufgefallen. Dieses Rechteck vergrößert den Kollisionsbereich des Griiffs, 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 Griiffs 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 Methode `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 derselben 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 Griiffs (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 Griiffs positionieren, stehen die linke und rechte Seite des Griiffs 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.

Die Variable `handleY` legt die y-Position des Griiffs relativ zur Bar-Instanz fest. Dies basiert auf dem Registrierungspunkt des Movieclips. Der Registrierungspunkt desselben Griiffs 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 allerdings einen beliebigen, keinen oder beide dieser Movieclips haben. Die Struktur dieser Leisten ist dieselbe, ihr Verhalten ist ähnlich, allerdings werden unterschiedliche Werte angezeigt. Ein Fortschritts-Movieclip wird gefüllt, wenn die FLV-Datei heruntergeladen wird (dies ist nur bei HTTP-Downloads sinnvoll, beim Streaming vom 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, benennen 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.

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 Ihre benutzerdefinierten UI-Komponenten auf derselben Zeitleiste und im selben Bild wie die FLVPlayback-Komponente platzieren und die Eigenschaft `skin` nicht festgelegt haben, verbindet sich die FLVPlayback-Komponente automatisch mit den UI-Komponenten, ohne dass Sie ActionScript hinzufügen müssen.

Wenn Sie mehrere FLVPlayback-Komponenten auf der Bühne haben oder wenn sich die benutzerdefinierten Steuerelemente und die FLVPlayback-Komponente nicht auf derselben Zeitleiste befinden, müssen Sie ActionScript schreiben, um die Custom UI-Komponenten mit der Instanz der FLVPlayback-Komponente zu verbinden. Dazu benennen Sie zunächst die FLVPlayback-Instanz und dann weisen Sie 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 Control-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;

```

Beispiel

Mit den folgenden Schritten werden die Steuerelemente StopButton, PlayPauseButton, MuteButton und SeekBar erstellt:

So erstellen Sie die Steuerelemente StopButton, PlayPauseButton, MuteButton und SeekBar:

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 Parameter „Skin“ 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. Benennen Sie die einzelnen Schaltflächen im Eigenschafteninspektor (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 Taste <Ctrl> (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. Legen Sie im Eigenschafteninspektor die Breite (B:) auf **40** und die Höhe (H:) auf **20** fest. 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 die folgenden Schritte für jeden dieser Movieclips aus:
- Wählen Sie die Option „Alles auswählen“.
 - Ändern Sie die Farbe zu rot (#FF0000).
 - Skalieren Sie um 300%.
 - Ändern Sie die X:-Position des Inhalts auf 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 als rot (#FF0000) fest.
18. Doppelklicken Sie im Ordner „FLVPlayback Skins/Seek Bar“ auf „SeekBarProgress“ und legen Sie die Farbe als gelb (#FFFF00) fest.
19. Doppelklicken Sie im Ordner „FLVPlayback Skins/Seek Bar“ auf „SeekBarHandle“ und legen Sie die Farbe als 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 benennen Sie die Instanz **my_seekbar**.

22.Fügen Sie im Bedienfeld „Aktionen“ in Bild 1 der Hauptzeitleiste eine Importanweisung für die Videoklassen ein und weisen Sie die Namen der Schaltflächen und der Vor-/Rücklaufleiste den entsprechenden FLVPlayback-Eigenschaften zu, wie im folgenden Beispiel:

```
import fl.video.*;
my_FLVPlayback.stopButton = my_stopbbtn;
my_FLVPlayback.playPauseButton = my_playpausbbtn;
my_FLVPlayback.muteButton = my_mutebbtn;
my_FLVPlayback.seekBar = my_seekbar;
```

23.Drücken Sie <Strg>+<Eingabe>, um den Film zu testen.

Erstellen neuer Skins

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.

Da Sie die Farbe einer Skin unabhängig von der Auswahl der Skin festlegen können, brauchen Sie die FLA-Datei nicht zu bearbeiten, wenn Sie eine andere Farbe verwenden möchten.

Wenn Sie eine Skin mit einer bestimmten Farbe erstellen und Sie nicht möchten, dass die Skin im Dialogfeld „Skin auswählen“ bearbeitet werden kann, verwenden Sie

`this.border_mc.colorMe = false;` im ActionScript-Code der Skin-FLA-Datei.

Informationen zum Festlegen einer Farbe für die Skin finden Sie unter [„Auswahl vordefinierter Skins“ auf Seite 243](#).

Wenn Sie die FLA-Dateien der installierten Flash-Skins betrachten, kommen Ihnen bestimmte Elemente auf der Bühne möglicherweise überflüssig vor, vieler dieser Elemente wurden jedoch in Pfadebenen platziert. Was tatsächlich in der SWF-Datei angezeigt wird, können Sie schnell mithilfe der Live-Vorschau unter Verwendung von Scale-9 überprüfen.

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, befinden sich die Movieclips der Skin auf der Hauptzeitleiste. 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 „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`. Was 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 ist nur die horizontale bzw. vertikale Position (*x* bzw. *y*) sowie die Höhe und die Breite des Platzhalters.

Neben den Standardsteuerelementen können Sie noch 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 wird nur benötigt, um zum Festlegen des Layouts ActionScript für die Clips verwenden zu können.

Der Clip „border_mc“ ist ein Sonderfall. Wenn Sie die Eigenschaft `FLVPlayback.skinAutoHide` mit dem Wert `true` belegen, 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 260](#).

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:

```
border_mc.colorMe = true;
```

ActionScript

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 Zustand jeder Komponente angibt. Diese Klassennamen werden in der Datei „SkinOverAll fla“ aufgeführt. Für die Schaltflächen „Pause“ und „Abspielen“ 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 alle auf Führungsebenen und können nicht exportiert werden. 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()
```

Nach diesem Abschnitt folgt ActionScript-Code, der die Mindestbreite 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
<code>anchorLeft</code>	Boolesch. Positioniert das Steuerelement relativ zur linken Seite der FLVPlayback-Instanz. Der Standardwert ist <code>true</code> , es sei denn, dass <code>anchorRight</code> ausdrücklich auf <code>true</code> festgelegt wurde – in diesem Fall ist der Standardwert <code>false</code> .
<code>anchorRight</code>	Boolesch. Positioniert das Steuerelement relativ zur rechten Seite der FLVPlayback-Instanz. Der Standardwert ist <code>false</code> .
<code>anchorBottom</code>	Boolesch. Positioniert das Steuerelement relativ zum unteren Rand der FLVPlayback-Instanz. Der Standardwert ist <code>true</code> , es sei denn, dass <code>anchorTop</code> ausdrücklich auf <code>true</code> festgelegt wurde – in diesem Fall ist der Standardwert <code>false</code> .
<code>anchorTop</code>	Boolesch. 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 Steuerelemente wurden 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 enthält 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 247](#).

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 „bufferingBarFill_mc“.

Vor-/Rücklaufleiste und Lautstärkeleiste

Die Vor-/Rücklaufleiste enthält 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, und diese funktioniert mit verschachtelten Movieclips nicht gut.

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 sind aber keine Verzerrungen sichtbar. 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.

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 Movieclips „SeekBar“ 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 [„Movieclips für Fortschritt und Füllung“ auf Seite 249](#).

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 247. Eine Ausnahme ist der Griff, der auf andere Weise implementiert wird. Weitere Informationen hierzu finden Sie im folgenden Abschnitt.

Handle

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 Movieclips „SeekBar“ 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 der Griff 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 Schaltflächen „Forward“ und „Back“ 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` festlegen, werden die SeekBar-Komponente und ihr Griff von der FLVPlayback-Komponente ausgeblendet, und die Schaltflächen „Play“ und „Pause“ 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 Schaltflächen „Play“ und „Pause“ deaktivieren, wenn die Komponente mit der Zwischenspeicherung 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 für diese Eigenschaft lautet `false`.

Verwenden von SMIL-Dateien

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.

Geben Sie den Speicherort einer SMIL-Datei mit dem Parameter `source` oder der Eigenschaft `FLVPlayback.source` (ActionScript) an. Weitere Informationen finden Sie unter [„Die Quelle“ auf Seite 225](#) und im Abschnitt zur Eigenschaft `FLVPlayback.source` im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

Das folgende Beispiel beschreibt 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 entweder einen einzelnen Link zu einer FLV-Quelldatei einfügen, oder – wenn Sie mehrere Dateien für verschiedene Bandbreiten von einem FMS streamen (wie im vorigen Beispiel) – Sie verwenden das Tag `<switch>`, 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 verwendet:

```

<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.

Übergeordneter 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 erforderliche 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.

Übergeordneter 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.

Übergeordneter 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`, welches die erforderliche Mindestbandbreite sowie die Attribute `src` und `dur` angibt.

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>
```

Die FLVPlaybackCaptioning-Komponente

6

Mit der FLVPlayback-Komponente können Sie einen Video-Player in Ihre Adobe Flash CS3 Professional-Anwendung einbinden, um heruntergeladene Adobe Flash Video-Dateien (FLV) und Streaming-FLV-Dateien abzuspielen. Weitere Informationen zur FLVPlayback-Komponente finden Sie in [Kapitel 5, „Die FLVPlayback-Komponente“](#) auf Seite 217.

Die FLVPlaybackCaptioning-Komponente ermöglicht Ihnen, die Unterstützung für Closed Captioning (codierter Anzeigetext, Untertitel) in Ihr Video einzubinden. Die Captioning-Komponente unterstützt den W3C-Standard Timed Text im XML-Format und enthält die folgenden Funktionen:

- **Untertitel mit eingebetteten Ereignis-Cue-Points** Verknüpfen Sie eingebettete Ereignis-Cue-Points in einer FLV-Datei mit XML, um Untertitel hinzuzufügen, anstatt eine Timed Text-XML-Datei zu verwenden.
- **Mehrere FLVPlaybackCaptioning-Instanzen** Erstellen Sie mehrere FLVPlaybackCaptioning-Instanzen für mehrere FLVPlayback-Instanzen.
- **Schaltfläche mit Umschaltfunktion** Ermöglichen Sie dem Benutzer das Ein- und Ausschalten von Untertiteln oder Begleittext durch eine Schaltfläche mit Umschaltfunktion.

Verwenden der FLVPlaybackCaptioning-Komponente

Sie verwenden die FLVPlaybackCaptioning-Komponente mit einer oder mehreren FLVPlayback-Komponenten. 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. Zusätzlich können Sie die FLVPlayback-Komponente mithilfe verschiedener Parameter anpassen.

Hinzufügen von Untertiteln zur FLVPlayback-Komponente

Sie können die FLVPlaybackCaptioning-Komponente jeder beliebigen FLVPlayback-Komponente hinzufügen. Weitere Informationen zum Hinzufügen von FLVPlayback-Komponenten zu einer Anwendung finden Sie unter „[Erstellen von Anwendungen mit der FLVPlayback-Komponente](#)“ auf Seite 220.

So fügen Sie die FLVPlaybackCaptioning-Komponente aus dem Bedienfeld „Komponenten“ hinzu:

1. Öffnen Sie im Bedienfeld „Komponenten“ den Ordner „Video“.
2. Ziehen Sie die FLVPlaybackCaptioning-Komponente (oder doppelklicken Sie darauf) und fügen Sie sie derselben Bühne hinzu wie die FLVPlayback-Komponente, der Sie Untertitel hinzufügen möchten.

HINWEIS

Adobe stellt zwei Beispieldateien 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). Diese Dateien finden Sie unter <http://www.helpexamples.com/flash/video>.

3. (Optional) Ziehen Sie CaptionButton-Komponente auf dieselbe Bühne wie die FLVPlayback- und FLVPlaybackCaptioning-Komponenten. Mit der CaptionButton-Komponente kann der Benutzer die Anzeige der Untertitel ein- und ausschalten.

HINWEIS

Um die CaptionButton-Komponente zu aktivieren, ziehen Sie sie auf dieselbe Bühne wie die FLVPlayback- und FLVPlaybackCaptioning-Komponenten.

4. Während auf der Bühne die FLVPlaybackCaptioning-Komponente ausgewählt ist, legen Sie auf der Registerkarte „Parameter“ des Eigenschafteninspektors die folgenden erforderlichen Informationen fest:

- Belegen Sie `showCaptions` mit `true`.
- Geben Sie die `source` der Timed Text-XML-Datei an, die heruntergeladen werden soll.

TIPP

Während Sie in Flash arbeiten, um die Untertitel zu testen, sollten Sie die Eigenschaft `showCaptions` auf `true` einstellen. Wenn Sie jedoch die `CaptionButton`-Komponente verwenden, damit der Benutzer die Untertitel ein- und ausschalten kann, sollten Sie die Eigenschaft `showCaptions` auf `false` einstellen.

Es stehen noch weitere Parameter zur Verfügung, mit denen Sie die FLVPlaybackCaptioning-Komponente anpassen können. Weitere Informationen finden Sie in „[Anpassen der FLVPlaybackCaptioning-Komponente](#)“ auf Seite 279 und im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

5. Wählen Sie „Steuerung“ > „Film testen“, um das Video zu starten.

Close collapsed procedure

So erstellen Sie eine Instanz mithilfe von ActionScript dynamisch:

1. Ziehen Sie die FLVPlayback-Komponente aus dem Bedienfeld „Komponenten“ in das Bedienfeld „Bibliothek“ („Fenster“ > „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.*;
```

HINWEIS

Das folgende Beispiel gilt für Adobe Flash CS3 für Windows. Der Speicherort der FLVPlayback-Skins auf dem Macintosh ist: Macintosh HD/Applications/Adobe Flash CS3/ Configuration/FLVPlayback Skins/ActionScript 3.0/ SkinUnderPlaySeekCaption.swf.

```
var my_FLVPlaybk = new FLVPlayback();
my_FLVPlaybk.x = 100;
my_FLVPlaybk.y = 100;
addChild(my_FLVPlaybk);
my_FLVPlaybk.skin = "install_drive:/Program Files/Adobe/Adobe Flash CS3/
  en/Configuration/FLVPlayback Skins/ActionScript 3.0/
  SkinUnderPlaySeekCaption.swf";
my_FLVPlaybk.source = "http://www.helpexamples.com/flash/video/
  caption_video.flv";
var my_FLVPlaybkcap = new FLVPlaybackCaptioning();
```

```
addChild (my_FLVPlaybackcap);
my_FLVPlaybackcap.source = "http://www.helpexamples.com/flash/video/
caption_video.xml";
my_FLVPlaybackcap.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 die Eigenschaft „skin“ mit ActionScript festlegen. Wenn Sie eine Skin mit 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.

Festlegen der Parameter für die FLVPlaybackCaptioning-Komponente

Die folgenden Parameter können für jede Instanz der FLVPlaybackCaptioning-Komponente im Eigenschafteninspektor oder im Komponenten-Inspektor festgelegt werden, um die Komponente weiter anzupassen. In der folgenden Liste sind die Eigenschaften mit einer kurzen Beschreibung aufgeführt:

- `autoLayout`. Legt fest, ob die FLVPlaybackCaptioning-Komponente die Größe des Untertitelbereichs festlegt. Der Standardwert lautet `true`.
- `captionTargetName`. Gibt den Namen der TextField- oder MovieClip-Instanz an, die die Untertitel enthält. Der Standardwert lautet „auto“.
- `flvPlaybackName`. Gibt den Namen der FLVPlayback-Instanz an, für die Sie Untertitel verwenden möchten. Der Standardwert lautet „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`. Gibt den Speicherort der Timed Text-XML-Datei an.

Weitere Informationen zu allen FLVPlaybackCaptioning-Parametern finden Sie im [Komponenten-Referenzhandbuch für ActionScript 3.0](#).

Festlegen des „source“-Parameters

Mit dem Parameter `source` geben Sie den Namen und den Speicherort der Timed Text-XML-Datei an, welche die Untertitel für den Film enthält. Geben Sie die URL direkt in den `source`-Zelle des Komponenten-Inspektors ein.

Anzeigen von Untertiteln

Damit Captions angezeigt werden, müssen Sie den Parameter `showCaptions` auf `true` einstellen.

Weitere Informationen zu allen Parametern der FLVPlaybackCaptioning-Komponente finden Sie im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

In den vorstehenden Beispielen haben Sie gelernt, wie Sie die FLVPlaybackCaptioning-Komponente erstellen und aktivieren, um Untertitel anzuzeigen. Es gibt zwei Arten von Quellen, die Sie für Untertitel verwenden können: (1) eine Timed Text-XML-Datei mit den Untertiteln oder (2) eine XML-Datei mit dem Text für die Untertitel, die mit eingebetteten Cue-Points verknüpft sind.

Verwenden von Timed Text-Untertiteln

Die FLVPlaybackCaptioning-Komponente ermöglicht die Verwendung von Untertiteln für die zugeordnete FLVPlayback-Komponente, indem eine Timed Text (TT)-XML-Datei heruntergeladen wird. Weitere Informationen zum Format Timed Text finden Sie im Abschnitt „AudioVideo Timed Text“ unter <http://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 alle unterstützten Timed Text-Tags finden Sie in *Anhang A, „Timed Text-Tags“*.

Die FLVPlaybackCaptioning-Komponente unterstützt die folgenden Timed Text-Tags:

- Absatzformatierung
 - Richten Sie Absätze rechtsbündig, linksbündig oder zentriert aus
- Textformatierung
 - Legen Sie die Textgröße als absolute Pixelgrößen oder im Deltastil fest (z. B. +2, -4)
 - Legen Sie die Farbe und die Schriftart für den Text fest
 - Formatieren Sie Text fett und kursiv
 - Richten Sie den Text aus

■ Sonstige Formatierung

- Legen Sie die Hintergrundfarbe der TextField-Komponente für Untertitel fest
- Legen Sie die Hintergrundfarbe der TextField-Komponente für Untertitel als transparent fest (Alpha 0)
- Stellen Sie den Zeilenumbruch der TextField-Komponente für Untertitel ein oder aus

Die FLVPlaybackCaptioning-Komponente verwendet denselben Zeitcode wie die FLV-Datei. Jeder Untertitel benötigt ein `begin`-Attribut, das festlegt, wann der Untertitel eingeblendet werden soll. Falls der Untertitel kein `dur`- oder `end`-Attribut hat, wird er beim Erscheinen des nächsten Untertitels bzw. am Ende der FLV-Datei ausgeblendet.

Das folgende Beispiel zeigt eine Timed Text-XML-Datei. Diese Datei („caption_video.xml“) enthält die Untertitel für die Datei „caption_video.flv“. Diese Dateien finden Sie unter <http://www.helpexamples.com/flash/video>.

```
<?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:fontSi
ze="+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>

```

Verwenden von Cue-Points mit Untertiteln

Cue-Points ermöglichen die Interaktion mit einem Video. Sie können zum Beispiel das Abspielen einer FLV-Datei beeinflussen oder an bestimmten Punkten im Video Text einblenden. Wenn Sie keine Timed Text XML-Datei haben, die Sie mit einer FLV-Datei verwenden können, betten Sie Ereignis-Cue-Points in eine FLV-Datei ein und verknüpfen Sie diese Punkte mit Text. In diesem Abschnitt finden Sie Informationen zu den Cue-Point-Standardeinstellungen für die die FLVPlaybackCaptioning-Komponente sowie einen kurzen Überblick über das Verknüpfen dieser Cue-Points mit Text. Weitere Informationen zum Einbetten von Ereignis-Cue-Points mit dem Videoimportassistenten oder mit Flash Video Encoder finden Sie in [Kapitel 16, „Mit Videos arbeiten“](#), in *Flash verwenden*.

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`. ActionScript-Cue-Points der FLVPlaybackCaptioning-Komponente 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 besteht aus einer Reihe von positiven Ganzzahlen, die jedes Mal erhöht wird, damit der Name eindeutig bleibt. Das Präfix enthält die Versionsnummer, die auch mit der FLVPlayback-Versionsnummer übereinstimmt. Für Adobe Flash CS3 müssen Sie die Versionsnummer auf 2.0 festlegen.
- **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 HTML-formatierte Text des Untertitels. Dieser Text wird direkt an die Eigenschaft `TextField.htmlText` übergeben. Die `FLVPlaybackCaptioning`-Komponente verfügt über die optionale Eigenschaft `text:n`, die mehrere Sprachspuren unterstützt. Weitere Informationen finden Sie unter „[Unterstützung mehrerer Sprachspuren mit eingebetteten Cue-Points](#)“ auf Seite 278.
 - `endTime:Number`. Der Zeitpunkt, zu dem der Untertitel 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 wurde (d. h. wenn die `FLVPlayback`-Instanz das Ereignis `VideoEvent.COMPLETE` auslöst). Geben Sie die Eigenschaft `endTime:Number` in Sekunden an.
`an.backgroundColor:uint`. Dieser Parameter legt `TextField.backgroundColor` fest. Diese Eigenschaft ist optional.
 - `backgroundColorAlpha:Boolean`. Wenn „backgroundColor“ einen Alphawert von 0 % aufweist, legt der Parameter `TextField.background = !backgroundColor` fest. Diese Eigenschaft ist optional.
 - `wrapOption:Boolean`. Dieser Parameter legt „`TextField.wordWrap`“ fest. Diese Eigenschaft ist optional.

Erstellen von Untertiteln für eingebettete Ereignis-Cue-Points

Wenn Sie keine Timed Text XML-Datei haben, die die Untertitel für Ihre FLV-Datei enthält, können Sie eine XML-Datei, die Untertitel enthält, mit eingebetteten Ereignis-Cue-Points verknüpfen. Im XML-Beispiel wird vorausgesetzt, dass Sie die folgenden Schritte ausgeführt haben, um eingebettete Ereignis-Cue-Points in Ihrem Video 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 und eine `FLVPlaybackCaptioning`-Komponente auf die Bühne.
- Legen Sie die source-Eigenschaften für die `FLVPlayback`-Komponente und die `FLVPlaybackCaptioning`-Komponente fest (der Speicherort der FLV-Datei und der Speicherort der XML-Datei).
- Veröffentlichen Sie die Datei.

Im folgenden Beispiel wird XML 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 mehrsprachige Videos mit eingebetteten Cue-Points. Weitere Informationen finden Sie unter „[Unterstützung mehrerer Sprachspuren mit eingebetteten Cue-Points](#)“ auf Seite 278.

Unterstützung mehrerer Sprachspuren mit eingebetteten Cue-Points

Die Eigenschaft `track` der FLVPlaybackCaptioning unterstützt mehrere Sprachspuren mit eingebetteten Cue-Points, sofern die Timed Text-XML-Datei die FLVPlaybackCaptioning-Standards für Cue-Points einhält. (Weitere Informationen hierzu finden Sie unter „[Cue-Point-Standards für die FLVPlaybackCaptioning-Komponente](#)“ auf Seite 275.) Die FLVPlaybackCaptioning-Komponente unterstützt jedoch keine mehrfachen Sprachspuren in separaten XML-Dateien. Um die Eigenschaft `track` zu verwenden, belegen Sie sie mit einem Wert, der nicht 0 ist. Wenn Sie zum Beispiel den Wert 1 einstellen (`track == 1`), durchsucht die FLVPlaybackCaptioning-Komponente die Cue-Point-Parameter. Wenn nichts gefunden wird, wird die Eigenschaft `text` in den Cue-Point-Parametern verwendet. Weitere Informationen finden Sie im Abschnitt über die `track`-Eigenschaft im *Komponenten-Referenzhandbuch für ActionScript 3.0*.

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 in der FLVPlayback-Komponente Untertitel verknüpfen. Weitere Informationen zum Öffnen mehrerer Video-Player finden Sie unter „[Verwenden mehrerer Video-Player](#)“ auf Seite 238. Um Untertitel in mehreren Video-Playern zu verwenden, erstellen Sie eine Instanz der FLVPlaybackCaptioning-Komponente für jeden `VideoPlayer` und legen Sie für `videoPlayerIndex` der FLVPlaybackCaptioning-Komponente den entsprechenden Index fest. Wenn nur ein `VideoPlayer` verwendet wird, hat der `VideoPlayer`-Index standardmäßig den Wert 0.

Im folgenden Beispiel werden einzelnen Videos einzelne Untertitel zu geordnet. Um dieses Beispiel nachzuvollziehen, müssen Sie die erfundenen URLs durch funktionierende URLs ersetzen.

```
captioner0.videoPlayerIndex = 0;
captioner0.captionURL = "http://www.[yourDomain].com/mytimedtext0.xml";
flvPlayback.play("http://www.[yourDomain].com/myvideo0.flv");
captioner1.videoPlayerIndex = 1;
captioner1.captionURL = "http://www.[yourDomain].com/mytimedtext1.xml";
flvPlayback.activeVideoIndex = 1;
flvPlayback.play ("http://www.[yourDomain].com/myvideo1.flv");
```

Anpassen der FLVPlaybackCaptioning-Komponente

Wenn die FLVPlaybackCaptioning-Komponente sofort einsetzen möchten, können Sie die Standardeinstellung verwenden, bei der die Untertitel direkt über der FLVPlayback-Komponente platziert werden. Sie können die FLVPlaybackCaptioning-Komponente anpassen, um die Untertitel anders zu positionieren.

Im folgenden Codebeispiel wird ein FLVPlayback-Objekt mit einer Schaltfläche zum Ein- und Ausschalten der Untertitel dynamisch erstellt.

So erstellen Sie ein FLVPlayback-Objekt mit Untertitel-Schaltfläche dynamisch

1. Platzieren Sie die FLVPlayback-Komponente bei den Koordinaten 0,0 auf der Bühne und geben Sie ihr den Instanznamen `player`.
2. Platzieren Sie die FLVPlaybackCaptioning-Komponente bei den Koordinaten 0,0 auf der Bühne und geben Sie ihr den Instanznamen `captioning`.
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-Videoexample, `caption_video.flv` gesetzt. Ändern Sie diese Variable zum Pfad der Captioning-Video-Komponente, der Sie eine CaptionButton-Komponente hinzufügen.

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. Fügen Sie die `FLVPlayback`- und die `FLVPlaybackCaptioning`-Komponente der Bibliothek hinzu.
7. Speichern Sie den folgenden Code als **`FLVPlaybackCaptioningExample.as`** in demselben Verzeichnis wie die FLA-Datei.
8. 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;
        }
    }
}
```

Weitere Informationen zu allen `FLVPlaybackCaptioning`-Parametern finden Sie im [*Komponenten-Referenzhandbuch für ActionScript 3.0*](#).

Timed Text-Tags

Die FLVPlaybackCaptioning-Komponente unterstützt Timed Text-Tags Captioning-XML-Dateien. Weitere Informationen über die Timed Text-Tags für Audio und Video finden Sie unter <http://www.w3.org>. In der folgenden Tabelle sind die unterstützten und nicht unterstützten Tags aufgeführt.

Funktion	Tag/Wert	Verwendung/ Beschreibung	Beispiel
Ignorierte Tags	metadata	Ignoriert / zulässig auf allen Dokumentebenen	
	set	Ignoriert / zulässig auf allen Dokumentebenen	
	xml:lang	Ignoriert	
	xml:space	Ignoriert / Verhalten wird geändert zu xml:space="default"	
	layout	Ignoriert / einschließlich aller region-Tags in einem layout- Tag-Abschnitt	
	br-Tag	Alle Attribute und Inhalte werden ignoriert.	

Funktion	Tag/Wert	Verwendung/ Beschreibung	Beispiel
Medien-Timing für Captions	begin	Nur in p-Tags zulässig.	<p begin="3s">
	attributes	Erforderlich für die Medienzeitbereitstellung von Captions.	
	dur attributes	Nur in p-Tags zulässig. Empfohlen. Falls nicht verwendet, wird die Caption mit der FLV-Datei beendet oder bei Beginn der nächsten Caption ausgeblendet.	
Uhr-Timing für Captions	end attributes	Nur in p-Tags zulässig. Empfohlen. Falls nicht verwendet, wird die Caption mit der FLV-Datei beendet oder bei Beginn der nächsten Caption ausgeblendet.	
	00:03:00.1	Vollständiges Uhrformat	
	03:00.1	Teilweises Uhrformat	
	10	Offset-Zeiten ohne Einheiten. Der Offset stellt Sekunden dar.	
	00:03:00:05	Nicht unterstützt.	
	00:03:00:05.1	Zeitformate, die Bilder oder	
	30f 30t	Ticks nicht unterstützt.	
Body-Tag	Body	Erforderlich / Unterstützung für nur ein body-Tag	<body> ... </body>

Funktion	Tag/Wert	Verwendung/ Beschreibung	Beispiel
Content-Tag	div-Tag	Null oder höher zulässig. Das erste Tag wird verwendet.	
	p-Tag	Null oder höher zulässig	
	span-Tag	Ein logischer Container für eine Abfolge von Einheiten mit Textinhalt. Keine Unterstützung für verschachtelte Bereiche (nested spans). Unterstützung für Attributstil-Tags.	
	br-Tag	Gibt einen ausdrücklichen Zeilenumbruch an.	
Stil-Tags (Alle style-Tags werden innerhalb des p-Tags verwendet)	style	Verweis auf ein oder mehrere Stilelemente. Kann als Tag und als Attribut verwendet werden. Als Tag wird ein ID-Attribut benötigt (der Stil kann im Dokument wiederverwendet werden). Unterstützung für ein oder mehrere style-Tags innerhalb von style-Tags.	
	tts:background Color	Legt eine style-Eigenschaft fest, die die Hintergrundfarbe eines Bereichs definiert. Alpha wird ignoriert, außer mit der Einstellung auf Null (alpha 0), welche den Hintergrund transparent definiert. Das Farbformat ist #RRGGBBAA	

Funktion	Tag/Wert	Verwendung/ Beschreibung	Beispiel
	tts:color	Legt eine style-Eigenschaft fest, die die Vordergrundfarbe definiert. Alpha wird für keine Farbe unterstützt. Der Wert transparent wird als schwarz übersetzt.	<pre><style id="3" style="2" tts:backgroundColor="white"/> "transparent" = #00000000 "black" = #000000FF "silver" = #C0C0C0FF "grey" = #808080FF "white" = #FFFFFFF "maroon" = #800000FF "red" = #FF0000FF "purple" = #800080FF "fuchsia"("magenta") = #FF00FFFF "green" = #008000FF "lime" = #00FF00FF "olive" = #808000FF "yellow" = #FFFF00FF "navy" = #000080FF "blue" = #0000FFFF "teal" = #008080FF "aqua"("cyan") = #00FFFFFF</pre>
	tts:fontFamily	Legt eine style-Eigenschaft fest, die die Schriftartfamilie definiert.	<pre>"default" = _serif "monospace" = _typewriter "sansSerif" = _sans "serif" = _serif "monospaceSansSerif" = _typewriter "monospaceSerif" = _typewriter "proportionalSansSerif" = _sans</pre>

Funktion	Tag/Wert	Verwendung/ Beschreibung	Beispiel
	tts:fontSize	Legt eine style-Eigenschaft fest, die die Schriftgröße definiert. Bei Angabe von zwei Werten wird nur der erste (vertikale) Wert verwendet. Prozentwerte und Einheiten werden ignoriert. Unterstützung für absolute Pixelgrößen (zum Beispiel 12) und relative Stilgrößen (zum Beispiel +2)	
	tts: fontStyle	Legt eine style-Eigenschaft fest, die den Schriftstil definiert.	"normal" "italic" "inherit"* * Das Standardverhalten; übernimmt den Stil vom umschließenden Tag.
	tts: fontWeight	Legt eine style-Eigenschaft fest, die die Schriftstärke definiert.	"normal" "bold" "inherit"* * Das Standardverhalten; übernimmt den Stil vom umschließenden Tag.
	tts: textAlign	Legt eine style-Eigenschaft fest, die festlegt, wie Inline-Bereiche in einem sie enthaltenden Block-Bereich ausgerichtet sind.	"left" "right" "center" "start" ("left") "end" ("right") "inherit"* *Übernimmt den Stil vom umschließenden Tag. Wenn kein textAlign-Tag gesetzt wird, ist die Standardeinstellung "left".

Funktion	Tag/Wert	Verwendung/ Beschreibung	Beispiel
	tts: wrapOption	Legt eine style-Eigenschaft fest, die definiert, ob der automatische Zeilenumbruch für das betreffende Element angewendet wird oder nicht. Diese Einstellung betrifft alle Absätze im Caption-Element.	"wrap" "noWrap" "inherit"* *Übernimmt den Stil vom umschließenden Tag. Wenn kein wrapOption-Tag gesetzt wird, ist die Standardeinstellung "wrap".
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		

Index

A

- ActionScript
 - FLVPlayback-Komponente hinzufügen 223
 - Komponenten hinzufügen 23
 - Komponenten-API 40
- ActionScript erstellen
 - Button 87
 - CheckBox 91
 - ColorPicker 94
 - ComboBox 99
 - DataGrid 106
 - DataProvider 63
 - Label 110
 - List 116
 - NumericStepper 119
 - ProgressBar 127
 - RadioButton 131
 - ScrollPane 135
 - Slider 139
 - TextArea 143
 - TextInput 148
 - TileList 152
 - UILoader 154
 - UIScrollBar 157
- ActionScript-Cue-Points
 - deaktivieren und aktivieren 232
 - FLVPlayback 230
 - löschen 232
 - zuweisen 232
- addChild()-Methode 23, 55
- addChildAt()-Methode 55
- addEventListener()-Methode in der
 - Ereignisverarbeitung 26
- addItem()-Methode 65

- addItem()-Methode 67
- addItemAt()-Methode 67
- Anpassen, Info 160
- Anzeigeliste
 - Komponenten entfernen 56
 - Komponenten verschieben 56
 - Objekte hinzufügen 55
- Anzeigeliste verwenden 54
- API, Komponenten 15
- Architektur, Komponenten 37
- Ausführen, Beispiele 35
- Authoring, Komponenten hinzufügen 21

B

- Barrierefreiheit, Komponenten 80
- Bearbeitbare Zelle 79
- Beispiele ausführen 35
- Benutzerdefinierte Komponenten 15
- Bibliothek
 - Bibliothek (Bedienfeld) 48
 - kompilierte Clips 48
- Bildschirmleseprogramme 80
- Button-Komponente
 - anpassen 170
 - erstellen 85
 - Interaktion 84
 - mit ActionScript erstellen 87
 - Parameter 85
 - Skins verwenden 172
 - Stile verwenden 171
 - verwenden 83

C

CellRenderer

- Bibliothekssymbol verwenden 75
- Bild verwenden 80
- Eigenschaften 78
- für bearbeitbare Zellen 79
- ICellRenderer implementieren 74
- Movieclip verwenden 80
- SWF verwenden 80
- verwenden 71
- Zellen formatieren 71

CheckBox-Komponente

- anpassen 174
- erstellen 90
- Interaktion 88
- mit ActionScript erstellen 91
- Parameter 89
- Skins verwenden 175
- Stile verwenden 174
- verwenden 88

ColorPicker-Komponente

- anpassen 176
- erstellen 94
- in Greetings-Anwendung 28
- in Greetings2-Anwendung 31
- Interaktion 93
- mit ActionScript erstellen 94
- Parameter 93
- Skins verwenden 177
- Stile verwenden 176
- verwenden 92

ComboBox-Komponente

- anpassen 178
- erstellen 98
- in Greetings-Anwendung 29
- in Greetings2-Anwendung 31
- Interaktion 96
- mit ActionScript erstellen 99
- Parameter 98
- Skins verwenden 180
- Stile verwenden 179
- verwenden 96

Components

- Architektur 37
- Skins, FLA-basiert 38

Cue-Points, FLVPlayback 229

- abfragen 234
- eingebettete aktivieren und deaktivieren 236
- entfernen 237
- Flash Video Cue-Points, Dialogfeld 231
- Navigation, suchen 235
- suchen 234
- verwenden 229

D

DataGrid-Komponente

- anpassen 181
- erstellen 103
- Interaktion 101
- mit ActionScript erstellen 106
- Parameter 103
- Skins verwenden 185
- Stile verwenden 181
- verwenden 100
- XML-Objekt 106

DataGrid-Objekte, CellRenderer anwenden 79

DataProvider

- Array verwenden 64
- bearbeiten 67
- Datenfeld anzeigen 65
- Elemente entfernen 67
- erstellen 61
- Liste mit Daten versehen 114
- merge() 69
- mit ActionScript erstellen 63
- sort() 69
- sortOn() 69
- verwenden 61

dataProvider, Parameter 61

DataProvider-Objekte, XML verwenden 66

Debugging, Komponentenanwendungen 44

defaultPushButton, Eigenschaft 59

Dokumentation

- Adobe Developer Center und Adobe Design Center 13

Info 12

Terminologieleitfaden 13

Download von Komponenten, Adobe Exchange 20

E

Eigenschaften

CellRenderer 78

festlegen 47

Eigenschafteninspektor 22

Einfache Anwendung 27

Ereignisse

Ereignisobjekt 53

und Listener 52

verarbeiten 52

Ereignisverarbeitung

addEventListener()-Methode 26

Unterschiede zu ActionScript 2.0 26

Externe Klassendatei 31

F

Festlegen, Texteeigenschaften 163

fl/accessibility/package-detail.html 81

FLA-basierte Komponenten 38

Flash Media Service 241

Flash Video Cue-Points, Dialogfeld 231

FLV-Dateien

abspielen 217

mehrere abspielen 237

Optionen 226

wechseln 239

FLVPlayback-Komponente

anpassen 242

Anwendungen erstellen 220, 270

Beschreibung 217

Cue-Points verwenden 229

FLV-Dateien streamen 241

Komponentenparameter 272

mehrere FLV-Dateien abspielen 237

mit ActionScript hinzufügen 223

mit dem Videoimportassistenten hinzufügen 222

Parameter 224

Skins erstellen 253

SMIL-Datei verwenden 260

source-Parameter festlegen 224

Version feststellen 25

verwenden 218

Video-Player verwenden 238

vordefinierte Skins 243

FLVPlaybackCaptioning-Komponente

Cue-Point-Standards 275

eingebettete Ereignis-Cue-Points für Untertitel

verwenden 276

mit ActionScript hinzufügen 271

Timed Text-Untertitel verwenden 273

Unterstützung mehrsprachiger Titel 278

Version feststellen 25

verwenden 270

vom Bedienfeld „Komponenten“ hinzufügen 270

FocusManager verwenden 57

Formatieren, Zellen 71

G

getChild()-Methode 55

getChildAt()-Methode 55

getChildByName()-Methode 55

getStyle()-Methode 162

Greetings (Anwendung)

ColorPicker 28

ComboBox 29

in einer FLA-Datei erstellen 28

in externer Klassendatei 31

RadioButton 28

TextArea 28

Größe von Komponenten ändern 50

I

ICellRenderer-Schnittstelle implementieren 74

In 273

Installieren von Komponenten 18, 20

Instanzen, Stile festlegen 162

K

Klassen und Komponenten, Vererbung 40

Klassenpfad 42

Kompilierte Clips im Bedienfeld „Bibliothek“ 48

Kompilierung

und eingebettete SWC 39

von Movieclips 39

Komponenten

- ActionScript-API 15
 - anzeigen 19
 - beim Authoring hinzufügen 21
 - benutzerbasierte Ordner 41
 - benutzerdefiniert 15
 - Dateien ändern 43
 - Debugging 44
 - Eigenschaften festlegen 47
 - eine einfache Anwendung 27
 - eingebettete SWC 39
 - Ereignisse verarbeiten 52
 - FLA-basiert 38
 - Größe ändern 50
 - herunterladen 20
 - hinzufügen und löschen 21
 - Info 15
 - installieren 18, 20
 - Klassenpfad 42
 - löschen 24
 - Merkmale 17
 - mit ActionScript hinzufügen 23
 - mit Eingabehilfen versehen 80
 - neu laden 43
 - Ordnerspeicherort 41
 - Parameter festlegen 22
 - Quelldateien, Speicherort 42
 - Skins erstellen 168
 - Stile festlegen 163
 - Stile für alle Komponenten festlegen 164
 - SWC-basiert 39
 - Tiefe im Container 54
 - UI-Typen 18
 - und addChild()-Methode 23
 - und Anzeigelimpe 54
 - und Live-Vorschau 51
 - Vererbung 40
 - Vorteile 16
 - zu einem Dokument hinzufügen 21
 - zur Laufzeit hinzufügen 23
 - Siehe auch: Namen der einzelnen Komponenten*
- ### Komponenten (Bedienfeld) 19
- Komponenteninstanzen
 - Stile abrufen 162
 - Stile festlegen 162
 - Stile für alle Instanzen festlegen 163
- ### Komponentenparameter
- anzeigen 45
 - festlegen 45
 - Siehe auch: Namen der einzelnen Komponenten*

L

Label-Komponente

- anpassen 187
- erstellen 109
- Interaktion 108
- mit ActionScript erstellen 110
- Parameter 108
- Skins verwenden 188
- Stile verwenden 188
- verwenden 108

List-basierte Komponenten

- und CellRenderer 60
- und Datenprovider 60
- und Zellen 60
- verwenden 59

List-Komponente

- anpassen 189
- Datenprovider verwenden 114
- erstellen 113
- Interaktion 111
- Interaktion mit einem Movieclip 115
- mit ActionScript erstellen 116
- Parameter 113
- Skins verwenden 190
- Stile verwenden 189
- verwenden 111

Listener, Ereignis 52

Live-Vorschau 51

Löschen, Komponenten 24

M

Movieclip

- kompilieren 39

N

Neu laden, Komponenten 43

numChildren (Eigenschaft) 55, 56

NumericStepper-Komponente

- anpassen 192
- Anwendungen erstellen 119
- Interaktion 117
- mit ActionScript erstellen 119
- Parameter 118
- Skins verwenden 194
- Stile verwenden 193
- verwenden 117

O

on(event) 26
Ordner „Component Assets“ 49

P

Pakete 40
Parameter
 Button-Komponente 85
 CheckBox-Komponente 89
 ColorPicker-Komponente 93
 ComboBox-Komponente 98
 DataGrid-Komponente 103
 eingeben 46
 FLVPlayback-Komponente 224
 Label-Komponente 108
 List-Komponente 113
 NumericStepper-Komponente 118
 ProgressBar-Komponente 122
 RadioButton-Komponente 130
 ScrollPane-Komponente 134
 Slider-Komponente 138
 TextArea-Komponente 142
 TextInput-Komponente 145
 TileList-Komponente 150
 UILoader-Komponente 153
 UIScrollBar-Komponente 155
ProgressBar-Komponente
 anpassen 195
 Anwendungen erstellen 123
 erstellen 123
 im Abfragemodus 124
 im Ereignismodus 123
 im manuellen Modus 126
 Interaktion 122
 mit ActionScript erstellen 127
 Parameter 122
 Skins verwenden 196
 Stile verwenden 195
 verwenden 121

Q

Quelldateien
 ändern 43
 und Klassenpfad 42
Quelldateien, Speicherort 42

R

RadioButton-Komponente
 anpassen 197
 Anwendungen erstellen 130
 erstellen 130
 in Greetings-Anwendung 28
 in Greetings2-Anwendung 31
 Interaktion 129
 mit ActionScript erstellen 131
 Parameter 130
 Skins verwenden 199
 Stile verwenden 198
 verwenden 129
Ressourcen, weitere von Adobe 13

S

ScrollPane-Komponente
 anpassen 200
 Anwendungen erstellen 135
 erstellen 135
 Interaktion 134
 mit ActionScript erstellen 135
 Parameter 134
 Skins verwenden 202
 Stile verwenden 201
 verwenden 133
setFocus()-Methode 58
setSize()-Methode 50
Skinning
 Button-Komponente 172
 CheckBox-Komponente 175
 ColorPicker-Komponente 177
 ComboBox-Komponente 180
 DataGrid-Komponente 185
 Definition 160
 FLVPlayback-Komponente 220
 Label-Komponente 188
 List-Komponente 190
 NumericStepper-Komponente 194
 ProgressBar-Komponente 196
 RadioButton-Komponente 199
 ScrollPane-Komponente 202
 Slider-Komponente 203
 TextArea-Komponente 206
 TextInput-Komponente 209
 TileList-Komponente 211
 UIScrollBar-Komponente 214

Skins

- Definition 166
- erstellen 169
- in der Bibliothek 49
- Informationen 165
- vordefinierte, FLVPlayback 243
- Zugriff auf der Bühne 166
- Zugriff in der Bibliothek 167

Slider-Komponente

- anpassen 202
- erstellen 138
- Interaktion 137
- mit ActionScript erstellen 139
- Parameter 138
- Skins verwenden 203
- Stile verwenden 202
- verwenden 136

SMIL-Datei, Speicherort angeben 225

Stile

- Einstellungen 161
- festlegen 160
- für eine Komponenteninstanz festlegen 162
- Standardstile abrufen 162

Stile für alle Komponenten festlegen 164

Stile, verwenden für

- Button 171
- CheckBox 174
- ColorPicker 176
- ComboBox 179
- DataGrid 181
- Label 188
- List 189
- NumericStepper 193
- ProgressBar 195
- RadioButton 198
- ScrollPane 201
- Slider 202
- TextArea 205
- TextInput 208
- TileList 210
- UIScrollBar 213

SWC

- Komponenten als 39
- und FLVPlayback 39
- und FLVPlaybackCaptioning 39
- zum Exportieren 40

SWC in FLA-basierten Komponenten 39

Systemanforderungen für Komponenten 12

T

Terminologie in der Dokumentation 13

TextArea-Komponente

- anpassen 204
- erstellen 142
- in Greetings-Anwendung 28
- in Greetings2-Anwendung 31
- Interaktion 141
- mit ActionScript erstellen 143
- Parameter 142
- Skins verwenden 206
- Stile verwenden 205
- verwenden 140

TextFormat, Texteigenschaften festlegen 163

TextInput-Komponente

- anpassen 207
- erstellen 146
- Interaktion 145
- mit ActionScript erstellen 148
- Parameter 145
- Skins verwenden 209
- Stile verwenden 208
- verwenden 144

TileList-Komponente

- anpassen 210
- erstellen 151
- Interaktion 149
- mit ActionScript erstellen 152
- Parameter 150
- Skins verwenden 211
- Stile verwenden 210
- verwenden 149

Timed Text-Tags 281

Typografische Konventionen 13

U

UI-Komponenten

- Arten 18
- Version feststellen 24

UIComponent-Klasse und Komponenten, Vererbung 40

UILoader-Komponente

- anpassen 212
- erstellen 154
- Interaktion 153
- mit ActionScript erstellen 154
- Parameter 153
- verwenden 153

UIScrollBar-Komponente

- anpassen 213
- erstellen 156
- Interaktion 155
- mit ActionScript erstellen 157
- Parameter 155
- Skins verwenden 214
- Stile verwenden 213
- verwenden 155

V

- Verarbeiten, Ereignisse 52
- Vererbung in Komponenten 40
- Version
 - von FLVPlayback feststellen 25
 - von FLVPlaybackCaptioning feststellen 25
 - von UI-Komponenten feststellen 24
- Video-Player verwenden 238
- Videoimportassistent 222
- Vorteile 16

Z

Zellen

- bearbeitbar, CellRenderer 79
 - in List-basierten Komponenten 60
- Zielpublikum 12
 - Zugriff auf Standardstile 162

