

Praxiswissen

Flex 3

- **Moderne Rich-Internet-Anwendungen entwickeln**
- **Berücksichtigt Flex Builder 3 und Flex 3 SDK**
- **Behandelt Adobe AIR**



O'REILLY®

Gerald Reinhardt

Einleitung	VII
1 Grundlagen von Adobe Flex 3	1
Einsatzbereiche von Flex-Anwendungen	2
Adobe Flash und Flex	5
Das Flex-Framework	7
Eine Flex-Anwendung entwickeln	9
MXML	12
ActionScript 3	14
Adobe Flex Builder 3 oder Adobe Flex SDK	15
2 Adobe Flex Builder 3	17
Den Flex Builder installieren	18
Den Flex Builder starten	21
Ein Flex-Projekt anlegen	23
Aufbau des Flex Builder	25
Die erste Anwendung	29
Fazit	34
3 Adobe Flex 3 SDK	35
Das Software Development Kit	35
Das Flex 3 SDK installieren	36
Auf der Kommandozeile arbeiten	37
Kompilieren mit dem Flex 3 SDK	39
Erweitertes Arbeiten mit dem SDK	42
Fazit	46
4 MXML	47
Das MXML-Prinzip	47
Allgemeine Syntax	48

	MXML in der Anwendungsentwicklung	52
	Änderungen im MXML-Code	54
	Fazit	57
5	ActionScript 3	59
	ActionScript-Code anlegen	59
	Funktionen	66
	Variablen	70
	ActionScript 3-Klassen	75
6	Eine Flex-Anwendung gestalten	93
	Grundlagen des Layouts	93
	Layoutcontainer	98
	Textinhalte	111
	Kleine Hilfstexte	115
	Bilder integrieren	116
	Mit Cascading Style Sheets arbeiten	120
	Optische Zustände	125
	CSS-Stildefinitionen und Skins visuell ändern	134
7	Mit externen Daten arbeiten	139
	Variablen und Arrays	139
	XML-Daten einsetzen	147
	Listenartige Inhalte darstellen	157
8	Inhalte steuern	173
	Eine Flex-Anwendung steuern	174
	Optische Zustände	177
	Mit der Sichtbarkeit von Komponenten arbeiten	178
	Unterschiedliche Inhalte laden	183
	Steuerungselemente	190
	Layoutkomponenten	202
	Ereignisse und Aktionen	206
9	Navigationskomponenten einsetzen	223
	Aufbau von Navigationskomponenten	224
	ViewStack	227
	TabNavigator	233
	Accordion	238
	Effekte und Animationen	241

10 Eigene Komponenten erstellen	245
Komponenten mit MXML erstellen	246
Komponenten mit ActionScript erstellen	254
11 Medien einsetzen	275
Fotos und Grafiken	276
Mit Filtern arbeiten	279
Weitere Effekte	281
Flash-Filme	283
Sounds	288
Videos	297
12 Diagramme	305
Diagrammtypen	305
Adobe Flex Charting	309
Liniendiagramme	312
Kreisdiagramme	316
Säulen- und Balkendiagramme	321
13 Formulare	333
Aufbau eines Formulars	334
Ein Formular anlegen	335
Ein Formular gestalten	337
Eingaben prüfen	351
Formularinhalte verarbeiten	362
14 Datenaustausch mit einem Webserver	369
Externe Inhalte verwalten und bereitstellen	369
HTTPService	377
Die Flex-Anwendung mit dem Webserver verbinden	382
Datenbankinhalte	384
Daten zum Server übertragen	395
Inhalte auf dem Server verarbeiten	400
Die Flex-Anwendung auf einem Webserver installieren	408
15 Debugging	413
Das Problems-Bedienfeld	414
Fehler bei der Arbeit mit externen Daten	415
Die Anwendung zum Prüfen anhalten	417
Auf Fehler reagieren	423

16 Adobe Integrated Runtime (AIR)	425
Was ist AIR?	425
AIR installieren	427
Eine AIR-Anwendung erstellen	428
Eine AIR-Installationsdatei erstellen	432
Das Fenster der AIR-Anwendung konfigurieren	434
Index	441

Eine Flex-Anwendung gestalten

In diesem Kapitel:

- Grundlagen des Layouts
- Layoutcontainer
- Textinhalte
- Kleine Hilfstexte
- Bilder integrieren
- Mit Cascading Style Sheets arbeiten
- Optische Zustände
- CSS-Stildefinitionen und Skins visuell ändern

Einen umfangreichen Ausflug in das Thema ActionScript haben wir nun bewältigt, und nach jeder Anstrengung sollte es eine kleine Phase der Erholung geben. In diesem Kapitel gehen wir es wieder ein wenig ruhiger und vor allem visueller an. Auf den folgenden Seiten beschäftigen wir uns intensiv mit der Gestaltung einer Anwendung. Ausgehend vom Standardlayout, das bei der Anlage einer neuen Flex-Anwendung automatisch zugewiesen wird, werden wir uns langsam vorarbeiten und zuerst das Grundlayout der Anwendung näher betrachten; dann werden Sie die Komponenten des Flex-Frameworks einsetzen und manipulieren lernen, Inhalte in die Anwendung integrieren und ihre Eigenschaften beeinflussen, und zu guter Letzt mit sogenannten »optischen Zuständen« eine interessante Form der Darstellung unterschiedlicher Layouts einsetzen.

Grundlagen des Layouts

Den Einstieg in die Gestaltungsmöglichkeiten einer Flex-Anwendung soll das `<Application>`-Tag bilden, denn in diesem Tag können Einstellungen vorgenommen werden, die sich auf die komplette Anwendung auswirken. Das `<Application>`-Tag wird automatisch im Code des MXML-Dokumentes angelegt, wenn Sie mit dem Flex Builder eine neue Anwendung anlegen.



Auch wenn wir diesen Vorgang schon einmal durchgeführt haben, sollten Sie das jetzt noch einmal tun. Legen Sie dazu in einem Projekt nochmals eine neue Flex-Anwendung an, um die Erläuterungen nachvollziehen und selbst ausprobieren zu können. Die Aufgaben zum Galerie-Projekt sind davon unabhängig.

Betrachten wir nun dieses MXML-Dokument unmittelbar nach der Anlage der Datei näher, um auch die Bedeutung der vorgenommenen Einträge zu erkennen.

```
<?xml version="1.0" encoding="UTF-8">
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">

  </mx:Application>
```

Vor dem Einsatz der ersten MXML-Komponente steht eine Ihnen vielleicht schon vertraute Zeile, die hier zwar notwendig ist, aber nicht ausschließlich im Flex-Framework zum Einsatz kommt.

```
<?xml version="1.0" encoding="UTF-8">
```

Dabei handelt es sich um die Deklaration des Dokumentes, in der darauf hingewiesen wird, dass es sich im Folgenden um eine inhaltliche Strukturierung nach XML-Schema handelt und eine Zeichenkodierung im UTF-8 Format vorliegt. Eine andere Kodierung ist innerhalb des Flex-Frameworks nicht möglich. Die von uns üblicherweise benutzten Umlaute und Sonderzeichen werden innerhalb der Textangaben problemlos übernommen. Schauen wir uns den zweiten Eintrag an.

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">

  </mx:Application>
```

Auf den ersten Blick sollten Sie erkennen, dass hier ein öffnendes und ein schließendes Tag angelegt wurden. Zwischen beide Tags wird der spätere Inhalt der Anwendung in MXML-Code angelegt und verwaltet. Bei der Anlage der MXML-Application werden zwei Parameter angegeben. Der Hinweis auf das gespeicherte MXML-Schema `xmlns:mx="http://www.adobe.com/2006/mxml"` gilt an dieser Stelle für alle nutzbaren MXML-Tags, dafür steht das Kürzel `mx`. Im Laufe der folgenden Kapitel beziehe ich mich auf das Anlegen eigener Komponenten. Auch deren Quelle muss dann in der Application angegeben werden. Belassen wir es hier bei den `mx`-Komponenten.

Wesentlich interessanter und für das spätere Aussehen der Anwendung entscheidend sind die Angabe des Parameters `layout` und die Zuweisung des Wertes `absolute`. Dieser Wert ist standardmäßig auf `absolute` eingestellt, d.h., auch bei den Dokumenten, die Sie bereits erstellt haben, haben Sie mit dieser Einstellung gearbeitet, denn Sie haben an der Standardeinstellung ja nichts verändert.

Das Pop-up-Menü in Abbildung 6-1 zeigt, dass insgesamt drei Möglichkeiten für die Zuweisung eines Wertes für den `layout`-Parameter zur Verfügung stehen. Sie können die ausgewählte Variante für die Anwendung und später auch für einzeln eingesetzte Komponenten jederzeit ändern, indem Sie den Wert innerhalb des Source-Modus manuell ändern oder diesen Wert im Bedienfeld *Flex Properties* für die ausgewählten Komponente mithilfe des Pop-up neu ordnen, wenn Sie im Design-Modus arbeiten.

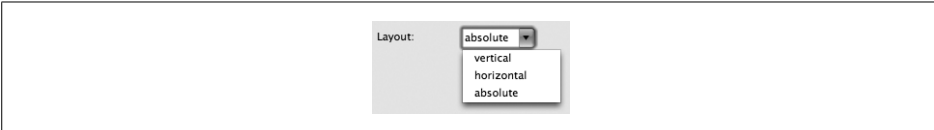


Abbildung 6-1: Das Layout-Pop-up beim Anlegen einer neuen Flex-Anwendung

Welche Wirkung hat nun der layout-Parameter? Besonders zu beachten ist die Tatsache, dass dieser Parameter sich eigentlich nicht auf die Komponente selbst auswirkt, sondern festlegt, in welcher Form die eingetragenen Inhalte angeordnet werden sollen. Tabelle 6-1 zeigt eine Übersicht über die drei Möglichkeiten, die Ihnen an dieser Stelle zur Verfügung stehen.

Tabelle 6-1: Die Einstellmöglichkeiten des Layout-Parameters

Layout	Beschreibung
vertical	Diese Einstellung bewirkt eine vertikale Anordnung aller eingefügten Komponenten. Dabei gilt die Reihenfolge der MXML-Tags, und die Angaben von x und y als Parameter haben keine Wirkung.
horizontal	Das horizontale Layout ist mit dem vertikalen Layout vergleichbar, die Anordnung erfolgt hier nicht unter-, sondern nebeneinander.
absolute	Dieser Wert legt fest, dass alle integrierten Komponenten und Inhalte auf die linke obere Ecke bezogen angeordnet werden. Dazu müssen die Angaben x für den Abstand von links und y für den Abstand von oben in den eingefügten Komponenten unbedingt angegeben werden.

Das Ergebnis dieser Einstellungen kann man erst erkennen, wenn man Inhalte in die Anwendung eingefügt hat. Beachten Sie in Abbildung 6-2 die Anordnung der Komponente Panel, die in das jeweilige Layout gezogen wurde, wobei keinerlei zusätzliche Angaben gemacht wurden.

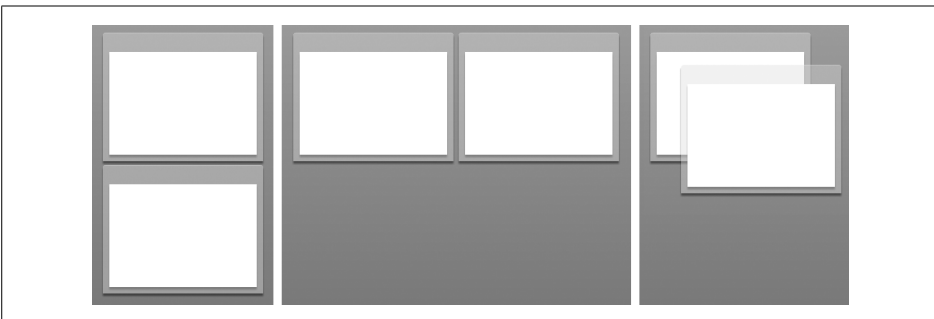


Abbildung 6-2: Die Anordnung der Komponenten vertical, horizontal und absolute

Da diese Eigenschaft Auswirkungen auf die eingebetteten Komponenten hat, kann die layout-Eigenschaft auch in allen anderen Flex-Komponenten zum Einsatz kommen, die als Layoutcontainer genutzt werden können. Sie werden diese Eigenschaften im Laufe dieses Kapitels noch mehrfach einsetzen.

Die Anwendung farblich gestalten

Bei der Anlage einer neuen Flex-Anwendung erhält diese den Standardstil zugewiesen, bei dem farbliche Festlegungen getroffen sind, die auch in allen anderen Komponenten zum Einsatz kommen, solange keine manuellen Änderungen vorgenommen werden. Als Standard ist für den Hintergrund ein Verlauf in einem grünlischen Farbton festgelegt. Zur farblichen Gestaltung der gesamten Anwendung stehen Ihnen zwei Eigenschaften zur Verfügung, die in Kombination das Erscheinungsbild ergeben: `backgroundGradientColors` und `backgroundGradientAlphas`. Tabelle 6-2 zeigt, was diese beiden Eigenschaften voneinander unterscheidet.

Tabelle 6-2: Die Eigenschaften `backgroundGradientColors` und `backgroundGradientAlphas`

Parameter	Beschreibung
<code>backgroundGradientColors</code>	Es handelt sich bei diesem Parameter um die Angabe von Farben für einen Farbverlauf. Die Angabe der Werte für die Farben erfolgt in eckigen Klammern und enthält die beiden hexadezimalen Farbwerte. Der erste Wert steht für die Farbe oben, der zweite Wert für die Farbe unten, getrennt werden die Angaben durch ein Komma, z.B. [#9CB0BA, #68808C]. Diese Angabe erzeugt übrigens den Standardfarbverlauf.
<code>backgroundGradientAlphas</code>	Bei diesem Parameter wird die Opazität (Lichtundurchlässigkeit) der eingesetzten Farbe angegeben. Dieser Wert kann zwischen 100% und 0% liegen. Auch dieser Wert wird in einer eckigen Klammer angegeben, wobei für 100% 1.0 eingetragen wird, für 50% entsprechend 0.5 usw., z.B. [1.0, 0.7].

Selbstverständlich lassen sich diese Einstellungen sowohl im Quellcode als auch visuell im Design-Modus des Flex Builder im Bereich *Fill* (Füllung) des Bedienfeldes *Flex Properties* (siehe Abbildung 6-3).

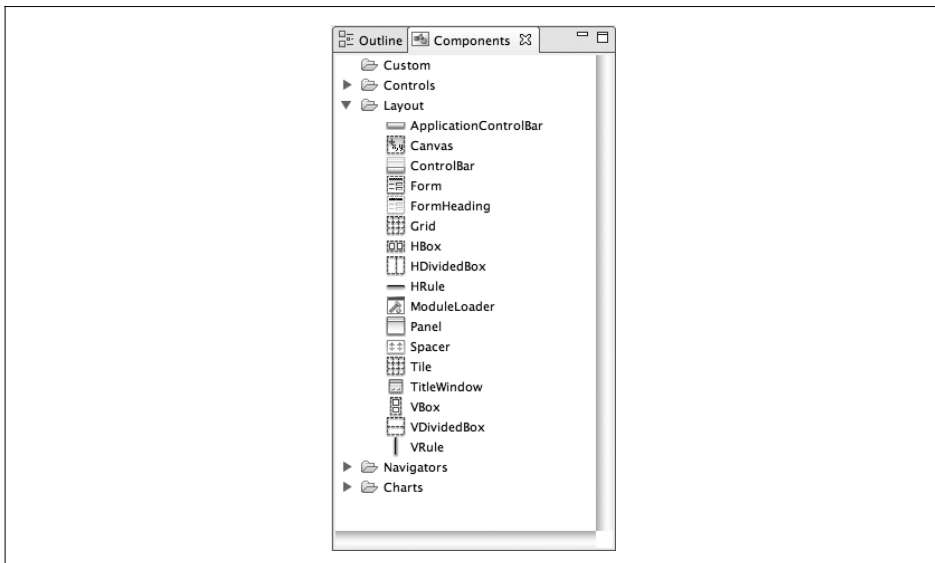


Abbildung 6-3: Der Bereich *Fill* im Bedienfeld *Flex Properties*

Bei den Einstellungen innerhalb des Bereichs *Fill* müssen Sie ein wenig umdenken. Die Füllung in der Anwendung selbst verläuft von oben nach unten, was beim Einsatz eines Verlaufs deutlich zu erkennen ist. Die Elemente zur Zuweisung der Farben und Transparenzen sind allerdings nebeneinander angeordnet. Mit den Elementen auf der linken Seite legen Sie Farbe und Transparenz für den oberen Teil der Anwendung fest. Die beiden rechten Elemente dienen zur Einstellung der Werte für den unteren Teil der Anwendung. Für den Quellcode ergibt sich an dieser Stelle für das Tag `mx:Application` folgende Ergänzung.

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"  
backgroundGradientAlphas="[1.0, 1.0]"  
backgroundGradientColors="[#9CB0BA, #68808C]" >
```



Die Hintergrundfarbe der Anwendung lässt sich durch eine Kombination aus Farben und Alphawerten festlegen. Schwarz (#000000) mit einem Alphawert von 50% ergibt z.B. ein Grau.

Ein Hintergrundbild einsetzen

Auch der Einsatz eines Hintergrundbildes ist in einer Flex-Anwendung möglich. Ein dafür existierender Parameter kann allerdings nur manuell im `mx:Application`-Tag eingefügt werden. Eine Auswahl in den Flex Properties steht nicht zur Verfügung. Mit dem Parameter `backgroundImage` können Sie ein Bild als Hintergrund zuweisen.

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"  
backgroundImage="bilderOrdner/DateiFuerDasHintergrundbild" >
```

Das Hintergrundbild wird dabei einmalig platziert und den Einstellungen entsprechend skaliert. Beachten Sie auch den Inhalt des einzusetzenden Bildes: Ein schlicht gehaltenes Bild ist besser geeignet als eines mit starken Kontrasten o.Ä., das Unruhe in das Layout bringen würde.

Maße und Position der Anwendung

Die schon beim Panel benutzen Angaben für die Abstände von links und oben sowie die Angaben für die Breite und Höhe einer Komponente können auch innerhalb der Anwendung eingesetzt werden, ohne dass ein Fehler angezeigt wird. Standardmäßig gibt es dafür innerhalb der Anwendung keine Angaben, da die gesamte Anwendung ins komplette Browserfenster ausgedehnt wird. Bei Angabe von Abmessungen erfolgt die Darstellung nur in der angegebenen Größe links oben im Browserfenster. Die Angaben der Abstände, als `x` und `y`, wirken sich auf die Anwendung nicht aus, sondern auf die erste einzubettende Komponente. Aus diesem Grund können Sie die Angaben für `width`, `height`, `x` und `y` getrost weglassen, außer Sie planen eine Anwendung mit speziellen Ausmaßen, die in eine andere Anwendung eingebettet werden soll.

Diese Angaben sollten für die Gestaltung der kompletten Anwendung vollkommen ausreichen; wenn Sie weitere Eigenschaften und ihre Auswirkungen ausprobieren wollen, können Sie das selbstverständlich tun.

Übung 1: Die Galerie FLEXibel anlegen

Damit Sie selbst Ihre ersten praktischen Erfahrungen machen können, sollten Sie sich nun dem Projekt zum Buch, der Galerie FLEXibel zuwenden. Die Dateien zum Buch finden Sie auf der Verlagswebsite unter http://examples.oreilly.de/german_examples/flex3basger/. Um die Galerie zunächst anzulegen, folgen Sie bitte dieser Anleitung:

- 1 Öffnen Sie die Datei *main.mxml* im Projekt Galerie FLEXibel und speichern Sie sie unter dem neuen Namen *ue_01.mxml*.
- 2 Wechseln Sie in den Design-Modus, falls Sie sich noch nicht darin befinden.
- 3 Wählen Sie die Anwendung (die gesamte Fläche) aus und achten Sie darauf, dass das Fenster *Flex Properties* offen ist.
- 4 Wählen Sie im Bereich *Fill* das linke Farbfeld und stellen Sie dort die Farbe Schwarz ein.
- 5 Wechseln Sie in den Source-Modus und ändern Sie die Angaben der Farben folgendermaßen:

```
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" backgroundGradientAlphas="[1.0, 1.0]" backgroundGradientColors="[#000000, #333333]">
```
- 6 Starten Sie die Anwendung und beachten Sie den Farbverlauf im Browser.
- 7 Kehren Sie in den Flex Builder zurück und ändern Sie im Design-Modus auch die zweite Farbe in schwarz. Starten Sie anschließend die Anwendung erneut. Beachten Sie die schwarze Hintergrundfarbe.
- 8 Stellen Sie nun im Design-Modus im Bereich *Fill* mit dem rechten Regler den Alphawert auf 50% ein. Dadurch ändern Sie die Farbintensität der Hintergrundfarbe im unteren Teil der Anwendung. Aus der Hintergrundfarbe Schwarz, die bisher eingestellt war, wird ein Verlauf von Schwarz zu Grau.

Layoutcontainer

Neben dem großen Container für die gesamte Anwendung gibt es eine Vielzahl anderer Komponenten im Flex-Framework, deren Aufgabe in erster Linie darin besteht, andere Komponenten aufzunehmen bzw. anzuordnen oder Inhalte in der Anwendung zu platzieren bzw. anzuzeigen. Unter Zuhilfenahme dieser Container entsteht innerhalb der erstellten Anwendung eine Struktur für den Aufbau des Inhalts, was die Anordnung der Inhalte erleichtert. Betrachten wir die im Flex-Framework enthaltenen Container erst einmal in einer Übersicht, bevor sie zum Einsatz kommen (siehe Tabelle 6-3).

Tabelle 6-3: Wichtige Container, die Flex von Hause aus mitbringt

Container	Beschreibung
<code>mx:Application</code>	Diese Komponente nimmt die kompletten Inhalte der Anwendung auf und bildet die Umhüllung der Anwendung. Bei der Erstellung einer Flex-Anwendung wird diese Komponente automatisch angelegt. Mithilfe der Application kann der Gesamteindruck der Anwendung gesteuert werden.

Tabelle 6-3: Wichtige Container, die Flex von Hause aus mitbringt (Fortsetzung)

Container	Beschreibung
mx:Canvas	Das ist eine Hülle zur Aufnahme weiterer Komponenten oder Inhalte. Diese Hülle bietet zum einen die Möglichkeit, in der Anwendung platziert zu werden, und zum anderen können die in ihr eingebetteten Inhalte wiederum individuell angeordnet werden. Die Komponente ist ohne Zuweisung entsprechender Eigenschaften nicht sichtbar. Die Canvas-Komponente kann Inhalte absolut positionieren (nur nicht die Anwendung selbst).
mx:VBox	Die vertikale Box dient zur Aufnahme von Komponenten und Inhalten, die innerhalb dieser Box ohne weitere Angaben untereinander angeordnet werden, vergleichbar mit einer zeilenartigen Darstellung innerhalb einer Tabelle. Auch diese Komponente ist innerhalb der Anwendung nur sichtbar, wenn entsprechende Eigenschaften zugewiesen werden.
mx:HBox	Bei der horizontalen Box handelt es sich um eine Komponente, die vergleichbar zur vertikalen Box eingesetzt wird, in der allerdings die Inhalte nebeneinander angeordnet werden, also in Form von Spalten.
mx:Panel	Diese Komponente ist eine spezielle Form von Container mit einem eigenen Aussehen, d.h. die Komponente wird in der Anwendung als Container auch sichtbar sein. Das Panel besteht aus einer gestalteten Fläche, in der auch ein Titel im Rahmen und Inhalt in der Fläche zugewiesen werden kann.
mx:TitleWindow	Vergleichbar mit der Komponente Panel – Unterschied: Das TitleWindow kann auch als Pop-up-Fenster eingesetzt werden.
mx:VDividedBox mx:HDividedBox	Diese beiden Boxen ähneln der VBox und der HBox. Die Aufteilung der Inhalte erfolgt hier allerdings boxfüllend, und es liegt eine Erweiterung vor, da zwischen den angeordneten Inhalten ein mit der Maus bewegbares Element angezeigt wird. Mit diesem Element lässt sich das Verhältnis der Breiten bei den horizontalen Boxen bzw. das Verhältnis der Höhen bei den vertikalen Boxen manipulieren.

Bisher haben Sie zwei dieser Containerkomponenten kennengelernt: Das Panel haben wurde mehrfach als Demonstrationsobjekt eingesetzt und die Application im vorigen Abschnitt intensiver betrachtet. Gehen wir nun am besten systematisch vor und betrachten wir das Zusammenspiel der Komponenten in einem strukturierten Layout.

Alle hier aufgelisteten Komponenten der Container für die Anordnung von Inhalten finden Sie bei aktivem Design-Modus des Flex Builder im *Components*-Bedienfeld (siehe Abbildung 6-4). Öffnen Sie die Kategorie *Layout*, falls sie geschlossen ist. Nehmen Sie zur Kenntnis, dass hier noch weitere Komponenten enthalten sind, die aber im Moment noch nicht zum Einsatz kommen.

Canvas als Hülle

Das Layout einer Anwendung hängt natürlich vom geplanten Design ab. An dieser Stelle werde ich zeigen, in welcher Form die Containerkomponenten eingesetzt werden, um Inhalte in der Anwendung anzuordnen. Am Ende werden Sie sehen, dass ein verschachteltes Konstrukt aus verschiedenen Komponenten besteht. Beginnen möchte ich dazu mit der Komponente Canvas, einer Hülle zur Aufnahme von Inhalten oder weiteren Hüllen und Containern. Innerhalb der Anwendung lässt sich eine Hülle anlegen, die später alles umschließen soll. Diese Hülle können Sie

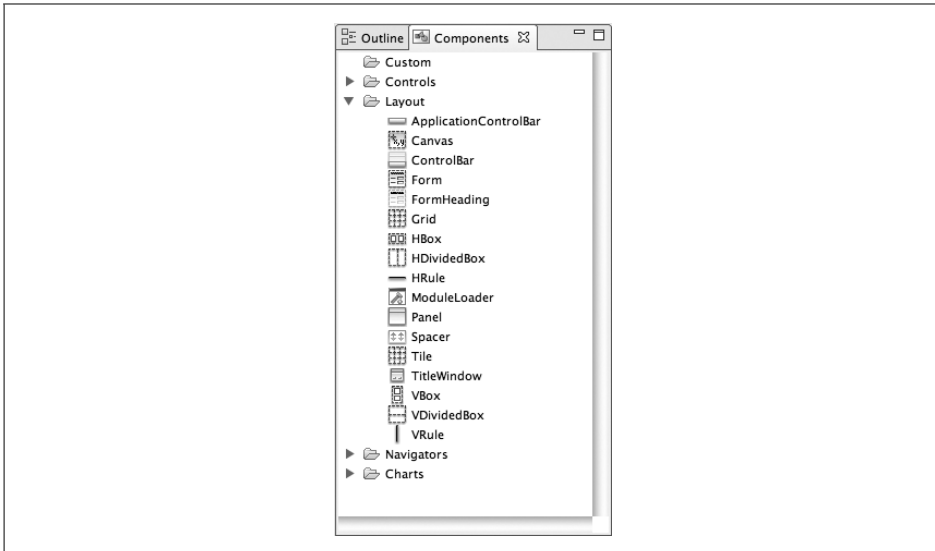


Abbildung 6-4: Die Kategorie Layout des Bedienfeldes Components mit allen Komponenten, die zum Layouten einer Flex-Anwendung eingesetzt werden können.

später noch mit Eigenschaften versehen. Im Design-Modus des Flex Builder reicht es aus, die Canvas-Komponente aus dem Bedienfeld in die Anwendung zu ziehen. Bewegen Sie sie mit der Maus ein wenig über die Anwendung, und Sie bekommen blaue Linien angezeigt, die als Hilfsmittel für die Ausrichtung der Komponente dienen. Benutzen Sie diese Linien zur Ausrichtung der Komponenten, werden diese automatisch mit entsprechenden Eigenschaften und Werten versehen. Nach dem Einfügen der Komponente sieht der erzeugte Quellcode in seiner einfachsten Form folgendermaßen aus.

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
  <mx:Canvas width="200" height="200">

    </mx:Canvas>
</mx:Application>
  
```

Je nachdem, wo Sie am Ende diese Komponente in der Anwendung abgelegt haben, gibt es noch Angaben zur Position, die in den folgenden Abschnitten erläutert werden, da es in der Entwicklung von Flex-Anwendung verschiedene Möglichkeiten der Positionierung gibt.

Feste Position der Hülle in der Anwendung

Bei der festen Positionierung der Komponenten wird die Position an einen bestimmten Ort festgesetzt und bleibt immer an dieser Stelle. Dazu werden die bereits benutzten Werte x für den Abstand von links und y für den Abstand von oben angegeben.

```
<mx:Canvas width="200" height="200" x="200" y="200">
</mx:Canvas>
```

Zentrierte Position in der Anwendung

In diesem Fall der Positionierung lässt sich die Anwendung im Browserfenster zentriert ausrichten. Im Design-Modus erscheint in der Mitte der Anwendung die blaue Linie. Im Quellcode gibt es dafür die zwei Parameter `verticalCenter` und `horizontalCenter`. Setzt man beide Werte auf 0, wird die Hülle exakt in der Mitte der Anwendung platziert.

```
<mx:Canvas width="200" height="200" verticalCenter="0" horizontalCenter="0">
</mx:Canvas>
```

Die Angaben für `verticalCenter` und `horizontalCenter` können auch benutzt werden, wenn nicht die Mitte als Positionierung gewählt werden soll. Dazu werden die Abweichungen durch die Angabe eines Pixelwertes vom Mittelpunkt aus angegeben. Das können entsprechend der Positionierung auch negative Werte sein.

```
<mx:Canvas width="200" height="200" verticalCenter="-200" horizontalCenter="190">
</mx:Canvas>
```

Constraints zur Positionierung einsetzen

Die Positionierung einer Komponente in der Anwendung oder in einem Panel- oder `TitleWindow`-Container lässt sich auch visuell manipulieren. Dazu dient der Bereich *Constraints* im Bedienfeld *Flex Properties*.

In diesem Bereich können Sie für jede Komponente einen Bezugspunkt für die Positionierung festlegen. Dazu müssen Sie nur bestimmen, auf welchen Rand die Positionsangaben der Komponente Bezug nehmen bzw. ob die Komponente mittig ausgerichtet werden soll. Horizontal kann der Bezug zum linken oder rechten Rand erfolgen oder ein Bezug zur Mitte hergestellt werden. Für die vertikale Ausrichtung betrifft das den oberen oder unteren Rand bzw. auch die Mitte der Anwendung. Die entsprechenden Abstände zur ausgewählten Ausrichtung werden in diesem Fenster ebenfalls mit angegeben (siehe Abbildung 6-5).

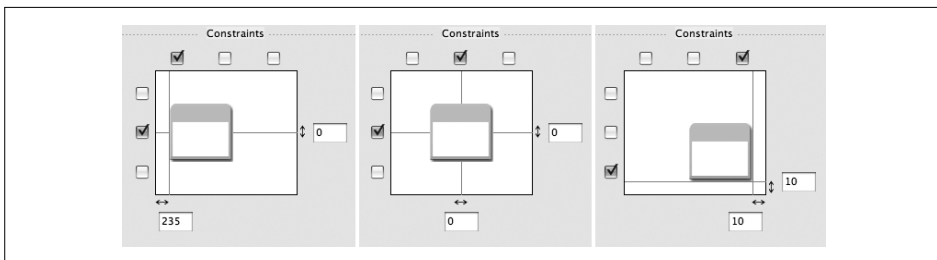


Abbildung 6-5: Der Bereich *Constraints* zur visuellen Platzierung von Komponenten

Im MXML-Code ergibt das unterschiedliche Positonsangaben für eine Komponente. Tabelle 6-4 zeigt eine Übersicht über diese Eigenschaften zur Positionierung.

Tabelle 6-4: Eigenschaften für die Positionierung

Bezugspunkt der Komponente	Eigenschaft zur Positionierung
linker Rand	x oder left
oberer Rand	y oder top
vertikale Mitte	verticalCenter
horizontale Mitte	horizontalCenter
rechts	right
unten	bottom



Bei der Positionierung sollten Sie sich für die horizontale und vertikale Ausrichtung im Hinblick auf einen Bezugspunkt entscheiden und auch nur diese Angabe in der Komponente einsetzen.

Nun kommt es bei der Komponente Canvas noch darauf an, ob diese ausschließlich zur Aufnahme und Positionierung der kommenden Inhalte vorgesehen ist, oder ob hier schon ein Element zur Gestaltung eingesetzt werden soll. Dazu bietet die Komponente noch weitere Eigenschaften, die ohne Angaben nicht berücksichtigt werden. So können Sie der Hülle auch eine Hintergrundfarbe, ein Hintergrundbild oder Transparenz in Form eines Alphawertes zuweisen.

```
<mx:Canvas width="200" height="200"
background-color="#741313"
alpha="0.4">

</mx:Canvas>
```

Sie können der Hülle auch einen Rahmen zuweisen, der diesem Container noch einen weiteren Touch verpassen kann, wenn eine geeignete Farbe zum Einsatz kommt und die Kanten unter Umständen abgerundet werden.

```
<mx:Canvas width="200" height="200"
background-color="#741313"
alpha="0.37"
cornerRadius="9"
borderStyle="solid"
borderColor="#FFFFFF">

</mx:Canvas>
```



Die Einstellungen für die Eigenschaften der Komponente Canvas können auch im Design-Modus vorgenommen werden. Achten Sie auf die Möglichkeiten im Bedienfeld Flex Properties. Dort finden Sie im Bereich Style eine Vielzahl von Einstellungsmöglichkeiten für Linien und Füllungen, und andere Angaben, die sich selbst erklären.

Nun können Sie innerhalb der existierenden Hülle weitere Umhüllungen einsetzen, um damit ein Gerüst für den späteren Inhalt aufzubauen (siehe Abbildung 6-6). Wer sich in HTML und CSS auskennt, wird eine Ähnlichkeit zum Box-Modell erkennen. Durch Einfügen und Anpassen lassen sich komplexe Layoutstrukturen aufbauen. Achten Sie aber auch auf die folgenden Abschnitte in diesem Kapitel, da es neben der Canvas-Komponente noch weitere Container gibt.

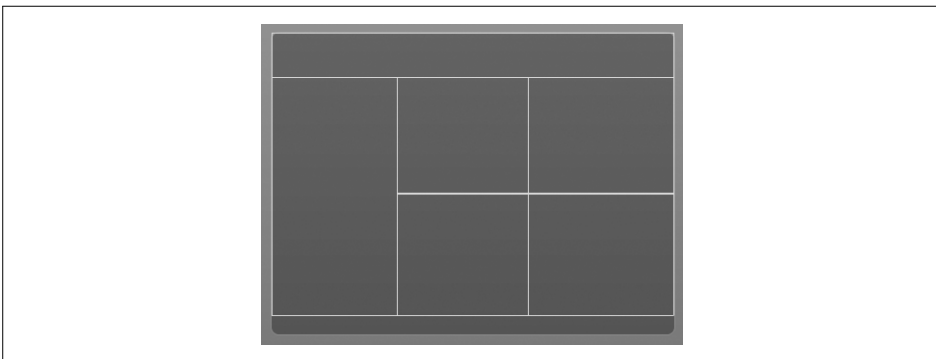


Abbildung 6-6: Eine Sammlung von Containern für das Layout einer Anwendung

Übung 2: Den Layoutcontainer der Anwendung anlegen

- 1 Öffnen Sie die Datei `ue_01.xml` und speichern Sie sie unter dem neuen Dateinamen `ue_02.xml`.
- 2 Aktivieren Sie den Design-Modus und öffnen Sie das *Components*-Bedienfeld und dort die Kategorie *Layout*.
- 3 Ziehen die Canvas-Komponente in die Anwendung und achten Sie darauf, dass Ihnen die blaue Linie der Zentrierung angezeigt wird. Lassen Sie die Maustaste los.
- 4 Stellen Sie nun im selben Bedienfeld im *Layout*-Bereich die Breite der Komponente auf 800 und die Höhe auf 600 Pixel.
- 5 Platzieren Sie die Umhüllung nun im Bereich *Constraints* horizontal auf *Zentriert* und vertikal auf *Mitte*, und die angegebenen Abstandswerte auf 0. Dadurch wird folgender Code erzeugt:

```
<mx:Canvas width="800" height="600" horizontalCenter="0" verticalCenter="0">
</mx:Canvas>
```
- 6 Wechseln Sie in den Source-Modus, um dort die Komponente Canvas manuell zu erzeugen.
- 7 Erzeugen Sie nun innerhalb der bereits erzeugten Komponente folgenden Quellcode:

```
<mx:Canvas width="800" height="90" left="0" top="0"
background-color="#FFFFFF">
</mx:Canvas>
```
- 8 Starten Sie die Anwendung.

HBox und VBox

Neben der absoluten Positionierung von Inhalten und Komponenten innerhalb der Anwendung können Inhalte auch *relativ* zueinander platziert werden. Dazu können Sie die Komponenten HBox (H steht für horizontal) und VBox (V steht für vertikal) verwenden. Beim Einsatz dieser Boxen werden die Inhalte oder auch weitere Container entweder horizontal oder vertikal angeordnet. Dabei werden keinerlei zusätzliche Angaben benötigt, da die Positionierung schon durch die Eigenschaften von HBox und VBox festgelegt ist.

```
<mx:VBox>
  <mx:Panel width="120" height="100" layout="absolute">

    </mx:Panel>
  <mx:Panel width="120" height="100" layout="absolute">

    </mx:Panel>
</mx:VBox>
<mx:HBox>
  <mx:Panel width="120" height="100" layout="absolute">

    </mx:Panel>
  <mx:Panel width="120" height="100" layout="absolute">

    </mx:Panel>
</mx:HBox>
```

Auch der Konstruktion des Layouts mit einer Verschachtelung von HBox- und VBox-Komponenten steht nichts im Wege, Sie müssen nur auf die Positionierung der einzelnen Boxen innerhalb des Quellcodes achten. Probieren Sie es bei Gelegenheit mal aus, indem Sie eine Aufteilung vornehmen und die Boxen ineinander platzieren.

Achten Sie bei der Arbeit innerhalb des Flex Builder auch auf die Anordnung aller eingesetzten Komponenten im Bedienfeld *Outline*. Hier können Sie die Reihenfolge der angelegten Komponenten sehen und beim Aufbau der Struktur die Hierarchie und die Verschachtelung der Komponenten auch visuell kontrollieren (siehe Abbildung 6-7).

Legen wir nun ein Konstrukt aus horizontalen und vertikalen Boxen an, um später die Inhalte in der Anwendung einfügen zu können.

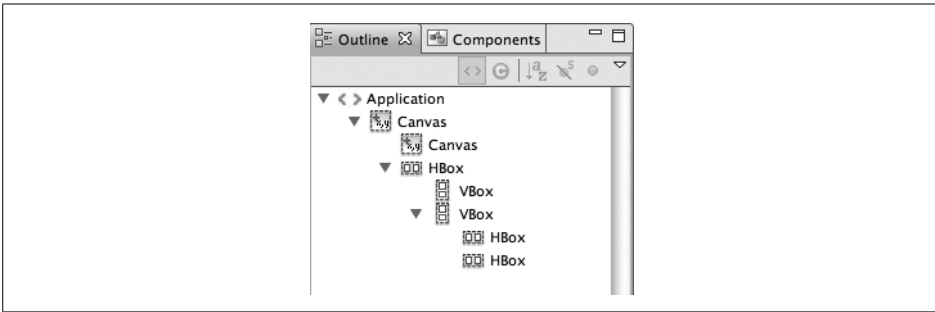


Abbildung 6-7: Das Bedienfeld Outline mit einer Übersicht der benutzten Komponenten

Übung 3: HBox- und VBox-Komponenten einsetzen

In dieser Übung soll das inhaltliche Grundgerüst für die Aufnahme der Inhalte der »Startseite« erstellt werden. Dafür werden die Komponenten HBox und VBox eingesetzt und ineinander verschachtelt, so dass nach Beendigung dieser Übung der Aufbau vorliegen sollte, der in Abbildung 6-8 zu sehen ist.

- 1 Öffnen Sie die Datei *ue_02.mxml* und speichern Sie sie unter dem Namen *ue_03.mxml*.
- 2 Aktivieren Sie den Design-Modus und das Bedienfeld *Components*.
- 3 Ziehen Sie die Komponente HBox in die Komponente Canvas unterhalb der weißen Komponente. Achten Sie vorerst nicht auf die Maße.
- 4 Ändern Sie nun im Bedienfeld *Flex Properties* die Eigenschaften folgendermaßen:
X: 0 Pixel, Breite: 800 Pixel, Y: 95 Pixel, Höhe: 505 Pixel. Das Ergebnis sollte folgender Quellcode sein:


```
<mx:Canvas width="800" height="600" horizontalCenter="0" verticalCenter="0">
  <mx:Canvas width="800" height="90" left="0" top="0"
    backgroundColor="#FFFFFF">
  </mx:Canvas>
  <mx:HBox x="0" width="800" y="95" height="505">
  </mx:HBox>
</mx:Canvas>
```
- 5 Wechseln Sie in den Source-Modus, um weitere Boxen zu erstellen. Legen Sie innerhalb der Komponente HBox zwei VBox-Komponenten an. Achten Sie dabei auf die unterschiedlichen Breitenangaben.


```
<mx:HBox x="0" width="800" y="95" height="505">
  <mx:VBox height="505" width="200">
  </mx:VBox>
  <mx:VBox height="505" width="590">
  </mx:VBox>
</mx:HBox>
```
- 6 Zum Abschluss soll nun noch in die rechte vertikale Box noch zwei horizontale eingebunden werden.


```
<mx:VBox height="505" width="590" x="0" y="0">
  <mx:HBox width="590" height="250">
  </mx:HBox>
  <mx:HBox width="590" height="245">
  </mx:HBox>
</mx:VBox>
```

Übung 3: HBox- und VBox-Komponenten einsetzen (Fortsetzung)

- 7 Speichern Sie das Dokument. Wenn Sie die Anwendung jetzt starten, dürfte keine Änderung zur vorherigen Übung sichtbar werden, da den Boxen bisher keine Eigenschaften zugewiesen wurden und der Inhalt noch fehlt. Darum kümmern wir uns in der nächsten Übung.

Achten Sie auch darauf, dass es verschiedene Möglichkeiten gibt, um die Struktur einer Anwendung einzurichten. Wundern Sie sich also nicht, wenn Sie beim Ausprobieren eines anderen Weges zum gleichen Ergebnis kommen. Nun können Sie natürlich auch direkt in diese bisher betrachteten Container Inhalte wie Text und Bilder einfügen. Dazu werde ich eine weitere Komponente ins Spiel bringen.



Abbildung 6-8: Das Konstrukt aus Canvas und Boxen

Panel oder TitleWindow

Mit der Panel-Komponente sind wir ja nun schon vertraut, da wir in den ersten Kapiteln immer wieder auf sie zurückgegriffen haben. Das hatte den einfachen Grund, dass mit wenigen Schritten ein sichtbares Element in eine Anwendung integriert und manipuliert werden konnte. Wie schon in der Übersicht zu Beginn dieses Abschnitts erwähnt, sind die beiden Komponenten Panel und TitleWindow quasi identisch; sie unterscheiden sich nur darin, dass das Panel statisch in die Anwendung eingebettet ist und das TitleWindow auch als Pop-up-Fenster funktionell eingesetzt werden kann.

Diese beiden Container sind auf der Basis einiger Elemente entstanden, die häufig in Anwendungen eingesetzt werden. Beide Komponenten enthalten diverse Eigenschaften, die in ihrer Kombination eine enorme Vielfalt von Designs zulassen. Schauen wir uns den Aufbau der Komponente in Abbildung 6-9 etwas genauer an.

Optisch bestehen diese Komponenten aus drei einzelnen Elementen:



Abbildung 6-9: Der typische Aufbau der Komponente Panel

- Die *Füllung (Fill)* stellt den Bereich dar, der später mit Inhalt aufgefüllt werden kann. Im Standard ist dieser Bereich weiß und zu 100% sichtbar.
- Der *Rahmen (Border)* hat bei Panel und TitleWindow eine besondere Form und bettet die eigentliche Füllung ein. Der obere Rahmen ist breiter gehalten, um die Titel aufzunehmen. Die Opazität ist nicht auf 100% gesetzt, um die Komponenten an die Anwendung anzupassen. Die Eckenrundungen des Rahmens sind nur oben angebracht, um die Wirkung einer Karte zu verstärken, wie auch der eingesetzte Schatten.
- Der *Titel* stellt eine Kennzeichnung oder Überschrift für den späteren Inhalt dar.

Die beschriebenen Eigenschaften lassen sich im Design-Modus in den Flex Properties wieder visuell einstellen. Für die manuelle Einrichtung der Komponente zeigt Tabelle 6-5 die wichtigsten Parameter der Komponenten.

Tabelle 6-5: Parameter der Komponenten Panel und TitleWindow

Parameter	Beschreibung
title	Dieser Parameter stellt die Beschriftung der Karte in Textform dar, die als Überschrift auf den Inhalt hinweist.
borderColor	Die den Inhalt umschließende Fläche lässt sich durch diesen Parameter einfärben.
alpha	Der Alphawert bestimmt die Sichtbarkeit des Rahmens der Karte.
cornerRadius	Mit diesem Parameter können Sie die Rundungen des Rahmens verändern, standardmäßig steht ein Radius von 4.
backgroundColor	Die eigentliche Fläche der Komponenten, die mit Inhalt aufgefüllt werden kann, wird über diesen Parameter farblich verändert.
backgroundAlpha	Soll die Inhaltsfläche ebenfalls transparent dargestellt werden, kann das über diesen Parameter bewirkt werden.



Selbstverständlich gelten auch bei den Komponenten Panel und TitleWindow die Angaben für Maße und Position.

Jedes Panel und jedes TitleWindow, das Sie aus dem Bedienfeld *Components* in die Anwendung ziehen, hat anschließend erst einmal alle Standardwerte. Änderungen sind dann selbstverständlich im Bedienfeld *Flex Properties* in den einzelnen Bereichen möglich. Lassen Sie vorerst bitte noch die Finger von der Einstellung der Liniart des Rahmens, dazu kommen wir weiter unten in diesem Kapitel.



Abbildung 6-10: Verschiedene Panels nach der Änderung einzelner Parameter

Selbstverständlich können diese Änderungen auch innerhalb des Quellcodes erfolgen. Im Folgenden sehen Sie den Quellcode für die Panels in Abbildung 6-10.

```
<mx:Panel x="10" y="10" width="190" height="160" layout="absolute"
title="Standard Panel">
</mx:Panel>
```

```
<mx:Panel x="208" y="10" width="190" height="160" layout="absolute"
title="Panelvariante 2"
borderColor="#9B0C0C"
cornerRadius="11"
alpha="0.57"
backgroundColor="#E78C8C"
backgroundAlpha="0.8">
</mx:Panel>
```

```
<mx:Panel x="406" y="10" width="190" height="160" layout="absolute"
title="Panelvariante 3"
cornerRadius="0"
borderColor="#000000"
alpha="1.0"
color="#FBFDFF"
backgroundColor="#000000">
</mx:Panel>
```

Titel der Komponenten Panel und TitleWindow

Auch die Titel der Komponenten können Sie individuell gestalten. Abgesehen vom Textinhalt selbst, den Sie statisch oder dynamisch einbinden können, lassen sich die Eigenschaften des Textes ebenfalls individuell anpassen. Ihnen stehen im Bereich *Text* der Kategorie *Style* im Bedienfeld *Flex Properties* die üblichen Werkzeuge zur Verfügung, um das Aussehen des Titels zu beeinflussen. Bei der Betrachtung

tung des erzeugten Quellcodes bzw. bei der manuellen Änderung der Eigenschaften des Titeltextes werden Sie sehen, dass eine Ähnlichkeit mit dem Einsatz von Cascading Style Sheets nicht zu verleugnen ist. Tabelle 6-6 zeigt eine Übersicht über die Texteeigenschaften, die auch visuell im Bedienfeld *Flex Properties* einstellbar sind.

Tabelle 6-6: Die Texteeigenschaften

Texteeigenschaft	Beschreibung
fontFamily	Dieser Parameter erlaubt die Zuweisung einer Schriftart. fontFamily="Courier New"
fontSize	Die Größe der Schrift können Sie mithilfe dieser Eigenschaft in Pixeln angeben. fontSize="13"
fontWeight	Um die Stärke der Schrift zu verändern, können Sie diesen Parameter einsetzen. fontWeight="bold"
fontStyle	Damit kann der Stil des Komponententitels angepasst werden. fontStyle="italic"
textDecoration	Hiermit können Sie die Titel der Komponenten durch eine Linie hervorheben. textDecoration="underline"
color	Die farbliche Gestaltung der Schrift erfolgt über die Zuweisung eines Wertes für diesen Parameter. color="#11B6D9"

Betrachten wir die komplette Komponente im Bezug auf die Formatierung des Titels für das Panel nochmals im Ganzen. Vergleichen Sie dazu auch Abbildung 6-10.

```
<mx:Panel x="406" y="10" width="190" height="160" layout="absolute"
cornerRadius="0" borderColor="#000000" alpha="1.0" backgroundColor="#000000"
title="Panelvariante 3"
color="#FAFBFC"
fontFamily="Georgia"
fontWeight="bold"
fontStyle="italic"
textDecoration="underline"
fontSize="14" >
```

```
</mx:Panel>
```

Die Zuweisung eines vorher festgelegten Style nach dem Prinzip der Arbeit mit CSS ist auch im Flex-Framework möglich. Wir werden uns weiter unten in diesem Kapitel damit beschäftigen.

Übung 4: Panel-Komponenten einfügen und anpassen

- 1 Öffnen Sie die Anwendung *ue_03.xml* und speichern Sie sie unter dem neuen Namen *ue_04.xml* ab.
- 2 Lassen Sie sich die Anwendung im Design-Modus anzeigen; öffnen Sie dazu das Bedienfeld *Components* und wählen Sie die Kategorie *Layout*.
- 3 Ziehen Sie die Panel-Komponente in den Bereich links unter der weißen Hülle in den ersten Bereich der angelegten HBox.
- 4 Ändern Sie in den *Flex Properties* die Breite des Panels auf 200 und die Höhe auf 165 Pixel.
- 5 Ziehen Sie eine weitere Panel-Komponente nach unterhalb des angelegten Panels und ändern Sie dort Breite und Höhe auf 200 Pixel.
- 6 Ein weiteres Panel soll unterhalb der aktuellen Panels angelegt werden. Ziehen Sie das Panel erneut aus dem *Components*-Bedienfeld in den linken Bereich der HBox. Ändern Sie hier die Breite auf 200 und die Höhe auf 75 Pixel.
- 7 Wählen Sie nun im *Outline*-Bedienfeld das oberste Panel aus und öffnen Sie in den *Flex Properties* den Bereich *Common*. Tragen Sie dort als Titel des obersten Panels im Feld *Title* »Aktuelle Ausstellung« ein.
- 8 Wählen Sie nun das Panel darunter aus und verfahren Sie wie in Schritt 7. Der Titel soll in diesem Panel »Neu im Galerie-Shop« lauten.
- 9 Auch das unterste Panel soll einen Titel erhalten: »Impressum«.
- 10 Ändern wir nun noch das Aussehen der Panels. Wählen Sie dazu wieder das erste Panel aus und ändern Sie im Bereich *Style* der *Flex Properties* die Textfarbe mit #FF9900 in Orange.
- 11 Ändern Sie auch die Schriftfarbe des Panels zwei in diesen Wert.
- 12 Im dritten Panel nehmen wir nun zusätzlich noch eine weitere Änderung vor. Wählen Sie dazu im Bereich *Border* die Farbe Weiß (#FFFFFF) und stellen Sie anschließend den Linientyp auf *solid*. Dadurch wird das Panel weiß. Für den Titel legen Sie eine graue Schriftfarbe (#999999) fest.
- 13 Wechseln Sie nun in den Source-Modus, um dort die Panels manuell bearbeiten zu können. Suchen Sie die zweite VBox, in der die beiden HBox-Komponenten eingetragen wurden.

```
<mx:VBox height="505" width="590" x="0" y="0">
  <mx:HBox width="590" height="250">
  </mx:HBox>
  <mx:HBox width="590" height="245">
  </mx:HBox>
</mx:VBox>
```
- 14 Ergänzen Sie nun in den HBoxes jeweils zwei Panels, die eine Breite von 290 und eine Höhe von 245 Pixel haben sollen.

```
<mx:HBox width="590" height="250">
  <mx:Panel width="290" height="245" layout="absolute">
  </mx:Panel>
  <mx:Panel width="290" height="245" layout="absolute">
  </mx:Panel>
</mx:HBox>
```
- 15 Starten Sie die Anwendung. Sie werden eine Sammlung von Komponenten sehen. Die Panel auf der linken Seite sind bereits mit einem Titel versehen worden, und die Schriftfarbe wurde verändert. Die in Schritt 14 erstellten Panel-Komponenten haben aktuell noch das Standardaussehen dieser Komponente.
- 16 Nun geht es noch darum die Standardwerte der vier Panels aus Schritt 14 zu ändern. Ergänzen Sie deshalb im XML-Tag der Panels einige Angaben.

```
<mx:Panel width="290" height="245" layout="absolute" cornerRadius="0"
borderStyle="solid" backgroundColor="#000000" color="#FFFFFF">
```

Übung 4: Panel-Komponenten einfügen und anpassen (Fortsetzung)

- 17** Zum Abschluss müssen den vier Panels noch Titel zugewiesen werden. Arbeiten Sie sich dabei von links oben nach rechts unten vor und nutzen Sie folgende Vorgaben.

Künstlerische Fotografien
Poster in verschiedenen Größen
Grafiken
Rahmen

- 18** Starten Sie die Anwendung.
-

Damit steht die Konstruktion der Startseite unserer Anwendung und sollte Abbildung 6-11 entsprechen. Wie Sie CSS-Stil-Definitionen zur Gestaltung der Komponenten und Texte erstellen und einsetzen können, betrachten wir weiter unten in diesem Kapitel.

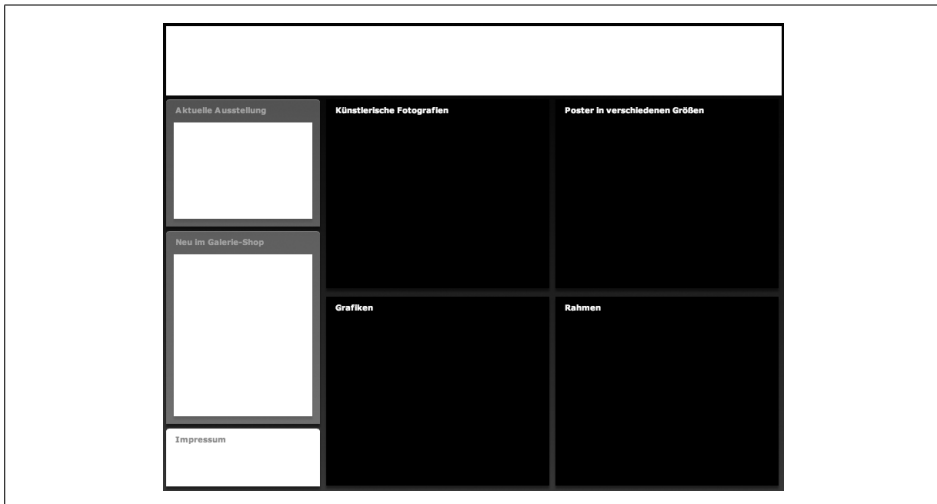


Abbildung 6-11: Die Anordnung der Komponenten vertikal, horizontal und absolute

Geben wir nun der Anwendung noch Inhalt, da die Struktur allein ja nicht unbedingt anspruchsvoll ist und den Benutzer nicht gerade zum Verweilen und Interagieren einlädt. Schauen wir uns vorerst die Elemente Text und Bilder an. Andere Medien sind Bestandteil von Kapitel 12.

Textinhalte

Beginnen wir mit dem Text, da dieser sehr häufig und in unterschiedlichster Form innerhalb einer Webanwendung zum Einsatz kommt und innerhalb des Flex-Frameworks dafür auch verschiedene Komponenten vorgesehen sind. Ich möchte an dieser Stelle nur auf den Text und seine Formen eingehen – die funktionellen

Komponenten, die in erster Linie in Formularen zum Einsatz kommen, werden in Kapitel 13 betrachtet. Klären wir zunächst, in welcher Form der Text in einer Anwendung zum Einsatz kommt. Tabelle 6-7 zeigt eine Übersicht über die möglichen Textformen.

Tabelle 6-7: Textformen innerhalb einer Flex-Anwendung

Textform	Beschreibung
Titel	Bezeichner auf dem Panel, in Navigationssystem, auf Schaltflächen und in anderen Textelementen
Bezeichner	Beschriftungen von Elementen und angezeigten Objekten in einer Anwendung
Mengentexte	Beschreibungen und Dokumentationen mit einer größeren Menge von Text, meistens über mehrere Zeilen verteilt und oft mit einem Scrollbalken versehen
Hinweistexte	Beschreibungen und Anmerkungen, die oft nur auf Mausaktionen hin kurzzeitig sichtbar werden und dem Anwender zusätzliche Informationen geben; diese Hinweistexte können auch über mehrere Zeilen verteilt sein
Eingabetexte	Textfelder, in die der Benutzer Texte eingeben kann
Verlinkungstexte	Texte, die als Verlinkung benutzt werden, um eine Form von Navigation zwischen den einzelnen Themen und Bereichen der Anwendung zu ermöglichen
Elementinhalte	Texte, die in anderen speziell konfigurierten Komponenten eingesetzt werden. Wir werden uns in Kapitel 7 mit diesen Komponenten näher beschäftigen.

Beginnen wir zunächst einmal damit, Texte in einer Anwendung anzeigen zu lassen. Es stehen Ihnen zwei Komponenten für diese Aufgabe zur Verfügung:

mx:Label

Diese Komponente stellt den angegebenen Text mit der festgelegten Formatierung innerhalb der Anwendung dar. Der Unterschied zu *mx:Text* besteht darin, dass hier der Text nur einzeilig angezeigt werden kann. Benutzen Sie diese Komponente für Beschriftungen, Hinweise oder Überschriften.

mx:Text

Diese Komponente gibt den eingetragenen Text wieder, genau wie die Label-Komponente. Der Unterschied zu Label besteht darin, dass der Text in mehreren Zeilen angezeigt werden kann. Diese Komponente eignet sich deshalb gut für Fließtexte, Sie können sie aber auch für einzeilige Texte nutzen.

Die Anwendung dieser Komponenten ist so einfach, dass Sie es sicherlich auch ohne die Hilfe des Buches hinbekommen werden, da sich die Arbeitsschritte immer wieder gleichen.



Achten Sie bei der Arbeit mit dynamischen Werten und Inhalten darauf, dass sich die Label-Komponente horizontal ausdehnt, wenn der Textinhalt die Breite der Komponente überschreitet, bei der Text-Komponente hingegen erfolgt ein Zeilenumbruch und die Ausdehnung nach unten.

Betrachten Sie im Design-Modus das Bedienfeld *Components* und dort die Kategorie *Controls*, die beide Komponenten enthält. Von dort aus lassen sich beide Varianten einfach in die Anwendung ziehen. Im Bedienfeld *Flex Properties* können Sie nun den Text für die Anzeige festlegen, indem Sie entweder im Feld *Label* oder im Feld *Text* den Inhalt eingeben. Selbstverständlich können Sie auch innerhalb der Label-Komponente oder der Text-Komponente direkt in der Anwendung schreiben. Klicken Sie dazu einfach nur in den aktuellen Text der jeweiligen Komponente. Das Ergebnis zeigt Abbildung 6-12.

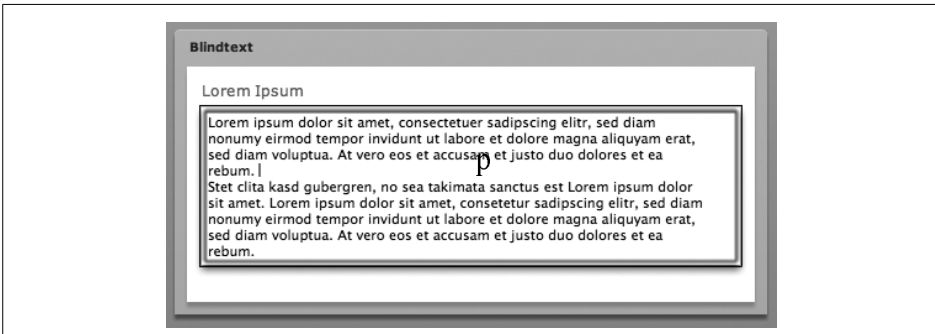


Abbildung 6-12: Änderung des Textinhalts direkt im Design-Modus des Flex Builder

Zur Beschreibung habe ich wieder eine Panel-Komponente zur Aufnahme der Komponenten Label und Text eingesetzt. Selbstverständlich können die beiden Komponenten auch innerhalb des Quellcodes manuell erzeugt werden.

```
<mx:Panel x="54" y="106" width="481" height="230" layout="absolute"
title="Blindtext">

    <mx:Label x="10" y="10" text="Lorem Ipsum" fontFamily="Verdana"
color="#2C7382" fontSize="12" width="95"/>

    <mx:Text x="10" y="30" text="Lorem ipsum dolor ... rebum. " width="441"
height="131" />

</mx:Panel>
```

Die Textformatierung erfolgt wie bei der Formatierung des Parameters *title*, den wir schon bei den Arbeiten mit dem Panel verändert haben. Entweder Sie benutzen dazu innerhalb des Flex Builder wieder das Bedienfeld *Flex Properties* oder die Parameter bei der Erzeugung der Inhalte im Quellcode.

Das Layout des Panels steht in unserem Beispiel auf *absolute*, d.h. die Inhalte werden absolut positioniert. Gestalterisch muss dann darauf geachtet werden, welche Abmessungen und welche Positionen die eingesetzten Elemente haben werden. Eine andere Möglichkeit besteht darin, dem Panel einen Parameter zuzuweisen, der einen Abstand zwischen dem Rand und dem eigentlichen Inhalt festlegt; dieser als Innenabstand bezeichnete Wert wird durch die Eigenschaft *padding* angegeben.

Ändern Sie dazu einfach das Layout des Panels in `vertical`. Die Werte für `x` und `y` werden so beim Laufen der Anwendung ignoriert und die Texte an die linke Kante verschoben. Um nun wieder einen Abstand zur Kante der Anzeigefläche zu bewirken, können für das Panel die Werte für den Abstand festgelegt werden, die Sie dem Quellcode entnehmen können und die sich selbst erklären.

```
<mx:Panel x="54" y="106" width="481" height="230" title="Blindtext"
  layout="vertical"
  paddingTop="10"
  paddingRight="10"
  paddingBottom="10"
  paddingLeft="10">

  <mx:Label text="Lorem Ipsum" fontFamily="Verdana" color="#2C7382" fontSize="12"
    width="95"/>
  <mx:Text text="Lorem ipsum dolor ... rebum. " width="441" height="131" />
</mx:Panel>
```

Merken Sie sich, dass die Text-Komponente immer dann eingesetzt werden muss, wenn Sie einen Text über mehrere Zeilen anzeigen lassen wollen.

Übung 5: Texte in die Startseite der Anwendungen einbinden

- 1 Öffnen Sie die Datei `ue_04.mxml` und speichern Sie sie unter dem Namen `ue_05.mxml`.
- 2 Wählen Sie nun den Design-Modus und öffnen Sie die Kategorie `Controls` im Bedienfeld `Components`.
- 3 Ziehen Sie die Komponente `Text` in den linken Bereich des weißen Rahmens im oberen Bereich der Anwendung. Ziehen Sie den Rahmen größer.
- 4 Schreiben Sie folgenden Text in diesen Textcontainer


```
www.galerie-flexibel.de
Praxiswissen Flex 3 - O'Reilly Verlag - 2008
Dezember 2008
```
- 5 Platzieren Sie nun noch den Textcontainer exakter, indem Sie im Bedienfeld `Flex Properties` diese Eigenschaften einstellen.
Breite: 350, Höhe: 60 X: 10, Y: 10
- 6 Ändern Sie nun die Einstellungen der Schriftfarbe auf den helleren Grauwert `#333333`.
- 7 Wechseln Sie nun in den Source-Modus und erstellen Sie dort die folgenden Tags für die aktuelle Ausstellung manuell. Achten Sie in den folgenden Codezeilen auf die Angabe von Sonderzeichen und Zeilenumbrüchen. Für einen Zeilenumbruch erfolgt die Angabe der Zeichen ``, das kaufmännische UND (&) wird durch `&` im Text angegeben.

```
<mx:Panel width="200" height="165" layout="absolute" title="Aktuelle
Ausstellung" color="#FF9900">
  <mx:Label x="0" y="4" text="TENERIFFA" width="180" textAlign="center"
    fontSize="14" fontWeight="bold" color="#911400"/>
  <mx:Text x="10" y="24" text="Fotowanderung von&#xd; Gerald Reinhardt"
    color="#666666" width="160" height="33" textAlign="center" fontSize="11"/>
</mx:Panel>
```

Übung 5: Texte in die Startseite der Anwendungen einbinden (Fortsetzung)

- 8 Ergänzen Sie nun noch das Panel für den Galerie-Shop in Vorbereitung auf die spätere Einbindung von Bildern.

```
<mx:Panel width="200" height="249" layout="absolute" color="#FF9900"
title="Neu im Galerie-Shop">
<mx:Label x="59" y="7" text="Fotoalben" fontSize="10" color="#000000"
fontWeight="bold"/>
<mx:Text x="28" y="111" text="Klebeecken & Strips" fontSize="10"
fontWeight="bold" color="#000000"/>
</mx:Panel>
```
 - 9 Kümmern wir uns nun noch um den Textinhalt im Impressum, den wir einfach halten werden, indem wir ein Label einsetzen, das eine kurze Beschreibung enthält und darunter den Namen des »Künstlers« (ich habe meinen gewählt). Nutzen Sie getrost Ihren eigenen.
Wechseln Sie wieder in den Design-Modus und ziehen Sie jeweils eine Label-Komponente und eine Text-Komponente in den Bereich Impressum und versuchen Sie diese ohne Vorgaben nach eigenem Ermessen anzupassen.
 - 10 Starten Sie die Anwendung.
-

Kleine Hilfstexte

Dezent eingesetzt erweitern kleine Hinweise die Nutzbarkeit und damit die Qualität einer Anwendung. Dazu wird bei Berührung des jeweiligen Objekts ein Hinweis in Form eines einfachen Fähnchens, das den festgelegten Text enthält, mit einer kurzen Zeitverzögerung angezeigt. Abbildung 6-13 zeigt den Hilfetext, der eingeblendet wird, wenn die Maus länger auf dem Button *Demonstration* verharrt.



Abbildung 6-13: Der Hinweistext wird als kleines Fähnchen am Button angezeigt

Die Zuweisung dieser Hinweistexte erfolgt bei den geeigneten Komponenten über einen Parameter, der innerhalb des MXML-Tags manuell mitangegeben werden muss. Im Bedienfeld *Flex Properties* ist dafür aktuell noch keine Option enthalten. Der Parameter für diese kleinen Hinweistexte heißt `toolTip`; ihm wird dann über den üblichen Wert der Hinweistext zugewiesen.

```
<mx:Button label="Demonstration" x="90" y="90"
toolTip="Wenn Sie hier klicken, ändern Sie die Welt auch nicht!" />
```

Der Text sollte natürlich passend ausgewählt und zugewiesen werden, und dabei ist weniger manchmal mehr. Achten Sie auch unbedingt darauf, welchen Elementen Sie diese Hinweise zuordnen, da Ihre Anwendung unter Umständen überladen wirken würde, wenn Sie jeder Komponente einen Hinweistext zuweisen.



Setzen Sie diese kleinen Hilfestellungen an der Stelle ein, wo dem Benutzer nicht aufgrund des Buttons oder anderer Navigationselemente ohnehin schon klar ist, was bei einem Klick auf den Button oder das Element passieren wird.

Übung 6: Einen Hinweistext zuweisen

- 1 Öffnen Sie die Flex-Anwendung *ue_05.xml* und speichern Sie sie unter dem Namen *ue_06.xml*.
- 2 Suchen Sie im Bedienfeld *Outline* das Panel (Impressum) und wählen Sie es aus.
- 3 Ergänzen Sie nun den MXML-Tag für diese Komponente um einen Hinweistext, indem Sie innerhalb des Tags folgenden Inhalt ergänzen:

`toolTip="Hier erfahren Sie Hintergründe zum Macher dieser Anwendung"`

Das komplette Tag muss nach dieser Ergänzung folgendermaßen aussehen:

```
<mx:Panel width="200" height="75" layout="absolute" title="Impressum"
borderColor="#FDFDFD" alpha="1.0" borderStyle="solid" color="#999999"
toolTip="Hier erfahren Sie Hintergründe zum Macher dieser Anwendung" >
```

- 4 Starten Sie die Anwendung.
-

Bilder integrieren

Typische Aufmacher moderner Internetanwendungen sind grafische Elemente und Fotos. Waren Bildinformationen in den Anfangsjahren des Internet noch mit der unausgesprochenen Auflage versehen, sie sparsam und gezielt einzusetzen, kommt es mittlerweile mehr auf die Bildkomposition an. Dabei spielt es keine Rolle, ob die Inhalte zur Gestaltung dienen, indem sie als Hintergrundbilder, Symbole oder Piktogramme eingesetzt werden, oder zur Unterstützung des Inhalts. Selbstverständlich kann das Bild auch der darzustellende Hauptinhalt sein, wie es in unserer Galerie ja der Fall sein wird.

An dieser Stelle sollten Sie sich aber nochmals an die Philosophie des Flex-Frameworks erinnern und daran denken, dass hier ausschließlich die Anwendung zusammengestellt wird. Die eingesetzten Bilder bzw. Grafiken müssen aus diesem Grund im Vorfeld in einem geeigneten Programm erstellt oder bearbeitet werden. Für die Vorbereitung der notwendigen Dateien ist es sehr wichtig, die Dateiformate zu kennen, die als Bilder und Grafiken in Flex-Anwendungen eingesetzt werden können. Eine knappe Übersicht sehen Sie in Tabelle 6-8.

Tabelle 6-8: Geläufige Formate für Bilder im Web

Dateiformat	Beschreibung
JPEG	JPEG ist eine Norm für verschiedene Formen der Bildkompression, die auf die Joint Photographic Experts Group zurückgeht. Bei der Komprimierung werden unterschiedliche Farbräume und Farbtiefen verwendet, um die Dateigrößen zu verringern. Dieses Dateiformat wird vorrangig für Fotos und vielfarbige Grafiken verwendet.

Tabelle 6-8: Geläufige Formate für Bilder im Web (Fortsetzung)

Dateiformat	Beschreibung
GIF	Bei GIF-(Graphics Interchange Format-)Dateien erfolgt eine verlustfreie Komprimierung von Bildern mit geringer Farbtiefe (bis zu 256 Farben pro Einzelbild). Für Logos und einfache unspektakuläre Grafiken ist dieses Grafikformat bestens geeignet. Beim Einsatz in Browsern und anderen geeigneten Programmen lassen sich mit GIF-Dateien auch Bildfolgen speichern und als Animationen anzeigen.
PNG	Die neueste Form der Komprimierung von Bilddaten erfolgt über das Format PNG (<i>Portable Network Graphics</i>). Dieses Format speichert die Bildinformationen verlustfrei und lässt unterschiedliche Farbtiefen zu. PNGs können Transparenzen aufweisen. Dazu nutzen sie den Alphakanal, einen zusätzlichen Kanal, der eine Farbe für die Transparenz speichert. Diese Eigenschaften erhöhen die Bedeutung der PNG-Dateien.



Die Beschränkung auf diese drei Dateiformate liegt an Einschränkungen der Bilddarstellung innerhalb von Browsern. In anderen Anwendungen von Adobe ist es auch möglich, andere Formate wie zum Beispiel Photoshop-Dateien einzusetzen. Hier erfolgt allerdings innerhalb der Software eine Umwandlung in ein geeignetes Format. Vielleicht wird das in Zukunft auch im Flex-Framework der Fall sein.

Dateien vorbereiten

Zur Vorbereitung des Einbindens von Fotos und grafischen Elementen gehört auch die Organisation der Dateien in dafür angelegten Ordnern. Für die einzubindenden Inhalte sollten Sie im Ordner Ihres Flex-Projektes einen Ordner anlegen, in dem alle Inhalte gespeichert werden, auf die zugegriffen werden soll. Ich habe mich dabei für die Ordner-Bezeichnung *assets* entschieden. Innerhalb dieses Ordners können Sie Ihre externen Inhalte speichern. Achten Sie genauestens auf die Notwendigkeit der Dateien für Ihre Anwendung, um auszuschließen, dass in diesem Ordner Daten gespeichert werden, die in der entwickelten Anwendung nicht zum Einsatz kommen. Alle eingesetzten Dateien müssen zur Laufzeit im angegebenen Ordner zur Verfügung stehen, da sie während des Ablaufs der Anwendung geladen werden – unabhängig davon, ob die Dateien statisch angegeben werden, über Data Binding zugewiesen werden oder erst während der Laufzeit der Anwendung feststehen.

Absolute und relative Angaben

Die Zuweisung einer Bilddatei erfolgt in einer Flex-Anwendung innerhalb des genutzten MXML-Tags als Parameter, der als Wert die Angabe der Datei und des jeweiligen Speicherortes enthält, also des Pfades, über den diese Datei während der Benutzung in die Anwendung geladen wird. Das ist vollkommen unabhängig davon, ob das Bild nun als Inhalt eingesetzt wird oder zur grafischen Gestaltung der Anwendung zum Einsatz kommt. Dabei unterscheidet man zwei Verfahren:

Absolute Pfade

Hier erfolgt die Zuweisung der gewünschten Datei durch die komplette Angabe des Speicherortes und des Dateinamens, normalerweise durch eine URL wie z.B. `http://www.websserver.de/bilder/bilddatei.jpg`.

Relative Pfade

Die Angabe der einzubindenden Datei erfolgt bei dieser Variante nicht über die vollständige Angabe des Speicherortes, sondern relativ zur Anwendung, indem der direkte Weg von der Anwendung zur Datei angegeben wird bzw. die Wege über die vorliegende Ordnungsstruktur. Dazu müssen sich die Dateien für die Anwendung und die Bilder innerhalb eines übergeordneten Ordners befinden, der bei der Erstellung des Flex-Projektes als Projektordner ausgewählt wurde.

Bilder einbinden

Für den Einsatz eines Bildes als inhaltlicher Bestandteil in der Flex-Anwendung gibt es ein MXML-Tag, das die Möglichkeit bietet, das Bild in die Anwendung zu integrieren und zusätzlich mit weiteren Eigenschaften zu versehen. Neben dem Namen der Komponente ist die Angabe der Quelle wichtig:

```
<mx:Image source="Angabe zur Bilddatei" />
```

Mit dem Kürzel `src` (für engl. *source*, Quelle) wird die Quelldatei angegeben. Und diese Quellangabe erfolgt nun als absolute Angabe oder relativer Pfad.

```
<mx:Image source="http://www.websserver.de/bilder/foto01.jpg" />
```

```
<mx:Image source="../../../fotos/foto02.jpg" />
```

Innerhalb des Tags können Sie weitere Angaben für die Bilder ergänzen, die Sie bereits aus anderen Komponenten kennen.

Selbstverständlich können Sie im Design-Modus die Komponente auch aus dem Bedienfeld *Components* aus der Kategorie *Controls* in die Anwendung ziehen und in der gewünschten Komponente platzieren. Innerhalb der Anwendung wird dazu vorerst nur ein Platzhalter angezeigt, dem nun noch die Bilddatei im Feld *Source* oder über den Ordner zugewiesen werden muss (siehe Abbildung 6-14).

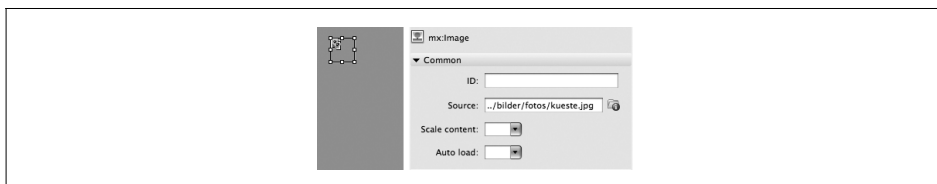


Abbildung 6-14: Der Image-Platzhalter und die Zuweisung der Quelldatei des Bildes

Nach Bestätigung der ausgewählten Datei wird das Bild nun in den angezeigten Platzhalter integriert, und seine aktuellen Maße werden für Breite und Höhe auto-

matisch übernommen. Die weiteren Eigenschaften des Bildes können innerhalb des Bedienfeldes und im Source-Modus manuell verändert werden.



Achten Sie beim Einbinden von Bildern, die auf einem Server abgelegt sind, immer darauf, dass Sie auch bei der Arbeit im Flex-Builder online sein müssen, wenn Sie diese Bilder sehen wollen.

Hintergrundbilder

Die Einbindung eines Hintergrundbildes ist auch für die vorhandenen Layoutkomponenten möglich. Dazu müssen Sie in den Source-Modus wechseln und die entsprechende Komponente auswählen und dort die Eigenschaft `backgroundImage` einfügen, die genauso behandelt werden muss wie `src`.

```
<mx:Canvas width="400" height="200" backgroundImage=" ../fotos/berge.jpg" >
</mx:Canvas>
```

Der Vorteil des Einsatzes eines Hintergrundbildes ist, dass auf dem Bild weitere Inhalte platziert werden können, was sonst nur bei der Anwendung des Layouts absolute und der Übereinanderstapelung von Komponenten möglich wäre.

Übung 7: Bilder in die Galerie FLEXibel einbinden

- 1 Öffnen Sie die Datei `ue_06.xml` und speichern Sie sie unter dem neuen Namen `ue_07.xml`.
- 2 Wechseln Sie in den Design-Modus und öffnen Sie im Bedienfeld *Components* die Kategorie *Controls*.
- 3 Ziehen Sie eine Image-Komponente in den Kopfbereich der Anwendung in den rechten Bereich.
- 4 Wählen Sie im Bedienfeld *Flex Properties* hinter dem Feld *Source* das kleine Ordnersymbol und suchen Sie die Datei `/assets/images/elemente/galerieLogo.png`.
- 5 Richten Sie die Komponente im Kopf der Anwendung nach eigenem Ermessen aus.
- 6 Ziehen Sie eine Image-Komponente in das Panel für die »Aktuelle Ausstellung« und platzieren Sie das Bild unterhalb der Komponente.
- 7 Wählen Sie im Bedienfeld *Flex Properties* hinter dem Feld *Source* das kleine Ordnersymbol und suchen Sie die Datei `/assets/images/motive/canjadas.jpg`.
- 8 Richten Sie das Bild zentriert im Panel aus.
- 9 Ziehen Sie eine Image-Komponente in das Panel für die »Shop-Angebote« und platzieren Sie das Bild unterhalb des Labels »Fotoalben«.
- 10 Wählen Sie im Bedienfeld *Flex Properties* hinter dem Feld *Source* das kleine Ordnersymbol und suchen Sie die Datei `/assets/images/produkte/alben.jpg`.
- 11 Richten Sie das Bild zentriert im Panel aus.
- 12 Ziehen Sie eine Image-Komponente in das Panel für die »Shop-Angebote« und platzieren Sie das Bild unterhalb des Labels »Klebeecken und Strips«.
- 13 Wählen Sie im Bedienfeld *Flex Properties* hinter dem Feld *Source* das kleine Ordnersymbol und suchen Sie die Datei `/assets/images/produkte/ecken.jpg`.
- 14 Richten Sie das Bild zentriert im Panel aus.

Übung 7: Bilder in die Galerie FLEXibel einbinden (Fortsetzung)

- 15 Wechseln Sie in den Source-Modus. Nun sollen den vier Beispiel-Panels noch die Darstellungsfotos zugewiesen werden.
- 16 Suchen Sie das Panel für »Künstlerische Fotografien« und fügen Sie innerhalb des Panels folgende Image-Komponente ein:

```
<mx:Image width="250" height="200" source="../../assets/images/beispiele/collageBlack.jpg"
horizontalCenter="-3" verticalCenter="-3"/>
```
- 17 Suchen Sie das Panel für »Poster in verschiedenen Größen« und fügen Sie innerhalb des Panels folgende Image-Komponente ein:

```
<mx:Image width="250" height="200" source="../../assets/images/beispiele/buntstiftBlack.jpg"
horizontalCenter="-3" verticalCenter="-3"/>
```
- 18 Suchen Sie das Panel für »Grafiken in verschiedenen Stilrichtungen« und fügen Sie innerhalb des Panels folgende Image-Komponente ein:

```
<mx:Image width="250" height="200" source="../../assets/images/beispiele/freskoBlack.jpg"
horizontalCenter="-3" verticalCenter="-3"/>
```
- 19 Suchen Sie das Panel für »Unterschiedliche Rahmen und Aufteilungen« und fügen Sie innerhalb des Panels folgende Image-Komponente ein:

```
<mx:Image width="250" height="200" source="../../assets/images/beispiele/teilungBlack.jpg"
horizontalCenter="-3" verticalCenter="-3"/>
```
- 20 Starten Sie die Anwendung.

Damit ist die Startseite für die Galerie FLEXibel fertig gelayoutet und sollte aussehen wie in Abbildung 6-15. Sollte das nicht der Fall sein, prüfen Sie bitte die zugewiesenen Werte aus den vorangegangenen Übungen.

Nachdem nun der prinzipielle Aufbau einer Anwendung demonstriert wurde, wobei auf die wesentlichsten Inhalte einer Anwendung Bezug genommen wurde, widmen wir uns einer anderen Weise der Formatierung und Zuweisung von Eigenschaften.

Mit Cascading Style Sheets arbeiten

Auch in einer Flex-Anwendung ist die Arbeit mit Stilen auf der Basis von Cascading Style Sheets vorgesehen und sinnvoll. Für alle Einsteiger, die sich bisher mit diesem Formatierungsschema noch nicht vertraut gemacht haben, folgt nun ein kleiner Überblick.

Mit Stilen definieren Sie Eigenschaften für die Darstellungen von Texten und das Aussehen von Komponenten. Bei der Definition neuer Stile werden den Eigenschaften der Komponenten Werte zugewiesen, die vom Standard abweichen. Dadurch wird Einfluss auf die optische Wirkung der einzelnen Komponenten bzw. einzelne Eigenschaften der Komponenten ausgeübt.



Abbildung 6-15: Das Layout der Startseite der Galerie FLEXibel

Der Einsatz von Stilen wird aufgrund dreier Merkmale mit Vorliebe auch bei der Entwicklung von Websites mit (X)HTML und CSS eingesetzt. Betrachten wir diese drei Merkmale.

Trennung von Gestaltung und Inhalt

Die Anlage der Stildefinitionen passiert im MXML-Dokument im Kopfbereich. Dabei erfolgt eine Zusammenstellung der Formatierungseigenschaften, die später automatisch zugewiesen werden oder per Zuweisung zum Einsatz kommen. Dadurch wird die Gestaltung nicht mehr direkt im MXML-Tag zusammengestellt.

Mehrfachnutzung

Aufgrund der automatischen oder manuellen Zuweisung der Stildefinitionen können die festgelegten Eigenschaften immer wieder benutzt werden, ohne sie jedes Mal neu zu erstellen. Das macht sich besonders bemerkbar, wenn Sie die Eigenschaften ändern, da sich eine einzelne Änderung automatisch auf alle Komponenten auswirkt.

Verschachtelung

Aus dem Begriff »Cascading« kann man entnehmen, dass diese Formatierungen miteinander verschachtelt werden können: In einer Definition wird die Schriftart definiert, in einer anderen die Größe und in der nächsten wieder die Farbe. Miteinander verschachtelt ergeben diese Stile das Aussehen der benutzen Komponenten.

Nehmen wir das Ganze am einfachen Beispiel der Zuweisungen einmal genauer unter die Lupe. Beginnen wir dazu mit der bisherigen Formatierung von Objekten:

```
<mx:Label text="Dieses Label ist nur ein Beispiel."  
fontFamily="Arial"  
fontSize="14"  
color="#0000FF"  
fontWeight="bold" />
```

Sie können diesem MXML-Tag entnehmen, welche Schriftformatierungen ausgewählt wurden, um die Arbeit mit CSS-Stildefinitionen zu erläutern. Dazu werden wir die aktuellen Formatierungen nutzen und sie in Stildefinitionen verschieben. Zu Beginn legen wir innerhalb von `mx:Application` im oberen Teil einen Bereich für die Definitionen an:

```
<mx:Style>  
  
</mx:Style>
```

Innerhalb dieses Bereichs werden nun die Stildefinitionen angelegt. Dabei gibt es drei unterschiedliche Formen, die entsprechend ihrer Anlage innerhalb der Anwendung unterschiedliche Auswirkungen haben.

Globale Stile

Diese Stile werden nach der Definition automatisch für alle eingesetzten Komponenten gültig und dementsprechend auch benutzt. Die Festlegung eines Stils erfolgt nach einem schon anderweitig benutzten Prinzip: Der Stilname wird angegeben, und dann werden innerhalb von geschweiften Klammern alle Definitionen für diesen Stil zusammengestellt.

```
<mx:Style>  
  global  
  {  
    fontWeight: bold;  
  }  
</mx:Style>
```

Stildefinitionen für Komponenten

An dieser Stelle wird die Bezeichnung einer Komponente als Stilname benutzt, und automatisch werden alle Komponenten dieses Typs mit dieser Stildefinition versehen.

```
<mx:Style>  
  global  
  {  
    fontWeight: bold;  
  }  
  Label
```

```

    {
      fontSize: 14px;
    }
  </mx:Style>

```

Für das Label wurde auf diesem Weg eine Schriftgröße von 14 Pixeln festgelegt. Achten Sie bei der Zuweisung der Werte bitte auch auf die Angaben von Einheiten, wenn es notwendig ist.

Eigene Stile definieren und einsetzen

Natürlich haben Sie auch die Möglichkeit, spezielle Formatierungen und eigene Bezeichnungen innerhalb der Anwendung zu verwenden. Dazu können individuelle Namen bei der Zuweisung benutzt werden. Wichtig ist bei dieser Definition ein Punkt vor dem Namen, um damit eine eigene Klasse zu definieren.

```

<mx:Style>
  global
  {
    fontWeight: bold;
  }
  Label
  {
    fontSize: 14px;
  }
  .meinStil
  {
    color: #0000FF;
  }
</mx:Style>

```

Nun möchten Sie natürlich die eigentliche inhaltliche Komponente noch anpassen. Achten Sie dabei darauf, dass die definierten Stile nicht mehr mitangegeben werden. Um das Ganze zu verdeutlichen, werden wie eine zweite Komponente hinzunehmen.

```

<mx:Label text="Dieses Label ist nur ein Beispiel."
fontFamily="Arial"
styleName="meinStil" />
<mx:Text text="Dieser Text ist nur ein Beispiel." y="20" />

```

Betrachten Sie die Anzeige einfach im Design-Modus, und Ihnen werden die Formate, die Sie festgelegt haben, angezeigt. Vergleichen Sie die Anzeige der beiden Textausgaben. Beide Texte sind fett, da der Wert `bold` als globaler Stil definiert wurde. Die Schrift in der Label-Komponente hat eine Größe von 14 Pixeln, und durch die Zuweisung der Klasse `meinStil` ist die Schriftfarbe Blau. Der Text in der Komponente Text hat aufgrund der Definition nur die Eigenschaft `bold`.



Achten Sie bei der Anlage von Stilen immer darauf, dass die Definitionen innerhalb des MXML-Tags auf jeden Fall Vorrang haben, damit sie trotz der Anwendung von Stildefinitionen individuell formatiert werden können.

Übung 8: Cascading Style Sheets in der Galerie FLEXibel

- 1 Öffnen Sie die Datei `ue_07.mxml` und speichern Sie sie unter dem Namen `ue_08.mxml`.
- 2 Wechseln Sie in den Source-Modus.
- 3 Legen Sie im oberen Bereich der Anwendung den Block zur Definition der Stile an.

```
<mx:Style>
</mx:Style>
```

- 4 Legen Sie eine Stildefinition für eine eigene Klasse an nennen Sie sie `.hinweis`.

```
<mx:Style>
.hinweis
{
}
</mx:Style>
```

- 5 Definieren Sie für den Stil die Schriftfarbe als blau und für die Stärke `bold`.

```
.hinweis
{
color: #0000FF;
fontWeight: bold;
}
```

- 6 Wenden Sie diesen Stil nun innerhalb des Kopftextes an und löschen Sie die bisherigen Angaben zur Formatierung aus diesem Tag heraus. Achten Sie auch hier wieder auf die Angabe der Zeilenumbrüche.

```
<mx:Text x="10" y="10" width="350" height="65"
text="www.galerie-flexibel.de&#xd;Praxiswissen Flex 3 - O'Reilly Verlag -
2008&#xd;&#xd; Dezember 2008&#xd;"
styleName="hinweis" />
```

- 7 Starten Sie die Anwendung.
-

Stildefinitionen auslagern

Die HTML-Entwickler unter Ihnen werden sich sicher schon gefragt haben, ob es auch möglich ist, die CSS-Stildefinitionen in ein separates Dokument auszulagern. Das hat den Vorteil, dass die Definitionen besser geordnet sind und schnell durch andere ersetzt werden können, um damit das Layout zu verändern. So eine Auslagerung ist möglich, und selbst der Wiederverwendung der kompletten Sammlung von Stilen steht nichts im Wege. Dafür sind zwei Schritte notwendig:

Erstellung einer CSS-Datei

Die sonst im Dokument erstellten Stildefinitionen werden in einer separaten CSS-Datei gespeichert, die auf Wunsch auch in einem eigenen Ordner gelagert werden kann.

Einbindung der CSS Datei

In der Flex-Anwendung wird die Datei, die als CSS-Datei nun alle Stildefinitionen enthält, im MXML-Tag für die Stildefinition als Quelle (source) angegeben.

```
<mx:Style source="formate.css" />
```

Wenn Sie den Flex Builder 3 nutzen, finden Sie weitere Informationen hierüber im Abschnitt »CSS-Stildefinitionen oder Skins visuell ändern« weiter unten in diesem Kapitel. Die Möglichkeit zur visuellen Arbeit an den CSS-Stildefinitionen bzw. sogar an den festgelegten Skins (dem Aussehen) der eingesetzten Komponenten ist im Flex Builder 3 neu hinzugekommen. Dazu komme ich am Ende des Kapitels.

Optische Zustände

Optische Zustände, sogenannte *states*, unterstützen Sie dabei, an den Vorgaben eines erstellten Layouts festzuhalten und trotzdem schnell und ohne großen Aufwand unterschiedliche Darstellungsweisen innerhalb eines bestimmten Themas zu definieren. Dabei sind es sowohl komplette Layouts als auch einzelne Bestandteile bzw. Bereiche einer entwickelten Anwendung, die man auf der Basis einer Neuordnung von Elementen oder durch Hinzufügen und Entfernen von Bestandteilen verändern kann. Selbst Änderungen von Eigenschaften existierender Elemente und das Ändern von Stilen können eine deutliche optische Änderung des vorhandenen Layouts bewirken bzw. die Funktionalität entscheidend erweitern. Dazu reicht es aus, ein wenig Code zu erzeugen, um die Eigenschaften der verwendeten Bestandteile der aktuellen Ansicht zu verändern, benutzte Elemente zu entfernen oder neue in die Ansicht einzubinden. Dieser Code kann manuell oder im Ansichtsmodus des Flex Builder erzeugt werden.

Zur optischen Unterstützung der Veränderungen der optischen Zustände bietet Flex eine Vielzahl von Effekten an, die den Wechsel zwischen den unterschiedlichen Ansichten durch animierte Übergänge hervorheben.

Der Einsatz von optischen Zuständen ist hinsichtlich Zielsetzung und Funktionalität in verschiedenen Formen und unterschiedlichem Umfang möglich. Betrachtet man den Umfang der Änderungen zwischen zwei voneinander abweichenden optischen Zuständen, sind zwei Varianten unterscheidbar, denen sich die nächsten beiden Abschnitte widmen.

Änderungen am Anwendungsaufbau

Ausgehend von einem Standardaufbau der betroffenen Anwendung beziehen sich die Abweichungen der optischen Zustände auf das komplette Layout. Durch eine Neuordnung der existierenden Boxen und Panels, durch die Änderung ihrer Eigenschaften und der benutzten Stile für die eingesetzten Inhalte oder durch das Entfernen bzw. Hinzufügen von mitgelieferten Flex- oder selbst erstellten Komponenten wird der gesamte optische Zustand verändert.

Abbildung 6-16 zeigt drei verschiedene Anwendungsansichten, bei denen das komplette Layout durch unterschiedlich gestaltete optische Zustände umgesetzt wurde. Achten Sie bitte bei den Abbildungen darauf, dass beim zweiten Zustand nur die Anordnung verändert wurde und in der dritten Ansicht zusätzliche Elemente hinzugefügt wurden. Das kann so weit getrieben werden, dass der komplette Inhalt vollständig gegen einen neuen ausgetauscht wird.

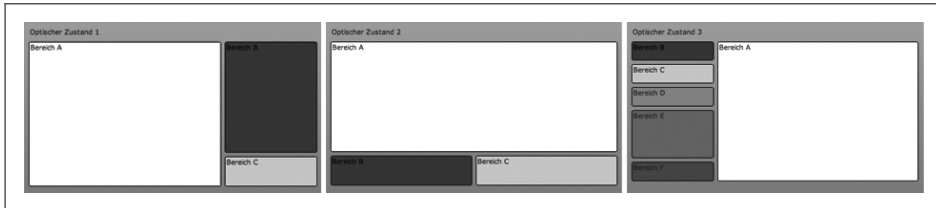


Abbildung 6-16: Drei verschiedene optische Zustände einer Flex-Anwendung

Änderungen innerhalb einer Komponente

Auch in kleinerem Umfang lassen sich die optischen Zustände innerhalb einer Flex-Anwendung effektiv einsetzen. So lassen sich optische Veränderungen als Ergebnis von bestimmten Benutzeraktivitäten innerhalb einer Komponente einsetzen. Das kann man durch das Ergänzen und Entfernen von inhaltlichen Bestandteilen der Komponente oder durch die Zuweisung neuer Stile zu den Bestandteilen oder der gesamten Komponente erreichen. So lassen sich Formulare durch die Interaktion und Auswahl bestimmter Elemente und Thematiken schnell erweitern oder verkleinern, ohne dass die Anwendung komplett verändert werden muss.

In Abbildung 6-17 ist die Änderung der optischen Zustände innerhalb einer Komponente zu sehen. Im Vergleich zu Abbildung 6-16 ändert sich nicht das komplette Layout, sondern nur ein Bestandteil davon.

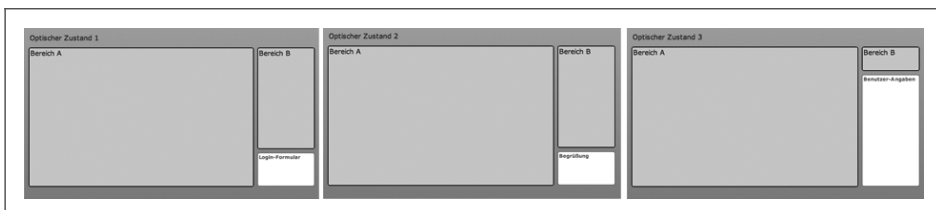


Abbildung 6-17: Zustände, bei denen sich nur einzelne Bestandteile einer Komponente ändern

Der Basiszustand einer Anwendung

Als Ausgangspunkt für die Erstellung unterschiedlicher optischer Zustände wird ein Layout erstellt, das als Basiszustand dient. Dieser Ausgangspunkt ist für die Anlage unterschiedlicher optischer Zustände unbedingt erforderlich, da die neuen Zu-

stände in erster Linie durch Änderungen im Bezug auf den vorher festgelegten Basiszustand erzeugt werden. Statt auf den Basiszustand kann man auch auf einen anderen optischen Zustand verweisen.

Der bisher beschrittene Weg bei der Arbeit mit dem Flex-Framework hat ein Layout für die Anwendung erstellt. Es ist ratsam, dieses Layout auch als optischen Basiszustand zu nutzen. Dieser Zustand ist auch der angezeigte optische Zustand, wenn die Anwendung gestartet wird. Was bei der Arbeit mit optischen Zuständen unerlässlich ist, ist die Zuweisung von IDs, da diese bei den Änderungen innerhalb der einzelnen Zustände unbedingt benötigt werden. Darauf aufbauend werden dann die einzelnen Zustände im Flex Builder angelegt.

Erstellen Sie zunächst am besten eine neue MXML-Application und wählen Sie für das Layout die Option *absolute* aus. Wechseln Sie im Flex Builder in den Source-Modus, falls dieser nicht schon aktiv ist.

Fügen Sie eine Panel-Komponente mit folgenden Eigenschaften ein:

Breite: *600 Pixel*, Höhe: *400 Pixel*, Abstand von links: *10 Pixel*, Abstand von oben: *40 Pixel*, Titel: *Produkt-Details* und ID: *prodDetail*.

```
<mx:Panel width="600" height="400" x="10" y="40" layout="absolute" title="Produkt-Details" id="prodDetail">
</mx:Panel>
```

Wechseln Sie in den Design-Modus des Flex Builder. Öffnen Sie im Bedienfeld *Components* die Liste *Layout*. Wählen Sie dort die Komponente *Panel* aus und ziehen Sie diese in den oberen Bereich rechts neben die schon vorhandene Panel-Komponente. Weisen Sie im Bedienfeld *Flex Properties* für die ausgewählte Komponenten (*mx:Panel*) folgende Werte zu:

Titel: *Zubehör*, Breite: *200 Pixel*, Höhe: *300 Pixel*, Abstand von links: *618 Pixel*, Abstand von oben: *40 Pixel*, ID: *zubehoer*.

Legen Sie eine dritte *Panel*-Komponente mit folgenden Eigenschaften an:

Titel: *Warenkorb*, Breite: *200 Pixel*, Höhe: *92 Pixel*, Abstand von links: *618 Pixel*, Abstand von oben: *348 Pixel*, ID: *warenkorb*. Lassen Sie sich dann die Applikation in der Vorschau anzeigen.

Optische Zustände anlegen

Bei der Nutzung des Flex Builder können Sie die optischen Zustände auf zwei verschiedenen Wegen anlegen. Im Source-Modus wird der MXML-Code in wenigen Schritten und ohne großen Aufwand manuell erzeugt, wenn man den eigentlichen Inhalt vorerst außer Acht lässt. Im Design-Modus steht Ihnen das Bedienfeld *States* zur Erstellungen und Verwaltung der optischen Zustände zur Verfügung.

Auf den folgenden Seiten werden wir unterschiedliche optische Zustände in einer Anwendung erstellen und anschließend die Navigation anlegen.

Einen optischen Zustand manuell anlegen

Die Anlage optischer Zustände erfolgt grundsätzlich in zwei Schritten: In der Application wird vor dem bisherigen Inhalt einmalig ein Bereich für die folgenden Zustände angelegt. Das MXML-Tag dafür ist `<mx:states>`.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:states>
    ...
  </mx:states>
</mx:Application>
```

Für jede optische Ansicht der Anwendung wird ein Bereich für den jeweiligen Zustand definiert, der mit einem Namen versehen wird und anschließend die Änderungen des Layouts und der zugewiesenen Inhalte umschließt. Das MXML-Tag für einen Zustand ist `<mx:State>` inklusive des Namens des Zustandes.

```
<mx:states>
  <mx:State name="zustandB">
  </mx:State>
</mx:states>
```

Für den Standardzustand bzw. das Layout beim Start der Anwendung wird kein Zustand definiert, da dieser automatisch vorliegt, wenn keine optischen Zustände definiert sind.

Öffnen Sie die eben erstellte MXML-Datei. Wechseln Sie in den Source-Modus und ergänzen Sie den Quellcode direkt hinter dem Tag `Application`, um anzugeben, dass neue optische Zustände folgen.

```
<mx:states>
</mx:states>
```

Geben Sie nun zwischen den beiden erstellten `states`-Tags die Definition für einen optischen Zustand an und weisen Sie ihm den Namen *zustandB* zu.

```
<mx:states>
  <mx:State name="zustandB">
  </mx:State>
</mx:states>
```

Einen optischen Zustand visuell anlegen

Die zweite Variante, innerhalb einer Anwendung verschiedene optische Zustände zu erzeugen, ist die im Design-Modus. Arbeiten Sie in diesem Modus, steht Ihnen standardmäßig ein Bedienfeld für die Bearbeitung und Verwaltung der Zustände (*States*) zur Verfügung. Ist das Fenster für die Zustände nicht sichtbar, lässt es sich über das Menü *Window* → *States* öffnen (siehe Abbildung 6-18).

Wie auch bei der manuellen Erstellung des MXML-Codes für optische Zustände erfolgt das visuelle Anlegen eines Zustands in zwei Schritten: der Anlage eines

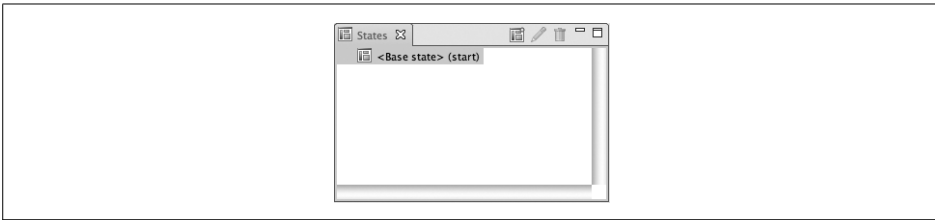


Abbildung 6-18: Die Verwaltung optischer Zustände im Adobe Flex Builder

neuen Zustandes und damit verbunden der Zuweisung der Bezeichnung für diesen optischen Zustand. Das erfolgt nun allerdings visuell, und der Flex Builder übernimmt die Erstellung des MXML-Codes.

Wählen Sie im Bedienfeld *States* die Option *New State*, indem Sie auf das entsprechende Symbol im Kopfbereich des Fensters klicken. Auch bei der visuellen Erstellung von optischen Zuständen müssen eindeutige Bezeichnungen zugewiesen werden. Das geschieht im Dialogfenster *New State* (siehe Abbildung 6-19).

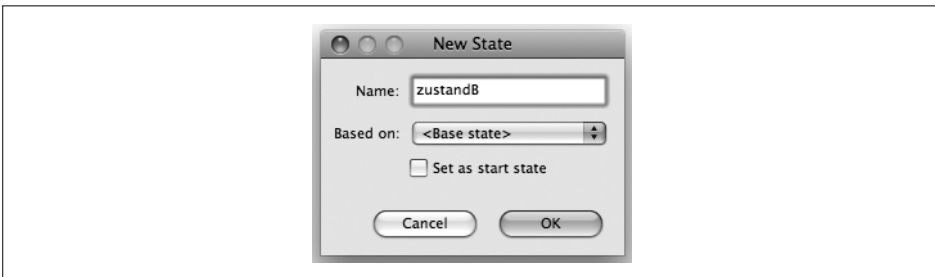


Abbildung 6-19: Das Dialogfenster zum Anlegen eines neuen optischen Zustands

Nun müssen die einzelnen optischen Zustände natürlich auch verändert werden, was in der Anwendung nicht nur das Layout betreffen, sondern auch inhaltliche Veränderungen nach sich ziehen wird. Dazu sind zwei verschiedene Vorgehensweisen möglich. Entweder werden bei den Eigenschaften der eingesetzten Elemente die zugewiesenen Eigenschaften oder Stile verändert. Oder die definierten Komponenten werden aus dem Layout entfernt bzw. neu in das existierende Layout eingebunden. Die Veränderungen innerhalb eines optischen Zustandes erfolgen grundsätzlich im Hinblick auf den Basiszustand. Eigenschaften der existierenden Komponenten werden verändert, indem eine oder mehrere Komponenten entfernt bzw. Komponenten dem optischen Zustand hinzugefügt werden.

Diese Änderungen innerhalb der optischen Zustände können im Source-Modus durch das manuelle Erzeugen von MXML-Code vorgenommen werden oder durch die visuelle Arbeit im Design-Modus, wobei die notwendigen Codefragmente vom Flex Builder eingefügt werden.

Ausgangspunkt für die Änderungen ist der aktuelle optische Zustand der aktuellen Flex-Anwendung, der erstellte Basiszustand. Bevor Änderungen an den einzelnen Zuständen vorgenommen werden, sollte sich der Entwickler sicher in den einzelnen Zuständen orientieren können.

Innerhalb des Source-Modus ist ein optischer Zustand durch das einleitende `<mx:State name="Bezeichnung">` erkennbar. Der Bereich zwischen dem einleitenden *MXML-Tag* und dem allgemeinen abschließenden Tag `</mx:State>` nimmt die gewünschten Änderungen für diesen Zustand auf. Die Erzeugung eines neuen Zustands im Design-Modus bewirkt im Code allerdings nur den Eintrag `<mx:State name="Bezeichnung" />`. Für die manuelle Änderung der Zustände muss diese Eintragung auch manuell verändert werden. Bei Veränderung der optischen Zustände im des Design-Modus erfolgt die diese Änderung automatisch.

Eigenschaften existierender Elemente ändern

Das Ändern der Eigenschaften aller angelegten Elemente bewirkt nach dem Aufruf des optischen Zustands eine Neuordnung innerhalb des Layouts, nämlich Änderungen der Maßangaben der Elemente oder der zugewiesenen Stile, Farben und Schriften. Im Quellcode erfolgt nun die Angabe der Änderungen, die die einzelnen Elemente betreffen. Dafür gibt es das spezielle MXML-Tag `<mx:SetProperty>`. Weiterhin wird als Ziel das Element angegeben, an dem diese Änderung vorgenommen werden soll: `target="{ID des Elements}"`. Anschließend erfolgt die Angabe, welche Eigenschaft man ändern will, indem die Option `name="Eigenschaft"` im MXML-Tag ergänzt wird, und abschließend wird der neue Wert mit `value="Neuer Wert"` zugewiesen.

Die vollständige Änderungsanweisung sieht folgendermaßen aus:

```
<mx:SetProperty target="{ID des Elements}" name="Eigenschaft"
value="Neuer Wert"/>
```

Selbstverständlich müssen auf diesem Weg alle zu ändernden Eigenschaften in einem MXML-Tag angegeben werden. Die änderbaren Eigenschaften hängen bei der Arbeit mit optischen Zuständen natürlich vom jeweiligen Element und seinen möglichen Eigenschaften ab. Für Änderungen der Abmessungen und Position eines Elements sind folgende Eintragungen erforderlich:

```
<mx:SetProperty target="{info}" name="width" value="200"/>
<mx:SetProperty target="{info}" name="height" value="400"/>
<mx:SetProperty target="{info}" name="x" value="500"/>
<mx:SetProperty target="{info}" name="y" value="30"/>
```

Nun werden bei Aufruf dieses Zustands die Eigenschaften des Elements mit der ID `info` auf die angegebenen Werte verändert, unabhängig von den Ausgangseigenschaften. Welche Eigenschaften hier angegeben werden müssen, hängt von den gewünschten Änderungen ab. Im angegebenen Beispiel sind es die Angaben zur

Position mit x und y und die Abmessungen der Komponente width und height. In den meisten Fällen müssen Änderungen für mehrere IDs vorgenommen werden.

Elemente ergänzen und löschen

Eine zweite Form der Änderung von optischen Zuständen besteht im Hinzufügen oder Entfernen von Komponenten oder Inhalten. Bei der Nutzung des Design-Modus sind diese Änderungen durch wenige Handgriffe erledigt, indem der Zustand aus der Liste ausgewählt wird und anschließend Komponenten aus der Anwendung entfernt oder neue hinzugefügt werden; die Arbeit läuft dabei ab wie beim Standardlayout. Der Quellcode dazu wird dann wiederum vom Flex Builder angelegt bzw. angepasst. Für die Arbeit innerhalb des Quellcodes reicht es nicht aus, die Komponenten im jeweiligen State-Block anzulegen. Da sich die einzelnen Zustände immer auf einen anderen Zustand beziehen, erfolgt die Angabe der Änderungen innerhalb der Zustände auch mit der Anweisung, welche Elemente entfernt und welche hinzugefügt werden. Beginnen wir mit dem Entfernen einer Komponente.

Dafür gibt es einen MXML-Tag mit dem Bezeichner `mx:RemoveChild`. Anschließend wird über den Parameter die ID der Komponente angegeben, die Sie entfernen möchten.

```

<mx:State name="zustandB">
  <mx:RemoveChild target="{idDerKomponente}" />
</mx>

```

Auch das Hinzufügen von Komponenten zu einem Zustand wird über ein MXML-Tag erledigt. Hier wird allerdings ein Block mit dem MXML-Tag `mx:AddChild` definiert, der dann alle hinzukommenden Komponenten enthalten wird.

```

<mx:State name="zustandB">
  <mx:RemoveChild target="{idDerKomponente}" />
  <mx:AddChild>
    <mx:Panel title="Das ist ein Beispiel." />
  </mx:AddChild>
</mx>

```

Zustände auf der Basis anderer Zustände

Beim Betrachten des Dialogfensters *New State* sind noch zwei weitere Optionen erkennbar (siehe Abbildung 6-20). Da sich die optischen Zustände grundsätzlich auf andere, schon angelegte Layouts oder Zustände beziehen, besteht selbstverständlich auch die Möglichkeit, den Bezug zu einem Layout bzw. zu einem schon existierenden Zustand herzustellen, um die eigentlichen Änderungen an den unterschiedlichen Zuständen so gering wie möglich zu halten.

Die Zuweisung eines existierenden Layouts ist auch durch das manuelle Erstellen des MXML-Codes möglich. Die Zuweisung erfolgt innerhalb des Einleitungs-Tags



Abbildung 6-20: Im Dialogfeld New State können Sie das Ausgangslayout auswählen.

des optischen Zustands durch die Ergänzung der Option `basedOn="Bezeichnung des Ausgangszustands"`. Wird diese Angabe nicht gemacht, bezieht sich der neu angelegte optische Zustand auf das Standardlayout.

```
<mx:State name="zustandD" basedOn="zustandB">
...
</mx:State>
```

In einzelne Zustände wechseln

Um die ganze Arbeit mit den unterschiedlichen optischen Zuständen abzurunden, ist es nun noch erforderlich, die Aufrufe der einzelnen Zustände umzusetzen. Dieser Vorgang ist einfacher, als Sie vielleicht denken. Dafür gibt es wieder ein bisschen ActionScript. Für die Arbeit mit Zuständen gibt es eine Eigenschaft mit der Bezeichnung `currentState` (»benutzter Zustand«), der noch die Bezeichnung des gewünschten Zustands zugewiesen werden soll. Alles wird wieder über ein Mausereignis aufgerufen. Komplett sieht das Ganze dann so aus:

```
click="currentState='NameDesZustands'"
```

Diese Angabe muss natürlich noch in eine geeignete Komponente platziert werden, wie zum Beispiel einen Button oder direkt ins Panel.

```
<mx:Button label="Thema 2" click="currentState='thema2'" />

<mx:Panel title="kleinShop" click="currentState='shopGross'">
</mx:Panel>
```



Geben Sie keinen optischen Zustand an, wird der Basiszustand aufgerufen. Lassen Sie dazu bei `currentState` den Wert leer: `currentState=''`.

Nun liegt es an Ihnen, wie Sie mit diesen optischen Zuständen arbeiten und welche Komponenten Sie für den Mausklick auswählen. Weitere Möglichkeiten zum Einsatz der Interaktion werden Sie in den folgenden Kapiteln kennenlernen.

Übung 9: Optische Zustände in der Galerie FLEXibel

Es sollen nun zwei optische Zustände für die aktuelle Ausstellung und den Shop in der Galerie FLEXibel angelegt und per Mausklick verwaltet werden. Das Ziel ist, den großen Bereich, der bei der Startseite durch die vier verschiedenen Bildangebote aufgefüllt ist, durch jeweils ein Panel zu ersetzen. Der Wechsel in die optischen Zustände soll per Mausklick auf die kleinen Panels erfolgen.

- 1 Öffnen Sie die Flex-Anwendung *ue_08.xml* und speichern Sie sie unter dem neuen Dateinamen *ue_09.xml*.
- 2 Wechseln Sie in den Design-Modus, wenn Sie sich nicht schon darin befinden.
- 3 Gehen Sie ins Bedienfeld *States* und klicken Sie auf das Symbol für die Anlage eines neuen optischen Zustand (*New State*). Tragen Sie als Namen *stateAusstellung* ein.
- 4 Wechseln Sie in den Source-Modus und ergänzen Sie ein weiteres *State*-Tag im Block für die *mx:States*. Weisen Sie ihm den Namen *stateShop* zu

```
<mx:states>
  <mx:State name="stateAusstellung">
  </mx:State>
  <mx:State name="stateShop">
  </mx:State>
</mx:states>
```

- 5 Wählen Sie im Bedienfeld *Outline* die unterste *HBox* aus und weisen Sie ihr die ID *hbox1* zu.

```
<mx:HBox width="590" height="245" x="0" y="0" id="hbox1">
```
- 6 Verfahren Sie ebenso mit der darüberliegenden *HBox* und nennen Sie die ID *hbox2*.

```
<mx:HBox width="590" height="245" x="0" y="0" id="hbox2">
```
- 7 In den optischen Zuständen werden nun die Inhalte entfernt, die nicht mehr benötigt, sondern ersetzt werden sollen. Dafür sind die *HBox*-Komponenten gut geeignet.

```
<mx:State name="stateAusstellung">
  <mx:RemoveChild target="{hbox1}"/>
  <mx:RemoveChild target="{hbox2}"/>
</mx:State>
<mx:State name="stateShop">
  <mx:RemoveChild target="{hbox1}"/>
  <mx:RemoveChild target="{hbox2}"/>
</mx:State>
```

- 8 Wechseln Sie in den Design-Modus und schalten Sie im Bedienfeld *States* zwischen den unterschiedlichen Zuständen hin und her.
- 9 Wählen Sie nun den Zustand *stateAusstellung*. Ziehen Sie eine *Panel*-Komponente in den nun leeren Bereich und passen Sie das Panel in seinen Eigenschaften an.

Breite: 590

Höhe: 500

Titel: Die Ausstellung

Schriftfarbe: FF9900 (orange)

ID: shopGross

Übung 9: Optische Zustände in der Galerie FLEXibel (Fortsetzung)

- 10 Ergänzen Sie nun noch den optischen Zustand `stateShop` durch das Hinzufügen eines Panels. Wechseln Sie dazu in den Source-Modus und ergänzen Sie im Quellcode den Block für den Zustand `stateShop`.

```
<mx:State name="stateShop">
  <mx:RemoveChild target="{hbox1}"/>
  <mx:RemoveChild target="{hbox2}"/>
  <mx:AddChild relativeTo="{vbox1}" position="lastChild">
    <mx:Panel width="590" height="500" layout="absolute"
      title="Der Shop" id="shopGross" color="#FF9900">
      </mx:Panel>
  </mx:AddChild>
</mx:State>
```

- 11 Ergänzen Sie nun noch die Übersichtspネル im Basiszustand mit den `click`-Zuweisungen. Diese Arbeiten lassen sich in der vorliegenden Version des Flex Builder nur im Source-Modus ausführen.

Wählen Sie dazu das erste Panel in der obersten VBox und ergänzen innerhalb des Tags das Verhalten.

```
<mx:Panel width="200" height="165" layout="absolute" title="Aktuelle
Ausstellung" color="#FF9900" id="panelAusstellung"
click="currentState='stateAusstellung'">
```

- 12 Wählen Sie dazu das zweite Panel in der obersten VBox und ergänzen innerhalb des Tags das Verhalten.

```
<mx:Panel width="200" height="250" layout="absolute" title="Neu im Galerie-
Shop" color="#FF9900" id="panelShop" click="currentState='stateShop'" >
```

- 13 Abschließend legen wir noch fest, dass bei einem Klick auf das Logo der Galerie zurück zum optischen Ausgangszustand gewechselt wird.

```
<mx:Image x="490.95" y="0.9" source="../assets/images/elemente/galerieLogo.
png" click="currentState=''" />
```

- 14 Starten Sie die Anwendung und klicken Sie anschließend auf die beiden Panels und das Logo der Anwendung.

Dies ist eine Auszug aus dem Buch "Praxiswissen Flex 3", ISBN 978-3-89721-860-4
http://www.oreilly.de/catalog/lex3basger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2009

CSS-Stildefinitionen und Skins visuell ändern

Wie schon im Abschnitt »Mit Cascading Style Sheets arbeiten« erwähnt, folgt nun ein Abschnitt, der sich ausschließlich an Anwender des Flex Builder 3 richtet. Selbstverständlich können auch alle anderen Leser von diesem Teil profitieren, denn Stildefinitionen lassen sich auch manuell einfach ändern, wie die festgelegten *Skins* (die optische Gestaltung der Komponenten) innerhalb der einzelnen Klassen.

Legen Sie innerhalb Ihres Testprojektes eine neue CSS-Datei an (siehe Abbildung 6-21). Diese Datei enthält vorerst keine Stildefinition, was zur Folge hat, dass im Pop-up-Menü *Style* im Design-Modus keine Stile aufgelistet werden; später wird das Pop-up-Menü hinter *Style* die Komponenten und CSS-Klassen auflisten, für die Stildefinitionen vorgenommen worden sind.

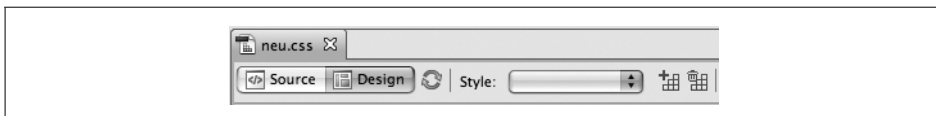


Abbildung 6-21: Die veränderte Kopfleiste des Arbeitsbereichs bei geöffneter CSS-Datei

Hinter dem Pop-up befinden sich zwei ähnlich aussehende Symbole: Der Papierkorb dient zum Löschen der angelegten Stildefinition, unabhängig davon, welche Form der Stildefinition vorliegt und welchen Umfang sie aufweist. Das zweite Symbol mit dem Plus (+) dient zur Anlage einer neuen Stildefinition. Nach Auswahl dieses Symbols müssen Sie noch festlegen, welche Form von Stildefinition angelegt werden soll, nämlich ob sie global sein soll oder eine selbst angelegte Klasse bzw. eine Komponente. Die Komponente können Sie anschließend noch aus dem Pop-up auswählen, da dort alle Komponenten enthalten sind. Auch hier besteht nochmals die Möglichkeit, eine Kombination zwischen Komponente und selbst benannter Klasse zu wählen. Am Beispiel der Button-Komponente sollen die folgenden Schritte näher erläutert werden (siehe Abbildung 6-22).



Abbildung 6-22: Im Dialogfeld *New Style* wird der Typ der Stildefinition festgelegt.

Nun werden Sie sehen, warum ich von diesem Werkzeug des Flex Builders so fasziniert bin. Die aktuell für die Standardeinstellung der Komponente *Button* vordefinierten Formatierungen werden angezeigt, und zwar in allen möglichen Erscheinungsformen, die innerhalb einer Flex-Anwendung auftreten können (siehe Abbildung 6-23). Einfach genial, oder? Sicherlich werden Sie nun erst einmal die Bezeichnungen begutachten. Sie stehen für die möglichen Zustände eines Buttons innerhalb der Anwendung. Werden keine Änderungen vorgenommen, sind die angezeigten Eigenschaften für die jeweiligen Einstellungen und Mausereignisse (Berühren, Maustaste drücken, Maustaste loslassen) gültig.

Nun muss man noch die entsprechenden Änderungen an den Komponenten vornehmen. Dazu können Sie wieder das Bedienfeld *Flex Properties* nutzen und dabei ihren kreativen Ideen freien Lauf lassen. Sie sehen die Auswirkungen Ihrer Änderungen sofort im Anzeigefenster (siehe Abbildung 6-24).

Aber Achtung! Im oberen Bereich des Bedienfeldes gibt es zwei ziemlich unscheinbare Buttons: *Style* und *Skin*. *Style* legt innerhalb des Projektes (also der Anwendung oder CSS-Datei) die Definition des optischen Aufbaus einer Komponente an,



Abbildung 6-23: Die Standarderscheinungsbilder der Komponente Button

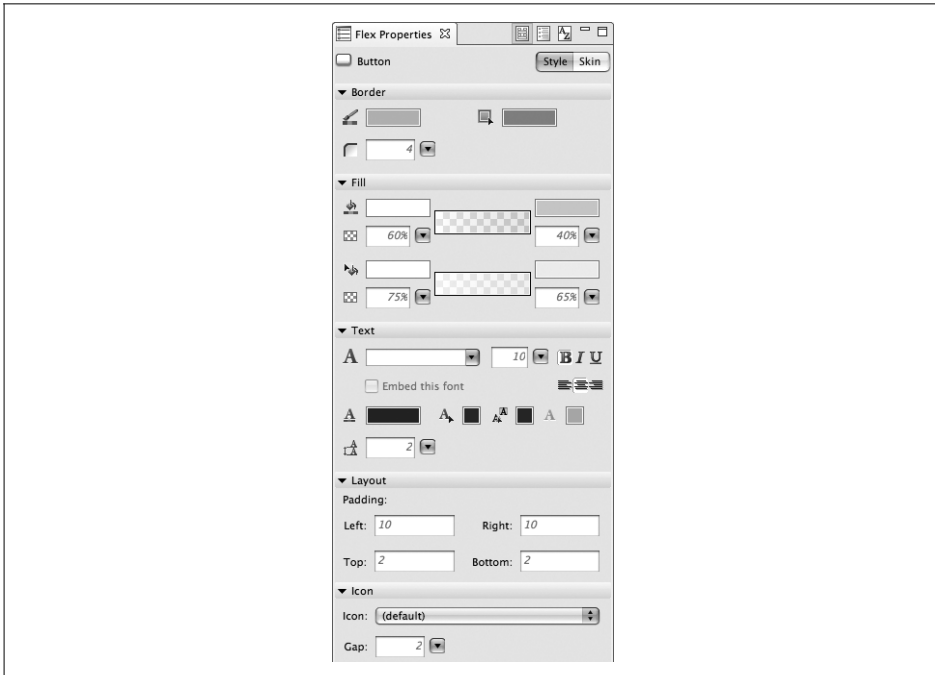


Abbildung 6-24: Das Bedienfeld Flex Properties im Einsatz zur Erstellung eines Stils

wobei der ursprüngliche Aufbau erhalten bleibt. *Skin* ändert den kompletten Aufbau der Komponente in seiner ActionScript-Klasse, d.h. die gemachten Veränderungen haben nach ihrer Bestätigung Auswirkungen auf die Standardeigenschaften der Komponente.

Damit liegt ein komfortables visuelles Werkzeug zur Arbeit an innerhalb des Flex Builder vor, was vielen Designern mit Sicherheit entgegenkommen wird.



Sie können auch innerhalb der Anwendung eine Formatierung an der Komponente vornehmen und diese anschließend in den *Flex Properties* über den Button *Convert to CSS* in eine CSS-Definition auslagern. Das funktioniert allerdings nur in einem neuen Projekt und wenn eine solche Datei existiert.