

# Korzystanie ze składników języka ACTIONSCRIPT® 3.0

## **Informacje prawne**

Informacje prawne można znaleźć na stronie [http://help.adobe.com/pl\\_PL/legalnotices/index.html](http://help.adobe.com/pl_PL/legalnotices/index.html).

# Spis treści

## Rozdział 1: Wprowadzenie

Przeznaczenie dokumentacji .....	1
Wymagania systemowe .....	1
Informacje o dokumentacji .....	2
Konwencje typograficzne .....	2
Terminy używane w tym podręczniku .....	2
Zasoby dodatkowe .....	2

## Rozdział 2: Informacje o składnikach języka ActionScript 3.0

Korzyści wynikające ze stosowania składników .....	4
Typy składników .....	6
Dodawanie lub usuwanie z dokumentu .....	7
Wyszukiwanie wersji składnika .....	9
Model obsługi zdarzeń języka ActionScript 3.0 .....	10
Prosta aplikacja .....	11

## Rozdział 3: Praca ze składnikami

Architektura składników .....	18
Praca z plikami składników .....	20
Debugowanie aplikacji ze składnikami .....	22
Ustawianie parametrów i właściwości .....	23
Biblioteka .....	24
Określanie rozmiaru składników .....	24
Podgląd aktywny .....	25
Obsługa zdarzeń .....	25
Praca z listą wyświetlania .....	27
Praca z klasą FocusManager .....	29
Praca ze składnikami opartymi na liście .....	31
Praca z dostawcą danych .....	31
Praca z mechanizmem renderującym komórki .....	39
Ułatwienia dostępu do składników .....	46

## Rozdział 4: Korzystanie ze składników interfejsu użytkownika

Korzystanie ze składnika Button .....	47
Korzystanie ze składnika CheckBox .....	50
Korzystanie ze składnika ColorPicker .....	53
Korzystanie ze składnika ComboBox .....	55
Korzystanie ze składnika DataGrid .....	58
Korzystanie ze składnika Label .....	64
Korzystanie ze składnika List .....	66
Korzystanie ze składnika NumericStepper .....	71
Korzystanie ze składnika ProgressBar .....	74
Korzystanie ze składnika RadioButton .....	79

**Spis treści**

Korzystanie ze składnika ScrollPane .....	82
Korzystanie ze składnika Slider .....	85
Korzystanie ze składnika TextArea .....	88
Korzystanie ze składnika TextInput .....	91
Korzystanie ze składnika TileList .....	94
Korzystanie ze składnika UILoader .....	97
Korzystanie ze składnika UIScrollBar .....	99
<b>Rozdział 5: Dostosowywanie składników interfejsu użytkownika</b>	
Informacje o dostosowywaniu składnika interfejsu użytkownika .....	102
Ustawianie stylów .....	102
Informacje o karnacjach .....	105
Dostosowywanie składnika Button .....	108
Dostosowywanie składnika CheckBox .....	110
Dostosowywanie składnika ColorPicker .....	111
Dostosowywanie składnika ComboBox .....	113
Dostosowywanie składnika DataGrid .....	115
Dostosowywanie składnika Label .....	120
Dostosowywanie składnika List .....	121
Dostosowywanie składnika NumericStepper .....	124
Dostosowywanie składnika ProgressBar .....	125
Dostosowywanie składnika RadioButton .....	127
Dostosowywanie składnika ScrollPane .....	129
Dostosowywanie składnika Slider .....	129
Dostosowywanie składnika TextArea .....	131
Dostosowywanie składnika TextInput .....	133
Dostosowywanie składnika TileList .....	134
Dostosowywanie składnika UILoader .....	136
Dostosowywanie składnika UIScrollBar .....	136
<b>Rozdział 6: Korzystanie ze składnika FLVPlayback</b>	
Użycie składnika FLVPlayback .....	139
Dostosowywanie składnika FLVPlayback .....	158
Korzystanie z pliku SMIL .....	169
<b>Rozdział 7: Korzystanie ze składnika FLVPlaybackCaptioning</b>	
Korzystanie ze składnika FLVPlaybackCaptioning .....	177
Korzystanie z podpisów Timed Text .....	179
Korzystanie z punktów sygnalizacji razem z podpisami .....	186
Odtwarzanie wielu plików FLV z podpisami .....	188
Dostosowywanie składnika FLVPlaybackCaptioning .....	189

# Rozdział 1: Wprowadzenie

Program Adobe® Flash® CS5 Professional jest standardowym narzędziem służącym do tworzenia atrakcyjnych, dynamicznych materiałów internetowych. Składniki są elementami konstrukcyjnymi zaawansowanych aplikacji internetowych, które udostępniają użytkownikom te treści. *Składnik* jest to klip filmowy z parametrami, które umożliwiają dostosowanie składnika podczas tworzenia treści w programie Flash albo z wykorzystaniem metod, właściwości i zdarzeń w czasie wykonywania programu w języku Adobe® ActionScript®. Składniki zostały zaprojektowane w taki sposób, aby umożliwiły programistom wielokrotne użycie tych samych fragmentów kodu oraz współużytkowanie kodu. Ponadto składniki są kontenerami zawierającymi złożoną funkcjonalność, którą projektanci mogą wykorzystywać i dostosowywać bez użycia języka ActionScript.

Składniki umożliwiają łatwe i szybkie budowanie stabilnych aplikacji Flash działających w sposób spójny i wyposażonych w jednolity interfejs użytkownika. W niniejszym podręczniku opisano sposób budowania aplikacji z wykorzystaniem składników języka Adobe ActionScript 3.0. *Skorowidz języka i składników Adobe® ActionScript® 3.0* zawiera opis interfejsów programowania aplikacji (API, ang. application programming interface) każdego ze składników.

Możliwe jest korzystanie ze składników opracowanych przez firmę Adobe®, pobieranie składników utworzonych przez innych programistów oraz tworzenie własnych składników.

## Przeznaczenie dokumentacji

Niniejszy podręcznik jest przeznaczony dla programistów, którzy budują aplikacje w środowisku Flash i chcą korzystać ze składników w celu przyspieszenia prac nad aplikacjami. Czytelnicy powinni być zaznajomieni z zasadami tworzenia aplikacji w języku Flash i pisania skryptów w języku ActionScript.

Użytkownicy z ograniczonym doświadczeniem w pisaniu skryptów w języku ActionScript mogą dodawać składniki do dokumentu, ustawiać ich parametry w Inspektorze właściwości lub Inspektorze składników i używać panelu Zachowania do obsługi zdarzeń związanych z tymi składnikami. Na przykład można skojarzyć zachowanie Przejdź do strony sieci Web ze składnikiem Przycisk, który po kliknięciu będzie otwierał adres URL w przeglądarce. Takie skojarzenie da się zrealizować bez pisania kodu w języku ActionScript.

Programiści zainteresowani tworzeniem bardziej zaawansowanych aplikacji mogą tworzyć składniki dynamicznie, ustawiać ich właściwości z poziomu kodu ActionScript i wywoływać metody w czasie wykonywania, zaś zdarzenia obsługiwać przy użyciu modelu detektorów zdarzeń.

Więcej informacji zawiera sekcja „[Praca ze składnikami](#)” na stronie 18.

## Wymagania systemowe

Składniki Flash nie mają żadnych wymagań systemowych wykraczających poza wymagania systemowe samego programu Flash.

Pliki SWF, w których używane są składniki z programu Flash CS3 lub nowszej wersji, muszą być odtwarzane za pomocą odtwarzacza Adobe® Flash® Player 9.0.28.0 lub nowszej wersji oraz muszą być publikowane dla wersji języka ActionScript 3.0 (odpowiednie ustawienie można wybrać po wywołaniu polecenia Plik > Ustawienia publikowania, na karcie Flash).

## Informacje o dokumentacji

W niniejszym dokumencie szczegółowo omówiono korzystanie ze składników przy tworzeniu aplikacji Flash. Przyjęto założenie, że użytkownik dysponuje ogólną wiedzą o programie Flash i języku ActionScript 3.0. Dostępna jest odrębna dokumentacja poświęcona programowi Flash i produktom pokrewnym.

Niniejszy dokument jest dostępny w postaci pliku PDF i jako pomoc online. Aby wyświetlić pomoc online, należy uruchomić program Flash i wybrać kolejno polecenia Pomoc > Pomoc programu Flash > Korzystanie ze składników języka Adobe ActionScript 3.0.

Informacje na temat programu Flash można znaleźć w następujących dokumentach:

- *Korzystanie z programu Flash*
- *ActionScript 3.0 — Podręcznik dla programistów*
- *Skorowidz języka Adobe ActionScript 3.0 dla programu Flash Professional*

## Konwencje typograficzne

W niniejszym podręczniku używane są następujące konwencje typograficzne:

- *Kursywa* oznacza wartość, którą należy zastąpić (np. w ścieżce do folderu).
- *Czcionka kodu* oznacza kod w języku ActionScript, w tym nazwy metod i właściwości.
- *Czcionka kodu + kursywa* oznacza element, który należy zastąpić (np. parametr w języku ActionScript).
- **Czcionka pogrubiona** oznacza wartość wprowadzaną przez użytkownika.

## Terminy używane w tym podręczniku

W podręczniku używane są następujące terminy:

**w czasie wykonywania** Gdy kod jest wykonywany w odtwarzaczu Flash Player.

**podczas tworzenia treści** Gdy użytkownik pracuje w środowisku programu Flash do tworzenia treści.

## Zasoby dodatkowe

Oprócz wymienionych podręczników firma Adobe udostępnia regularnie aktualizowane artykuły, pomysły i koncepcje projektowe oraz przykłady w serwisach Adobe Developer Center oraz Adobe Design Center.

Dodatkowe przykłady składników można znaleźć pod adresem [www.adobe.com/go/learn\\_fl\\_samples\\_pl](http://www.adobe.com/go/learn_fl_samples_pl).

### Centrum programistów Adobe

Centrum programistów Adobe to źródło najbardziej aktualnych informacji o języku ActionScript, artykułów na temat tworzenia rzeczywistych aplikacji oraz informacji o ważnych, nowych zagadnieniach i problemach. Serwis Developer Center jest dostępny pod adresem [www.adobe.com/go/flash\\_devcenter\\_pl](http://www.adobe.com/go/flash_devcenter_pl).

**Wprowadzenie****Centrum projektowania Adobe**

Serwis ten pozwala poznać najnowsze trendy w dziedzinie grafiki i animacji cyfrowej. Zawiera prace czołowych artystów, nowe koncepcje i oferuje możliwość doskonalenia umiejętności przy użyciu samouczków, szczególnie opisanych procesów i zaawansowanych technik. Samouczki, artykuły i inspirujące galerie są aktualizowane dwa razy w miesiącu. Serwis Design Center jest dostępny pod adresem [www.adobe.com/go/fl\\_designcenter\\_pl](http://www.adobe.com/go/fl_designcenter_pl).

# Rozdział 2: Informacje o składnikach języka ActionScript 3.0

Składniki oprogramowania Adobe® Flash® Professional CS5 są klipami filmowymi z parametrami, które umożliwiają użytkownikowi modyfikowanie ich wyglądu i działania. Składnik może być prostym elementem sterowania na interfejsie użytkownika, takim jak RadioButton, CheckBox lub może zawierać treść, taką jak obszar Lista albo DataGrid.

Składniki umożliwiają łatwe i szybkie budowanie stabilnych aplikacji Flash w sposób zapewniających ich spójne działanie i wygląd. Zamiast tworzyć niestandardowe przyciski, pola złożone i listy użytkownik może wykorzystać składniki Flash, które implementują takie elementy sterowania. Wystarczy je przeciągnąć z panelu Składniki do dokumentu aplikacji. Wygląd i działanie tych składników można łatwo dostosować, aby dopasować je do potrzeb aplikacji.

Wszystkie te czynności można wykonać bez zaawansowanej znajomości języka ActionScript, ale możliwe jest również korzystanie z języka ActionScript 3.0 w celu modyfikowania działania składnika lub w celu zaimplementowania nowego działania. Dla każdego składnika istnieje unikalny zestaw metod, właściwości i zdarzeń ActionScript, które razem tworzą *aplikacyjny interfejs programistyczny* (API) składnika. Interfejs API umożliwia tworzenie składników i manipulowanie nimi podczas działania aplikacji.

Interfejs API umożliwia również tworzenie nowych, niestandardowych składników. Składniki utworzone przez członków społeczności Flash można pobrać z witryny Adobe Exchange, pod adresem [http://www.adobe.com/go/flash\\_exchange\\_pl](http://www.adobe.com/go/flash_exchange_pl). Informacje na temat tworzenia składników można znaleźć pod adresem [http://www.adobe.com/go/learn\\_fl\\_creating\\_components\\_pl](http://www.adobe.com/go/learn_fl_creating_components_pl).

Architektura składników języka ActionScript 3.0 zawiera klasy bazowe dla wszystkich składników, karnacje i style (które umożliwiają dostosowywanie wyglądu), model obsługi zdarzeń, funkcje zarządzania obszarem aktywnym, interfejs zapewniania dostępu i inne.

**Uwaga:** Produkt Adobe Flash CS5 zawiera składniki języka ActionScript 2.0 oraz składniki języka ActionScript 3.0. Tych dwóch zestawów składników nie można mieszać. W danej aplikacji należy korzystać z jednego lub drugiego zestawu. Program Flash CS5 prezentuje składniki ActionScript 2.0 lub składniki ActionScript 3.0 — jest to uzależnione od tego, czy użytkownik otworzył plik w języku ActionScript 2.0 czy plik ActionScript 3.0. Podczas tworzenia nowego dokumentu Flash należy określić plik Flash (ActionScript 3.0) lub plik Flash (ActionScript 2.0). Podczas otwierania istniejącego dokumentu program Flash sprawdza ustawienia publikacji w celu określenia zestawu składników, jaki powinien być używany. Informacje na temat korzystania ze składników języka ActionScript 2.0 zawiera podręcznik *Korzystanie ze składników języka Adobe® ActionScript® 2.0*.

Pełną listę składników Flash ActionScript 3.0 zawiera sekcja „[Typy składników](#)” na stronie 6.

## Korzyści wynikające ze stosowania składników

Składniki umożliwiają oddzielenie procesu projektowania aplikacji od procesu kodowania. Umożliwiają programistom tworzenie funkcji, z których projektanci mogą korzystać w aplikacjach. Programiści mogą ubudowywać często używane funkcje w składniki, a projektanci mogą dostosowywać wielkość, lokalizację i działanie składników poprzez zmianę ich parametrów. Mogą również zmieniać wygląd składników poprzez edycję elementów graficznych lub karnacji.



Składniki współkorzystają z najważniejszych funkcji, takich jak style, karnacje i funkcje zarządzanie obszarami aktywnymi. Po dodaniu pierwszego składnika do aplikacji te główne funkcje zajmują około 20 kilobajtów pamięci. W miarę dodawania innych składników wstępnie przydzielone miejsce w pamięci jest współużytkowane przez dodane składniki, dzięki czemu wielkość aplikacji jest mniejsza.

W niniejszej sekcji przedstawiono niektóre zalety korzystania ze składników ActionScript 3.0.

**Potęga języka ActionScript 3.0** — jest to wydajny, obiektowy język programowania, który jest istotnym krokiem na drodze ewolucji możliwości programu Flash Player. Język jest przeznaczony do budowania bogatych w środki wyrazu aplikacji Internetowych na bazie wielokrotnie wykorzystywanego kodu. Język ActionScript 3.0 jest oparty na międzynarodowej specyfikacji języka skryptowego ECMAScript oraz zgodny ze specyfikacją języka ECMAScript (ECMA-262) edition 3. Szczegółowe wprowadzenie do języka ActionScript 3.0 zawiera publikacja *ActionScript 3.0 — podręcznik dla programistów*. Więcej informacji na temat języka zawiera [Skorowidz języka ActionScript 3.0](#).

**Składniki interfejsu użytkownika w formacie FLA** — zapewniają łatwy dostęp do karnacji w celu szybkiego dostosowywania podczas tworzenia. Te składniki udostępniają również style — między innymi style karnacji, które umożliwiają dostosowywanie aspektów wyglądu składników i ładowanie karnacji w czasie wykonywania. Więcej informacji zawiera sekcja „[Dostosowywanie składników interfejsu użytkownika](#)” na stronie 102 oraz [Skorowidz języka ActionScript 3.0](#).

**Nowy składnik FVLPlayback dodaje składnik FLVPlaybackCaptioning** wraz z funkcjami obsługi pełnego ekranu; udoskonalonym podglądem na żywo; karnacjami, które umożliwiają dodawanie kolorów i ustawień alfa; a także udoskonalone funkcje pobierania FLV i opcje definiowania układów.

**Inspektor właściwości i inspektor składników** — umożliwiają zmianę parametrów składników podczas tworzenia w programie Flash. Więcej informacji zawierają sekcje „[Praca z plikami składników](#)” na stronie 20 i „[Ustawianie parametrów i właściwości](#)” na stronie 23.

**Nowe okno dialogowe kolekcji** — przeznaczone dla składników ComboBox, List i TileList — umożliwia określanie wartości właściwości `dataProvider` za pośrednictwem interfejsu użytkownika. Więcej informacji zawiera sekcja „[Tworzenie dostawcy danych](#)” na stronie 31.

**Model zdarzeń języka ActionScript 3.0** — dzięki niemu aplikacje mogą nasłuchiwać zdarzeń i wywoływać programy obsługi zdarzeń w celu odpowiadania na te zdarzenia. Więcej informacji zawierają sekcje „[Model obsługi zdarzeń języka ActionScript 3.0](#)” na stronie 10 i „[Obsługa zdarzeń](#)” na stronie 25.

**Klasy Manager** — stanowią łatwy sposób obsługi obszarów aktywnych i zarządzania stylami w aplikacji. Więcej informacji zawiera [Skorowidz języka ActionScript 3.0](#).

**Klasa podstawowa UIComponent** — udostępnia podstawowe metody, właściwości i zdarzenia dla składników stanowiących jej rozszerzenie. Wszystkie składniki interfejsu użytkownika ActionScript 3.0 dziedziczą z klasy UIComponent. Więcej informacji zawiera opis klasy UIComponent w [Skorowidzu języka ActionScript 3.0](#).

**Korzystanie z SWC** w składnikach interfejsu użytkownika w formacie FLA udostępnia definicje ActionScript jako zasoby wewnątrz osi czasu składnika, co znacznie przyspiesza kompilację.

**Łatwo rozszerzalna hierarchia klas** — za pomocą języka ActionScript 3.0 można tworzyć unikalne przestrzenie nazw, można importować klasy i podklasy w razie potrzeby w celu rozszerzenia składników.

Więcej informacji zawiera [Skorowidz języka ActionScript 3.0](#).

**Uwaga:** Program Flash CS5 obsługuje składniki w formacie FLA oraz w formacie SWC. Więcej informacji zawiera sekcja „[Architektura składników](#)” na stronie 18.

## Typy składników

Składniki Flash są instalowane podczas instalowania programu Flash CS5.

Do składników języka ActionScript 3.0 należą następujące składniki interfejsu użytkownika:

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

Oprócz składników interfejsu użytkownika język Flash ActionScript 3.0 zawiera również następujące składniki i klasy obsługi:

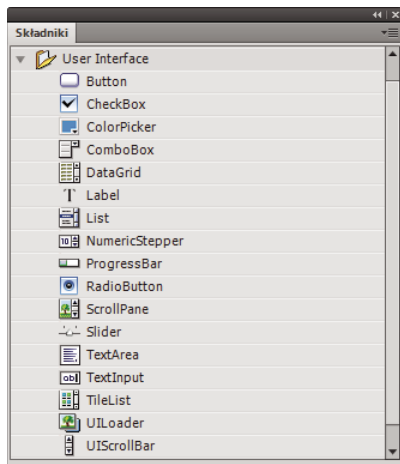
- Składnik FLVPlayback (fl.video.FLVPlayback), który jest składnikiem w formacie SWC.  
Składnik FLVPlayback umożliwia łatwe dołączanie odtwarzacza wideo do aplikacji Flash w celu odtworzenia progresywnego wideo strumieniowego przez HTTP z usługi Adobe® Flash® Video Streaming Service (FVSS) lub z serwera Adobe Macromedia® Flash® Media Server (FMS). Więcej informacji zawiera sekcja „[Korzystanie ze składnika FLVPlayback](#)” na stronie 139.
- Składniki FLVPlayback Custom UI są w formacie FLA i współpracują ze składnikami FLVPlayback w wersjach ActionScript 2.0 i ActionScript 3.0. Więcej informacji zawiera sekcja „[Korzystanie ze składnika FLVPlayback](#)” na stronie 139.
- Składnik FLVPlayback Captioning, który umożliwia umieszczanie podpisów dla składnika FLVPlayback. Patrz „[Korzystanie ze składnika FLVPlaybackCaptioning](#)” na stronie 177.

Pełną listę składników języka ActionScript 3.0 oraz klas, które je obsługują, zawiera [Skorowidz języka ActionScript 3.0](#).

### Wyświetlanie składników Flash:

Składniki języka Flash ActionScript 3.0 można wyświetlać na panelu Składniki — w tym celu należy wykonać następujące czynności.

- 1 Uruchom program Flash.
- 2 Utwórz nowy plik Flash (ActionScript 3.0) lub otwórz istniejący dokument Flash, w którym Ustawienia publikowania określają język ActionScript 3.0.
- 3 Wybierz opcje Okno > Składniki, aby otworzyć panel składników, jeśli nie został jeszcze otwarty.



Panel Składniki ze składnikami interfejsu użytkownika

Dodatkowe składniki można również pobrać z witryny Adobe Exchange pod adresem [http://www.adobe.com/go/flash\\_exchange\\_pl](http://www.adobe.com/go/flash_exchange_pl). W celu zainstalowania składników pobranych z witryny Exchange należy pobrać narzędzie Adobe® Extension Manager z witryny [http://www.adobe.com/go/exchange\\_pl](http://www.adobe.com/go/exchange_pl), a następnie zainstalować je. Kliknij odsyłacz Adobe Exchange Home i poszukaj odsyłacza do Extension Manager.

Na panelu Składniki w programie Flash może być widoczny każdy składnik. W celu zainstalowania składników na komputerze Windows® lub Macintosh® wykonaj następujące czynności.

#### Instalowanie składników na komputerze z systemem Windows lub Macintosh:

- 1 Zamknij program Flash.
- 2 Umieść plik SWC lub FLA zawierający składnik w następującym folderze na dysku twardym:
  - W systemie Windows:  
C:\Program Files\Adobe\Adobe Flash CS5\język\Configuration\Components
  - W systemie Macintosh:  
Macintosh HD:Programy:Adobe Flash CS5:Configuration:Components
- 3 Uruchom program Flash.
- 4 Wybierz opcje Okno > Składniki, aby wyświetlić składnik na panelu Składniki, jeśli ten panel nie został jeszcze otwarty.

Więcej informacji o plikach składników zawiera sekcja „Praca z plikami składników” na stronie 20

## Dodawanie lub usuwanie z dokumentu

Po przeciągnięciu składnika w formacie FLA z panelu Składniki do stołu montażowego program Flash importuje edytowalny klip filmu do biblioteki. Po przeciągnięciu składnika w formacie SWC do stołu montażowego program Flash importuje do biblioteki skompilowany klip. Po zaimportowaniu do biblioteki składnika można przeciągać jego instancje do stołu montażowego z panelu Biblioteka lub panelu Składniki.

## Dodawanie składników podczas tworzenia

W celu dodania składnika do dokumentu można przeciągnąć go z panelu Składniki. Właściwości każdej instancji składnika można ustawić w inspektorze właściwości lub na karcie Parametry w inspektorze składników.

- 1 Wybierz polecenia Okno > Składniki.
- 2 Kliknij dwukrotnie składnik w panelu Składniki lub przeciągnij składnik do obszaru stołu montażowego.
- 3 Wybierz składnik na stole montażowym.
- 4 Jeżeli Inspektor właściwości jest niewidoczny wybierz Okno > Właściwości > Właściwości.
- 5 W inspektorze właściwości wprowadź nazwę dla instancji składnika.
- 6 Wybierz opcje Okno > Inspektor składników, a następnie wybierz kartę Parametry, aby określić parametry dla instancji.

Więcej informacji zawiera sekcja „[Ustawianie parametrów i właściwości](#)” na stronie 23.

- 7 Zmień wielkość komponentu zgodnie z wymaganiami, edytując wartości dotyczące szerokości (W:) i wysokości (H:).

Więcej informacji na temat określania wielkości określonych typów składników zawiera sekcja „[Dostosowywanie składników interfejsu użytkownika](#)” na stronie 102.

- 8 Wybierz opcje Sterowanie > Testuj film albo naciśnij klawisze Control+Enter, aby skompilować dokument i wyświetlić rezultaty zastosowanych ustawień.

Kolor i format tekstu można również zmienić formatując składnik poprzez ustawienie właściwości stylu lub dostosowując jego wygląd poprzez edycję karnacji składnika. Więcej informacji na te tematy zawiera sekcja „[Dostosowywanie składników interfejsu użytkownika](#)” na stronie 102.

Jeśli podczas tworzenia składnik zostanie przeciągnięty na stół montażowy, do składnika można się odwołać, korzystając z nazwy jego instancji (na przykład: `moj_przycisk`).

## Dodawanie składników w czasie wykonywania za pomocą ActionScript

Aby możliwe było dodanie składnika do dokumentu za pomocą języka ActionScript w środowisku wykonawczym, składnik najpierw należy umieścić w bibliotece aplikacji (Okno > Biblioteka) podczas kompilowania pliku SWF. Aby dodać składnik do biblioteki, przeciągnij go z panelu Składniki do panelu Biblioteka. Więcej informacji na temat biblioteki zawiera sekcja „[Biblioteka](#)” na stronie 24.

Plik klasy składnika należy również zaimportować, aby udostępnić jego interfejs API w aplikacji. Pliki klas składników są zainstalowane w *pakietach*, które zawierają jedną lub większą liczbę klas. W celu zaimportowania klasy składnika użyj instrukcji `import` i określ nazwę pakietu oraz nazwę klasy. Do zaimportowania klasy `Button` posłuży na przykład poniższa instrukcja `import`:

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

Więcej informacji na temat tego, w którym pakiecie znajduje się składnik, zawiera [Skorowidz języka ActionScript 3.0](#). Więcej informacji o lokalizacji plików źródłowych składników zawiera sekcja „[Praca z plikami składników](#)” na stronie 20.

W celu utworzenia instancji składnika należy wywołać metodę konstruktora ActionScript składnika. Przykład: poniższa instrukcja tworzy instancję `Button` o nazwie `aButton`:

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

Końcowym krokiem jest wywołanie statycznej metody `addChild()` w celu dodania instancji składnika do stołu montażowego lub kontenera aplikacji. Przykład: poniższa instrukcja dodaje instancję `aButton`:

```
addChild(aButton);
```

W tym momencie można użyć interfejsu API komponentu w celu dynamicznego określenia wielkości składnika i jego pozycji na stole montażowym w celu nasłuchiwanie zdarzeń oraz w celu ustawienia właściwości i zmodyfikowania jego działania. Więcej informacji na temat interfejsu API dla określonego składnika zawiera [Skorowidz języka ActionScript 3.0](#).

Więcej informacji na temat metody `addChild()` zawiera sekcja „Praca z listą wyświetlania” na stronie 27.

## Usuwanie składników

W celu usunięcia instancji składnika ze stołu montażowego podczas tworzenia należy po prostu wybrać instancję i nacisnąć klawisz Delete. To spowoduje usunięcie instancji ze stołu montażowego, ale nie spowoduje usunięcia składnika z aplikacji.

Aby usunąć składnik z dokumentu Flash po umieszczeniu go na stole montażowym lub w bibliotece, należy usunąć z biblioteki składnik i skojarzone z nim zasoby. Nie wystarczy tylko usunąć składnika ze stołu montażowego. Jeśli składnik nie zostanie usunięty z biblioteki, zostanie dołączony do aplikacji podczas jej kompilowania.

- 1 W panelu Biblioteka zaznacz symbol dla składnika.
- 2 Kliknij przycisk Usuń u dołu panelu Biblioteka lub przycisk Usuń w menu panelu Biblioteka.

Powtórz te czynności, aby usunąć wszystkie zasoby skojarzone z danym składnikiem.

Informacje na temat sposobów usuwania składników z ich kontenera podczas działania aplikacji zawiera sekcja „Usuwanie składnika z listy wyświetlania” na stronie 28.

## Wyszukiwanie wersji składnika

Składniki języka Flash ActionScript 3.0 udostępniają właściwość wersji, którą w razie potrzeby można wyświetlić w celu udostępnienia w sekcji pomocy Adobe Technical Support lub gdy wymagane jest określenie wersji używanego składnika.

### Wyświetlanie numeru wersji składnika interfejsu użytkownika:

- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij składnik na stół montażowy i nadaj nazwę instancji. Na przykład: przeciągnij ComboBox na stół montażowy i nadaj mu nazwę **aCb**.
- 3 Naciśnij klawisz **F9** lub wybierz polecenia Okno > Operacje, aby otworzyć panel Operacje.
- 4 Kliknij 1. klatkę na głównej osi czasu, a następnie dodaj następujący kod do panelu Operacje.

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

Numer wersji, podobny do przedstawionego na poniższej ilustracji, powinien pojawić się w panelu Wyjście.

W przypadku składników `FLVPlayback` i `FLVPlaybackCaptioning` należy odwoływać się do nazw klas, a nie do nazw instancji, ponieważ numer wersji jest przechowywany w stałej klasy.

### Wyświetlanie numeru wersji dla składników `FLVPlayback` i `FLVPlaybackCaptioning`:

- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij składniki `FLVPlayback` i `FLVPlaybackCaptioning` do panelu Biblioteka.
- 3 Naciśnij klawisz **F9** lub wybierz polecenia Okno > Operacje, aby otworzyć panel Operacje.

- 4 Kliknij 1. klatkę na głównej osi czasu, a następnie dodaj następujący kod do panelu Operacje.

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

Numery wersji pojawiają się w panelu Wyjście.

## Model obsługi zdarzeń języka ActionScript 3.0

Język ActionScript 3.0 wprowadza pojedynczy model obsługi zdarzeń, który zastępuje różne modele obsługi, jakie istniały w poprzednich wersjach języka ActionScript. Nowy model zdarzeń jest oparty na specyfikacji Document Object Model (DOM) Level 3 Events Specification.

Dla programistów, którzy mają doświadczenie w używaniu metody `addEventListener()` języka ActionScript 2.0, pomocne może być określenie różnic między modelem obsługi zdarzeń w języku ActionScript 2.0 a modelem w języku ActionScript 3.0. W poniższej liście przedstawiono kilka z najważniejszych różnic między tymi dwoma modelami:

- W celu dodania programów nasłuchiwanie zdarzeń w języku ActionScript 2.0 w niektórych przypadkach należy użyć metody `addEventListener()`, a w innych metody `addEventListener()` — natomiast w języku ActionScript 3.0 we wszystkich przypadkach należy korzystać z metody `addEventListener()`.
- W języku ActionScript 2.0 nie ma przepływu zdarzeń, co oznacza, że metoda `addEventListener()` może być wywoływana tylko dla obiektu, który rozgłasza zdarzenie, natomiast w języku ActionScript 3.0 metoda `addEventListener()` może być wywoływana dla dowolnych obiektów, które są częścią przepływu zdarzeń.
- W języku ActionScript 2.0 detektory zdarzeń mogą być funkcjami, metodami lub obiektami, a w języku ActionScript 3.0 tylko funkcje i metody mogą nasłuchiwać zdarzeń.
- Składnia `on(event)` nie jest obsługiwana w języku ActionScript 3.0, dlatego nie można dołączyć kodu zdarzenia ActionScript do klipu filmowego. W celu dodania programu nasłuchiwanie zdarzeń można użyć tylko metody `addEventListener()`.

Poniższy przykład, który nasłuchuje zdarzenia `MouseEvent.CLICK` dla komponentu `Button` o nazwie `aButton`, przedstawia prosty model obsługi zdarzeń języka ActionScript 3.0:

```
aButton.addEventListener(MouseEvent.CLICK, clickHandler);
function clickHandler(event:MouseEvent):void {
    trace("clickHandler detected an event of type: " + event.type);
    trace("the event occurred on: " + event.target.name);
}
```

Więcej informacji na temat obsługi zdarzeń w języku ActionScript 3.0 zawiera sekcja *Programowanie w języku ActionScript 3.0*. Więcej informacji na temat obsługi zdarzeń w języku ActionScript 3.0 dla składników zawiera sekcja „Obsługa zdarzeń” na stronie 25.

## Prosta aplikacja

W niniejszej sekcji przedstawiono czynności, jakie należy wykonać w celu utworzenia prostej aplikacji ActionScript 3.0 z wykorzystaniem składników Flash i narzędzia do tworzenia treści Flash. Przykład został przedstawiony jako plik FLA w kodzie ActionScript wprowadzonym na osi czasu, a także jako zewnętrzny plik klasy ActionScript z plikiem FLA, który zawiera tylko składniki w bibliotece. Zwykle tworzone są większe aplikacje z wykorzystaniem zewnętrznych plików klas, dzięki czemu możliwe jest współużytkowanie kodu przez klasy i aplikacje, co umożliwia łatwiejsze zarządzanie aplikacjami. Więcej informacji na temat programowania w języku ActionScript 3.0 zawiera sekcja *Programowanie w języku ActionScript 3.0*.

### Projekt aplikacji

Pierwszy przykład aplikacji ze składnikiem ActionScript to odmiana standardowej aplikacji „Hello World” — jej projekt jest dość prosty:

- Aplikacja ma nazwę Greetings.
- Korzysta ze składnika TextArea w celu wyświetlenia powitania, które na początku brzmi Hello World.
- Korzysta ze składnika ColorPicker, który umożliwia zmianę koloru tekstu.
- Korzysta z trzech składników RadioButton, które umożliwiają ustawienie małego, większego i największego rozmiaru tekstu.
- Korzysta ze składnika ComboBox, który umożliwia wybór innego powitania z listy rozwijanej.
- Aplikacja korzysta ze składników z panelu Składniki, a ponadto tworzy elementy aplikacji za pomocą kodu ActionScript.

Po określeniu tej definicji można rozpocząć tworzenie aplikacji.

### Tworzenie aplikacji Greetings

Poniższe kroki dotyczą tworzenia aplikacji Greetings za pomocą narzędzia do tworzenia treści Flash używanego w celu utworzenia pliku FLA, umieszczenia komponentów na stole montażowym oraz w celu dodania kodu ActionScript do osi czasu.

#### Tworzenie aplikacji Greetings w pliku FLA:

- 1 Wybierz polecenie Plik > Nowy.
- 2 W oknie dialogowym Nowy dokument wybierz typ Plik Flash (ActionScript 3.0), a następnie kliknij przycisk OK. Zostanie otwarte nowe okno.
- 3 Wybierz opcje Plik > Zapisz, nadaj plikowi Flash nazwę **Greetings fla**, a następnie kliknij przycisk Zapisz.
- 4 Na panelu Składniki Flash wybierz składnik TextArea i przeciągnij go na stół montażowy.
- 5 W oknie Właściwości, ze składnikiem TextArea wybranym na stole montażowym, wpisz **aTa** jako nazwę instancji i wprowadź następujące informacje:
  - Wprowadź **230** dla wartości Sz (szerokość).
  - Wprowadź **44** dla wartości W (wysokość).
  - Wprowadź **165** dla wartości X (położenie w poziomie).
  - Wprowadź **57** dla wartości Y (położenie w pionie).
  - Wprowadź tekst **Hello World!** dla parametru tekstu na karcie Parametry.

- 6 Przeciągnij składnik ColorPicker na stół montażowy, umieść go po lewej stronie obszaru TextArea, a następnie nadaj mu nazwę **txtCp**. Wprowadź następujące informacje do Inspektora właściwości:
  - Wprowadź **96** dla wartości X.
  - Wprowadź **72** dla wartości Y.
- 7 Przeciągnij pojedynczo trzy składniki RadioButton do stołu montażowego i nadaj im nazwy: **smallRb**, **largerRb** oraz **largestRb**. Do Inspektora właściwości wprowadź następujące informacje dotyczące tych składników:
  - Wprowadź **100** dla wartości Sz i **22** dla wartości W każdego z tych składników.
  - Wprowadź **155** dla wartości X.
  - Wprowadź **120** dla wartości Y dla **smallRb**, **148** dla **largerRb** oraz **175** dla **largestRb**.
  - Wprowadź **fontRbGrp** dla parametru groupName każdego z tych składników.
  - Na karcie Parametry wprowadź dla składników etykiety: **Small**, **Larger** i **Largest**.
- 8 Przeciągnij składnik ComboBox na stół montażowy i nadaj mu nazwę **msgCb**. Do Inspektora właściwości wprowadź następujące informacje dotyczące tego składnika:
  - Wprowadź **130** dla wartości Sz (szerokość).
  - Wprowadź **265** dla wartości X.
  - Wprowadź **120** dla wartości Y.
  - Na karcie Parametry wprowadź **Greetings** dla parametru wywołania.
  - Kliknij dwukrotnie pole tekstowe dla parametru dataProvider, aby otworzyć okno dialogowe Wartości.
  - Kliknij znak plus i zastąp wartość etykiety ciągiem znaków **Hello World!**
  - Powtórz poprzedni krok, aby dodać wartości etykiet **Have a nice day!** i **Top of the Morning!**
  - Kliknij przycisk OK, aby zamknąć okno dialogowe Wartości.
- 9 Zapisz plik.
- 10 Jeśli panel Operacje nie jest jeszcze otwarty, otwórz go, naciskając klawisz **F9** lub wybierając przycisk Operacje w menu Okno. Kliknij 1. klatkę na głównej osi czasu, a następnie wprowadź następujący kod do panelu Operacje:

```
import flash.events.Event;
import fl.events.ComponentEvent;
import fl.events.ColorPickerEvent;
import fl.controls.RadioButtonGroup;

var rbGrp:RadioButtonGroup = RadioButtonGroup.getGroup("fontRbGrp");
rbGrp.addEventListener(MouseEvent.CLICK, rbHandler);
txtCp.addEventListener(ColorPickerEvent.CHANGE, cpHandler);
msgCb.addEventListener(Event.CHANGE, cbHandler);
```

Pierwsze trzy wiersze powodują import klas zdarzeń, z których korzysta aplikacja. Gdy użytkownik wchodzi w interakcje z jednym z tych trzech składników, dochodzi do zdarzenia. Następne pięć wierszy rejestruje programy obsługi zdarzeń dla zdarzeń, których aplikacja będzie nasłuchiwać. Zdarzenie `click` występuje dla składnika `RadioButton`, gdy użytkownik go kliknie. Zdarzenie `change` występuje, gdy użytkownik wybierze inny kolor w składniku `ColorPicker`. Zdarzenie `change` występuje dla pola `ComboBox`, gdy użytkownik wybierze inne powitanie z listy rozwijanej.

Czwarty wiersz importuje klasę `RadioButtonGroup`, dzięki czemu aplikacja może przypisać program nasłuchiwanie zdarzeń do grupy składników `RadioButton` - nie ma konieczności przypisywania programu nasłuchiwanie do każdego przycisku osobno.



- 11 Dodaj poniższy wiersz kodu do panelu Operacje, aby utworzyć obiekt `tf` `TextFormat`, z którego korzysta aplikacja w celu zmiany właściwości `size` i `color` stylu tekstu w obszarze `TextArea`.

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

- 12 Dodaj poniższy kod w celu utworzenia funkcji obsługi zdarzeń `rbHandler`. Ta funkcja obsługuje zdarzenie `click`, gdy użytkownik kliknie jeden ze składników `RadioButton`.

```
function rbHandler(event:MouseEvent):void {  
    switch(event.target.selection.name) {  
        case "smallRb":  
            tf.size = 14;  
            break;  
        case "largerRb":  
            tf.size = 18;  
            break;  
        case "largestRb":  
            tf.size = 24;  
            break;  
    }  
    aTa.setStyle("textFormat", tf);  
}
```

Ta funkcja korzysta z instrukcji `switch` w celu sprawdzenia właściwości `target` obiektu `event` w celu ustalenia składnika `RadioButton`, który jest aktywowany przez zdarzenie. Właściwość `currentTarget` zawiera nazwę obiektu, który wywołał zdarzenie. W zależności od tego, który składnik `RadioButton` kliknął użytkownik, aplikacja zmienia wielkość czcionki w obszarze `TextArea` na 14, 18 lub 24 punkty.

- 13 Dodaj poniższy kod w celu zaimplementowania funkcji `cpHandler()`, która będzie obsługiwała zmianę wartości w składniku `ColorPicker`:

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

Funkcja ustawia właściwość `color` obiektu `tf` `TextFormat` na kolor wybrany w składniku `ColorPicker`, a następnie wywołuje metodę `setStyle()` w celu zastosowania tego koloru dla tekstu w instancji `aTa` `TextArea`.

- 14 Dodaj poniższy kod w celu zaimplementowania funkcji `cbHandler()`, która będzie obsługiwała zmianę opcji wybranej w składniku `ComboBox`:

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

Ta funkcja po prostu zastępuje tekst w obszarze `TextArea` tekstem wybranym w składniku `ComboBox`, `event.target.selectedItem.label`.

- 15 Wybierz opcje Sterowanie > Testuj film albo naciśnij klawisze `Control+Enter`, aby skompilować kod i przetestować aplikację `Greetings`.

W poniższej sekcji przedstawiono sposób tworzenia tej samej aplikacji za pomocą zewnętrznej klasy `ActionScript` oraz pliku `FLA`, który zawiera tylko wymagane składniki w bibliotece.

#### Tworzenie aplikacji `Greetings2` za pomocą zewnętrznego pliku klasy:

- 1 Wybierz polecenie `Plik > Nowy`.
- 2 W oknie dialogowym `Nowy dokument` wybierz typ `Plik Flash (ActionScript 3.0)`, a następnie kliknij przycisk `OK`.

Zostanie otwarte nowe okno.

3 Wybierz opcje Plik > Zapisz, nadaj plikowi Flash nazwę **Greetings2.fla**, a następnie kliknij przycisk Zapisz.

4 Przeciągnij każdy z poniższych składników z panelu Składniki do biblioteki:

- ColorPicker
- ComboBox
- RadioButton:
- TextArea

W skompilowanym pliku SWF wykorzystywany będzie każdy z tych zasobów, dlatego wymagane jest dodanie ich do biblioteki. Przeciągnij składniki na dół panelu Biblioteka. W miarę dodawania tych składników do biblioteki inne zasoby (takie jak List, TextInput i UIScrollBox) są dodawane automatycznie.

5 W oknie Właściwości, dla klasy Document Class, wpisz **Greetings2**.

Jeśli program Flash wyświetli ostrzeżenie z informacją o tym, że definicja klasy dokumentu nie została znaleziona, zignoruj to ostrzeżenie. Klasa Greetings2 zostanie zdefiniowana w kolejnych krokach. Ta klasa definiuje główną funkcję dla aplikacji.

6 Zapisz plik Greetings2.fla.

7 Wybierz polecenie Plik > Nowy.

8 W oknie dialogowym Nowy dokument wybierz typ Plik ActionScript, a następnie kliknij przycisk OK.

Zostanie otwarte nowe okno skryptu.

9 Wprowadź poniższy kod do okna skryptu:

```
package {
    import flash.display.Sprite;
    import flash.events.Event;
    import flash.events.MouseEvent;
    import flash.text.TextFormat;
    import fl.events.ComponentEvent;
    import fl.events.ColorPickerEvent;
    import fl.controls.ColorPicker;
    import fl.controls.ComboBox;
    import fl.controls.RadioButtonGroup;
    import fl.controls.RadioButton;
    import fl.controls.TextArea;
    public class Greetings2 extends Sprite {
        private var aTa:TextArea;
        private var msgCb:ComboBox;
        private var smallRb:RadioButton;
        private var largerRb:RadioButton;
        private var largestRb:RadioButton;
        private var rbGrp:RadioButtonGroup;
        private var txtCp:ColorPicker;
        private var tf:TextFormat = new TextFormat();
        public function Greetings2() {
```

Skrypt definiuje klasę ActionScript 3.0 o nazwie Greetings2. Skrypt wykonuje następujące czynności:

- Importuje klasy, które będą używane w pliku. Zwykle te instrukcje importu są dodawane w przypadku odwoływania się do innych klas w kodzie, ale w celu skrócenia zapisu w tym przykładzie przedstawiono import wszystkich w jednym kroku.

- Deklaruje zmienne, które reprezentują różne typy obiektów składników, które będą dodawane do kodu. Inna zmienna tworzy obiekt `tf` `TextFormat`.
- Definiuje funkcję konstruktora `Greetings2()` dla klasy. W poniższych krokach do tej funkcji będą dodawane wiersze, a do klasy będą dodawane inne metody.

10 Wybierz opcję Plik > Zapisz, nadaj plikowi nazwę **Greetings2.as**, a następnie kliknij przycisk Zapisz.

11 Dodaj poniższe wiersze kodu, do funkcji `Greeting2()`:

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

Funkcja powinna mieć teraz następującą postać:

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

12 Dodaj poniższe wiersze kodu za nawiasem zamykającym metody `Greeting2()`:

```
private function createUI() {
    bldTxtArea();
    bldColorPicker();
    bldComboBox();
    bldRadioButtons();
}
private function bldTxtArea() {
    aTa = new TextArea();
    aTa.setSize(230, 44);
    aTa.text = "Hello World!";
    aTa.move(165, 57);
    addChild(aTa);
}
private function bldColorPicker() {
    txtCp = new ColorPicker();
    txtCp.move(96, 72);
    addChild(txtCp);
}
private function bldComboBox() {
    msgCb = new ComboBox();
    msgCb.width = 130;
    msgCb.move(265, 120);
    msgCb.prompt = "Greetings";
    msgCb.addItem({data:"Hello.", label:"English"});
    msgCb.addItem({data:"Bonjour.", label:"Français"});
    msgCb.addItem({data:"¡Hola!", label:"Español"});
    addChild(msgCb);
}
private function bldRadioButtons() {
    rbGrp = new RadioButtonGroup("fontRbGrp");
    smallRb = new RadioButton();
    smallRb.setSize(100, 22);
```

```
        smallRb.move(155, 120);
        smallRb.group = rbGrp; // "fontRbGrp";
        smallRb.label = "Small";
        smallRb.name = "smallRb";
        addChild(smallRb);
        largerRb = new RadioButton();
        largerRb.setSize(100, 22);
        largerRb.move(155, 148);
        largerRb.group = rbGrp;
        largerRb.label = "Larger";
        largerRb.name = "largerRb";
        addChild(largerRb);
        largestRb = new RadioButton();
        largestRb.setSize(100, 22);
        largestRb.move(155, 175);
        largestRb.group = rbGrp;
        largestRb.label = "Largest";
        largestRb.name = "largestRb";
        addChild(largestRb);
    }
}
```

Te wiersze wykonują następujące czynności:

- Inicjują składniki używane w aplikacji.
- Ustawiają wielkość, pozycję i właściwości każdego składnika.
- Dodają każdy składnik do stołu montażowego, korzystając z metody `addChild()`.

**13** Za nawiasem zamykającym metody `bldRadioButtons()` dodaj poniższy kod dla metody `setUpHandlers()`:

```
private function setUpHandlers():void {
    rbGrp.addEventListener(MouseEvent.CLICK, rbHandler);
    txtCp.addEventListener(ColorPickerEvent.CHANGE, cpHandler);
    msgCb.addEventListener(Event.CHANGE, cbHandler);
}

private function rbHandler(event:MouseEvent):void {
    switch(event.target.selection.name) {
        case "smallRb":
            tf.size = 14;
            break;
        case "largerRb":
            tf.size = 18;
            break;
        case "largestRb":
            tf.size = 24;
            break;
    }
    aTa.setStyle("textFormat", tf);
}

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

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

Te funkcje definiują programy nasłuchiwanie zdarzeń dla składników.

14 Wybierz opcje Plik > Zapisz, aby zapisać plik.

15 Wybierz opcje Sterowanie > Testuj film albo naciśnij klawisze Control+Enter, aby skompilować kod i przetestować aplikację Greetings2.

## **Programowanie i uruchamianie kolejnych przykładów**

Po zaprogramowaniu i uruchomieniu aplikacji Greetings użytkownik posiada już podstawową wiedzę, jaka jest wymagana do uruchamiania innych przykładów kodu, jakie zostały przedstawione w niniejszej publikacji. Istotny kod ActionScript 3.0 w każdym przykładzie będzie wyróżniony i omówiony, a użytkownik powinien kopiować poszczególne przykłady z tej publikacji do pliku FLA, a następnie kompilować i uruchamiać plik.

# Rozdział 3: Praca ze składnikami

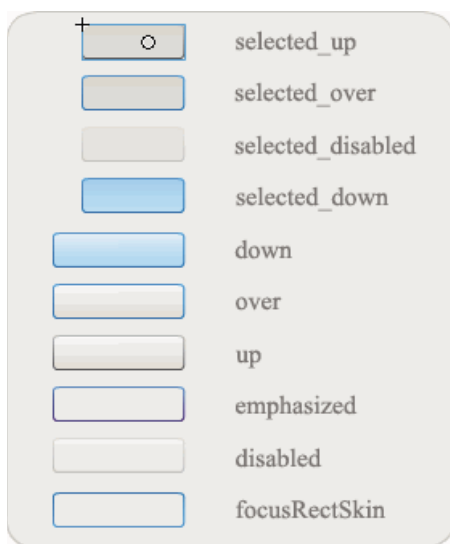
## Architektura składników

Składniki języka Adobe® ActionScript® 3.0 są obsługiwane przez program Adobe® Flash Player w wersji 9.0.28.0 i późniejszych. Składniki te nie są kompatybilne ze składnikami zbudowanymi w wersjach programu Flash wcześniejszych niż CS4. Informacje na temat korzystania ze składników języka Adobe® ActionScript® 2.0 zawiera podręcznik *Korzystanie ze składników języka Adobe® ActionScript® 2.0* oraz *Skorowidz języka i składników Adobe® ActionScript® 2.0*.

Składniki interfejsu użytkownika w języku Adobe ActionScript 3.0 są zaimplementowane jako składniki w formacie FLA, jednak program Flash CS5 obsługuje zarówno składniki w formacie SWC, jak i FLA. Na przykład FLVPlayback i FLVPlaybackCaptioning to składniki w formacie SWC. W folderze Składniki można umieszczać składniki dowolnego z tych dwóch typów, dzięki czemu będą one widoczne w panelu Składniki. Składniki należące do każdego z tych typów różnią się pod względem budowy, dlatego zostaną tutaj opisane osobno.

### Składniki języka ActionScript 3.0 w formacie FLA

Składniki interfejsu użytkownika w języku ActionScript 3.0 są plikami w formacie FLA zawierającymi wbudowane karnacje. Dostęp do składników można uzyskiwać, klikając je dwukrotnie na stole montażowym. Karnacje składnika i pozostałe zasoby są umieszczone na klatce nr 2 osi czasu. Gdy użytkownik kliknie dwukrotnie składnik, program Flash automatycznie przechodzi do klatki nr 2 i otwiera paletę karnacji składnika. Na poniższej ilustracji przedstawiono paletę karnacji wyświetlaną w przypadku składnika Button.



Karnacje składnika Button

Więcej informacji na temat karnacji składników i dostosowywaniu składników zawierają sekcje „[Dostosowywanie składników interfejsu użytkownika](#)” na stronie 102 oraz „[Dostosowywanie składnika FLVPlayback](#)” na stronie 158.

Aby przyspieszyć kompilowanie aplikacji i uniknąć konfliktów z ustawieniami języka ActionScript 3.0, składniki interfejsu użytkownika programu Flash CS5 w formacie FLA zawierają także pakiet SWC ze skompilowanym kodem ActionScript składnika. Składnik SWC ComponentShim jest umieszczony na stole montażowym na klatce nr 2 każdego składnika interfejsu użytkownika i udostępnia wstępnie skompilowane definicje. Aby składnik był dostępny dla kodu w języku ActionScript, musi znajdować się na stole montażowym lub musi być umieszczony w bibliotece i we właściwościach Powiązanie musi mieć wybraną opcję eksportowania w pierwszej klatce. Aby utworzyć składnik w języku ActionScript, należy ponadto zaimportować klasę za pomocą instrukcji `import`; umożliwi to dostęp do składnika. Więcej informacji na temat instrukcji `import` zawiera [Skorowidz języka ActionScript 3.0 dla programu Flash Professional](#).

## Składniki w formacie SWC

Składniki w formacie SWC także pierwotnie składały się z pliku FLA i pliku klasy ActionScript, ale zostały skompilowane i wyeksportowane w formacie SWC. Plik SWC to pakiet wstępnie skompilowanych symboli Flash i kodu ActionScript umożliwiający uniknięcie ponownej kompilacji symboli i fragmentów kodu, które nie ulegają zmianie.

FLVPlayback i FLVPlaybackCaptioning to składniki w formacie SWC. Mają one karnacje zewnętrzne, a nie wbudowane. Domyślną karnację składnika FLVPlayback można zmienić, wybierając inną ze zbioru wstępnie zdefiniowanych karnacji, dostosowując składniki interfejsu użytkownika w panelu Składniki (składniki BackButton, BufferingBar itd.) lub tworząc karnację niestandardową. Więcej informacji zawiera sekcja „[Dostosowywanie składnika FLVPlayback](#)” na stronie 158.

W programie Flash istnieją następujące metody konwertowania klipu filmowego na skompilowany klip:

### Kompilowanie klipu filmowego

- Kliknij klip filmowy w panelu Biblioteka prawym przyciskiem myszy (Windows) lub kliknij go, trzymając naciśnięty klawisz Control (Macintosh), a następnie wybierz polecenie Konwertuj na skompilowany klip.

Skompilowany klip zachowuje się tak samo, jak klip filmowy, na podstawie którego powstał, ale jego wyświetlanie i publikacja odbywa się znacznie szybciej niż w przypadku zwykłego klipu filmowego. Skompilowanych klipów nie można edytować, ale ich właściwości mogą być wyświetlane w Inspektorze właściwości i w Inspektorze składników.

Składniki SWC zawierają skompilowany klip, wstępnie skompilowane definicje składnika w języku ActionScript oraz inne pliki opisujące składnik. Użytkownik może eksportować utworzone samodzielnie składniki w postaci plików SWC przeznaczonych do rozpowszechniania.

### Eksport pliku SWC

- Zaznacz klip filmowy w panelu Biblioteka i kliknij go prawym przyciskiem myszy (Windows) lub kliknij go, trzymając naciśnięty klawisz Control (Macintosh), a następnie wybierz polecenie Eksportuj plik SWC.

**Uwaga:** *Format plików SWC generowanych przez program Flash CS4 i nowsze wersje jest zgodny z formatem SWC programu Flex, a zatem możliwa jest wymiana plików SWC między tymi dwoma produktami, choć nie zawsze obywa się to bez niezbędnych modyfikacji.*

Informacje na temat tworzenia składników w formacie SWC można znaleźć pod adresem [www.adobe.com/go/learn\\_fl\\_creating\\_components\\_pl](http://www.adobe.com/go/learn_fl_creating_components_pl).

## Interfejs API składników języka ActionScript 3.0

W języku ActionScript 3.0 każdy składnik jest zbudowany na bazie klasy ActionScript 3.0 umieszczonej w folderze pakietu o nazwie w postaci `fl.nazwa_pakietu.nazwa_klasy`. Na przykład składnik Button jest instancją klasy Button, a nazwa jego pakietu to `fl.controls.Button`. Z nazwy pakietu należy skorzystać, importując klasę składnika do aplikacji. Do zaimportowania klasy Button może posłużyć następująca instrukcja:

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

Więcej informacji i lokalizacji plików klas składników zawiera sekcja „Praca z plikami składników” na stronie 20.

Klasa składnika definiuje metody, właściwości, zdarzenia i style umożliwiające interakcje ze składnikiem w aplikacji użytkownika. Składniki interfejsu użytkownika w języku ActionScript 3.0 są podklasami klas Sprite oraz UIComponent i dziedziczą z tych klas właściwości, metody oraz zdarzenia. Klasa Sprite to podstawowy element składowy listy wyświetlania, podobny do składnika MovieClip, ale w odróżnieniu od niego pozbawiony osi czasu. Klasa UIComponent stanowi bazową klasę wszystkich składników wizualnych — zarówno interaktywnych, jak i nieinteraktywnych. Ścieżkę dziedziczenia każdego składnika, a także jego właściwości, metody, zdarzenia i style opisano w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional* firmy Adobe.

Wszystkie składniki języka ActionScript 3.0 działają zgodnie z modelem obsługi zdarzeń w języku ActionScript 3.0. Więcej informacji na temat obsługi zdarzeń zawiera sekcja „Obsługa zdarzeń” na stronie 25 oraz podręcznik *Programowanie w języku ActionScript 3.0*.

## Praca z plikami składników

W tej sekcji omówiono miejsca przechowywania plików składników oraz plików źródłowych w języku ActionScript, a także sposób dodawania i usuwania składników w panelu Składniki.

### Miejsce przechowywania plików składników

Składniki programu Flash są przechowywane w folderze Configuration na poziomie aplikacji.

**Uwaga:** Więcej informacji na temat tych składników zawiera sekcja „Foldery konfiguracji instalowane z programem Flash” w podręczniku *Korzystanie z programu Flash*.

Składniki są instalowane w następujących lokalizacjach:

- Windows 2000 lub Windows XP: C:\Program Files\Adobe\Adobe Flash CS5\język\Configuration\Components
- Mac OS X: Macintosh HD:Programy:Adobe Flash CS5:Configuration:Components

Wewnątrz folderu Components składniki interfejsu użytkownika są zapisane w pliku User Interface fla, natomiast składniki FLVPlayback (FLVPlaybackAS3.swc) i FLVPlaybackCaptioning znajdują się w folderze Video.

Składniki można także zapisywać w następujących lokalizacjach przypisanych do konkretnych użytkowników:

- Windows 2000 lub Windows XP: C:\Documents and Settings\nazwa\_użytkownika\Ustawienia lokalne\Dane aplikacji\Adobe\Adobe Flash CS5\en\Configuration\Components
- Windows Vista: C:\Użytkownicy\nazwa\_użytkownika\Ustawienia lokalne\Dane aplikacji\Adobe\Adobe Flash CS5\pl\Configuration\Components

**Uwaga:** W systemie Windows folder Dane aplikacji jest domyślnie ukryty. Aby pokazać ukryte pliki i foldery, wybierz ikonę Mój komputer, aby otworzyć Eksploratora Windows, wybierz polecenia Narzędzia > Opcje folderów, a następnie wybierz kartę Widok. Na karcie Widok zaznacz pole wyboru Pokaż ukryte pliki i foldery.



- Mac OS X: Macintosh HD:Użytkownicy:<nazwa\_użytkownika>:Biblioteki:Application Support:Adobe Flash CS5:Configuration:Components

## Miejsce przechowywania plików źródłowych składników

Pliki klas języka ActionScript (pliki AS, czyli *pliki źródłowe*) składników są w systemach Windows 2000 i Windows XP instalowane w następujących folderach aplikacji:

**Składniki interfejsu użytkownika** C:\Program Files\Adobe\Adobe Flash CS5\pl\Configuration\Component Source\ActionScript 3.0\User Interface\fl

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

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

W systemie Mac OS X pliki źródłowe składników są zapisane w następujących folderach:

**Składniki interfejsu użytkownika** Macintosh HD:Programy:Adobe Flash CS5:Configuration:Component Source>ActionScript 3.0>User Interface:fl

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

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

## Pliki źródłowe składników i ścieżka klas

Ponieważ składniki w języku ActionScript 3.0 zawierają skompilowany kod, nie należy określać położenia plików ich klasy ActionScript w zmiennej Classpath. Uwzględnienie lokalizacji plików klas składników w ścieżce klas wydłuży czas kompilowania aplikacji. Jeśli jednak program Flash napotka odwołania do plików klas w zmiennej Classpath, pliki klas będą zawsze miały wyższy priorytet niż skompilowany kod zawarty w składniku.

Jedną z sytuacji uzasadniających dodanie lokalizacji plików źródłowych składników do zmiennej Classpath jest debugowanie aplikacji ze składnikami. Więcej informacji zawiera sekcja „[Debugowanie aplikacji ze składnikami](#)” na stronie 22.

## Modyfikowanie plików składników

Po każdej modyfikacji, dodaniu lub usunięciu składników w formacie SWC lub po dodaniu nowych składników w formacie FLA do środowiska programu Flash należy ponownie załadować te składniki do panelu Składniki, aby stały się dostępne. Ponowne załadowanie składników następuje po ponownym uruchomieniu programu Flash lub po wybraniu polecenia Ładuj ponownie z menu panelu Składniki. Ponowne załadowanie spowoduje pobranie przez program Flash wszelkich składników dodanych do folderu Składniki.


### Ponowne ładowanie składników do panelu Składniki, gdy program Flash jest uruchomiony:

- Wybierz polecenie Ładuj ponownie z menu panelu Składniki.

**Usuwanie składnika z panelu Składniki:**

- Usuń plik FLA, SWC lub MXP z folderu Components i ponownie uruchom program Flash albo wybierz polecenie Ładuj ponownie z menu panelu Składniki. Plik MXP jest to plik składnika pobrany z witryny Adobe Exchange.

Możliwe jest usuwanie i zastępowanie składników w formacie SWC w trakcie pracy programu Flash, a ponowne załadowanie spowoduje, że zmiany zostaną odzwierciedlone. Jeśli jednak modyfikacja lub usunięcie dotyczyło składników w formacie FLA, zmiany zostaną odzwierciedlone dopiero po zakończeniu pracy i ponownym uruchomieniu programu Flash. Możliwe jest natomiast dodawanie składników FLA i ładowanie ich za pomocą polecenia Ładuj ponownie.

 Firma Adobe zaleca, aby przed zmodyfikowaniem jakiegokolwiek pliku składnika programu Flash (FLA lub AS) utworzyć kopię zapasową tego pliku. Umożliwi to odtworzenie oryginalnego składnika, gdyby zaszła taka konieczność.

## Debugowanie aplikacji ze składnikami

Składniki ActionScript 3.0 zawierają cały swój kod źródłowy, co skraca czas kompilacji aplikacji, w których są wykorzystywane. Jednak debugger programu Flash nie umożliwia inspekcji kodu w skompilowanych klipach. Dlatego, jeśli planowane jest debugowanie aplikacji na poziomie kodu źródłowego składników, należy dodać pliki źródłowe składników do zmiennej Classpath.

Lokalizacja folderów z pakietami składników jest określona względem lokalizacji plików źródłowych składników danego typu. Aby uwzględnić w ścieżce klas odwołania do wszystkich plików źródłowych języka ActionScript 3.0 dla wszystkich składników interfejsu użytkownika, należy dodać do zmiennej Classpath następującą lokalizację pakietów interfejsu użytkownika:

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

**Uwaga:** Spowoduje to przesłonięcie skompilowanego kodu wszystkich składników interfejsu użytkownika i wydłuży czas kompilowania aplikacji. Jeśli z jakiegokolwiek powodu zmodyfikowano plik źródłowy składnika, składnik ten może działać inaczej niż dotychczas.

Aby ustawić zmienną Classpath, należy wybrać polecenie Preferencje z menu Edycja, następnie z listy kategorii wybrać pozycję ActionScript, po czym kliknąć przycisk Ustawienia ActionScript 3.0. Aby dodać nowy wpis, należy kliknąć ikonę plusa nad oknem, w którym wyświetlane są bieżące ustawienia.

Zmienna \$(AppConfig) oznacza folder konfiguracji programu Flash CS5 wewnątrz lokalizacji, w której jest zainstalowany program Flash CS5. Zwykle ta ścieżka ma następującą postać:

- Windows 2000 lub Windows XP: C:\Program Files\Adobe\Adobe Flash CS5\język\Configuration\
- Mac OS X: Macintosh HD:Programy:Adobe Flash CS5:Configuration

**Uwaga:** Jeśli konieczne jest zmodyfikowanie pliku źródłowego składnika, firma Adobe zdecydowanie zaleca skopiowanie oryginalnego pliku źródłowego do innej lokalizacji i dodanie jej do ścieżki klas.

Więcej informacji na temat lokalizacji plików źródłowych składników zawiera sekcja „[Miejsce przechowywania plików źródłowych składników](#)” na stronie 21.

## Ustawianie parametrów i właściwości

Każdy składnik ma parametry, które można modyfikować w celu wpływania na wygląd i zachowanie składnika. Parametry są właściwościami klasy składnika i są wyświetlane w Inspektorze właściwości i Inspektorze składników. Najczęściej używane właściwości są widoczne jako parametry tworzenia treści; pozostałe można ustawiać tylko za pomocą języka ActionScript. Wszystkie parametry, jakie można ustawić w narzędziu do tworzenia treści, można także ustawić w języku ActionScript. Wartości parametrów ustawione w języku ActionScript przesłaniają wartości ustawione w narzędziu do tworzenia treści.

Większość składników interfejsu użytkownika w języku ActionScript 3.0 dziedziczy właściwości i metody z klasy `UIComponent` oraz z odpowiedniej klasy bazowej. Na przykład klasy `Button` i `CheckBox` dziedziczą właściwości zarówno z klasy `UIComponent`, jak i z klasy `BaseButton`. Odziedziczone właściwości składnika oraz właściwości jego własnej klasy są dostępne dla użytkownika. Na przykład składnik `ProgressBar` dziedziczy właściwość `ProgressBar.enabled` ze składnika `UIComponent`, ale ma również swoją własną właściwość `ProgressBar.percentComplete`. W ramach interakcji z instancją składnika `ProgressBar` można uzyskiwać dostęp do obu tych właściwości. Więcej informacji na temat właściwości poszczególnych składników zawierają opisy odpowiednich klas w [dokumentacji języka ActionScript 3.0](#).

Parametry instancji składnika można ustawiać za pomocą Inspektora właściwości albo Inspektora składników.

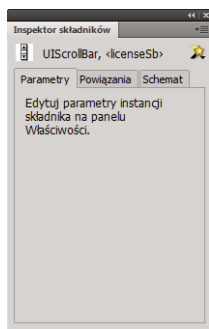
### Wprowadzanie nazwy instancji składnika w Inspektorze właściwości:

- 1 Wybierz polecenie Okno > Właściwości > Właściwości.
- 2 Zaznacz instancję składnika na stole montażowym.
- 3 Wprowadź nazwę instancji składnika w polu <Nazwa instancji>, które znajduje się pod listą rozwijaną z wybraną pozycją Klip filmowy. Możesz też kliknąć kartę Parametry i wprowadzić nazwę w polu poniżej słowa *Składnik*. Wprowadź wartości parametrów, które chcesz ustawić.

Dobłą praktyką jest dodawanie do nazwy instancji przyrostka, który informuje o rodzaju składnika; poprawia to czytelność kodu w języku ActionScript. Na przykład nazwa instancji `licenseSb` wskazuje, że składnik jest paskiem przewijania (`ScrollBar`), który przewija umowę licencyjną w obszarze tekstowym `licenseTa`.

### Wprowadzanie parametrów instancji składnika w Inspektorze składników:

- 1 Wybierz polecenie Okno > Inspektor składników.
- 2 Zaznacz instancję składnika na stole montażowym.
- 3 Kliknij kartę Parametry i wprowadź wartości dowolnych spośród wymienionych parametrów.



Parametry składników w Inspektorze składników

## Ustawianie właściwości składników w języku ActionScript

W języku ActionScript dostęp do właściwości lub metod obiektu lub instancji na stole montażowym uzyskuje się za pomocą operatora kropki (.). Jest to tzw. notacja z kropką. Wyrażenie w notacji z kropką rozpoczyna się od nazwy instancji, po której następuje kropka oraz element będący celem odwołania. Na przykład następujący kod w języku ActionScript ustawia właściwość `width` instancji klasy `CheckBox` `aCh`, nadając jej szerokość 50 pikseli:

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

Poniższa instrukcja `if` sprawdza, czy użytkownik zaznaczył pole wyboru:

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

## Biblioteka

Po dodaniu składnika do dokumentu program Flash importuje go jako klip filmowy do panelu Biblioteka. Można również przeciągnąć składnik z panelu Składniki bezpośrednio do panelu Biblioteka, a następnie dodać jego instancję na stół montażowy. W każdym przypadku należy dodać składnik do biblioteki, aby możliwy był dostęp do elementów jego klasy.

Jeśli składnik został dodany do biblioteki, to przed utworzeniem jego instancji za pomocą języka ActionScript należy zaimportować jego klasę przy użyciu instrukcji `import`. W instrukcji `import` należy określić zarówno nazwę pakietu składnika, jak i nazwę jego klasy. Na przykład następująca instrukcja importuje klasę `Button`:

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

Gdy użytkownik umieści składnik w bibliotece, program Flash importuje także folder z zasobami składnika zawierający karnacje odpowiadające jego różnym stanom. *Karnacje* składnika to kolekcje symboli składających się na postać graficzną składnika w aplikacji. Pojedyncza karnacja to reprezentacja graficzna (lub klip filmowy) składnika w konkretnym stanie.

Zmieniając zawartość folderu zasobów składnika, można w razie potrzeby modyfikować jego karnacje. Więcej informacji zawiera sekcja „[Dostosowywanie składników interfejsu użytkownika](#)” na stronie 102.

Gdy składnik znajduje się w bibliotece, możliwe jest dodawanie do dokumentu kolejnych instancji składnika poprzez przeciągnięcie jego ikony na stół montażowy z panelu Składniki lub z panelu Biblioteka.

## Określanie rozmiaru składników

Do zmiany rozmiaru instancji składników służy narzędzie Przekształcanie swobodne i metoda `setSize()`. Metodę `setSize()` można wywołać z każdej instancji składnika (patrz `UIComponent.setSize()`), aby zmienić rozmiar tej instancji. Poniższy kod zmienia rozmiar instancji składnika `List` na szerokość 200 pikseli i wysokość 300 pikseli:

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

Rozmiar składnika nie jest automatycznie dopasowywany do rozmiaru jego etykiety. W wypadku dodania do dokumentu instancji składnika, która jest za mała, aby możliwe było wyświetlenie całej etykiety, etykieta zostanie obcięta. Należy zmienić rozmiar składnika, aby dopasować go do jego etykiety.

Więcej informacji na temat określania rozmiaru składników zawierają ich poszczególne wpisy w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

## Podgląd aktywny

Funkcja Podgląd aktywny, która jest domyślnie włączona, umożliwia wyświetlanie składników na stole montażowym w takiej postaci, w jakiej będą widoczne w opublikowanej treści Flash; rozmiary wyświetlanych składników są przybliżone.

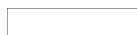
Aby włączyć lub wyłączyć Podgląd aktywny:

- Wybierz polecenie Sterowanie > Włącz podgląd aktywny. Symbol zaznaczenia obok opcji oznacza, że jest ona włączona.

To, jakie parametry będą odzwierciedlone w podglądzie aktywnym, zależy od typu składnika. Więcej informacji na temat parametrów składników uwzględnionych w podglądzie na żywo zawierają opisy poszczególnych składników w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).



Składnik Button z włączonym podglądem aktywnym



Składnik Button z wyłączonym podglądem aktywnym

Składniki w podglądzie aktywnym nie działają. Aby przetestować działanie składników, należy użyć polecenia Sterowania > Testuj film.

## Obsługa zdarzeń

Każdy składnik rozgłasza zdarzenia w odpowiedzi na interakcje z użytkownikiem. Na przykład, gdy użytkownik kliknie przycisk (składnik Button), składnik ten wywołuje zdarzenie `MouseEvent.CLICK`, a gdy użytkownik wybierze element na liście, składnik List wywołuje zdarzenie `Event.CHANGE`. Zdarzenia są także wywoływane po spełnieniu pewnych ważnych warunków wewnętrznych w składniku, na przykład zakończenie ładowania treści w instancji składnika `UILoader` powoduje wygenerowanie zdarzenia `Event.COMPLETE`. Aby obsługiwać zdarzenie, należy napisać w języku ActionScript kod wykonywany po zajściu tego zdarzenia.

Do zdarzeń składnika zaliczają się zdarzenia wszystkich klas, z których ten składnik dziedziczy. Oznacza to, że wszystkie składniki interfejsu użytkownika w języku ActionScript 3.0 dziedziczą zdarzenia z klasy `UIComponent`, ponieważ jest to klasa bazowa dla składników interfejsu użytkownika. Aby wyświetlić listę zdarzeń rozgłaszanych przez składnik, należy zapoznać się z sekcją Zdarzenia opisu klasy składnika w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

Pełny opis mechanizmów obsługi zdarzeń w języku ActionScript 3.0 zawiera podręcznik *Programowanie w języku ActionScript 3.0*.

## Informacje o detektorach zdarzeń

Poniżej przedstawiono najważniejsze zasady obsługi zdarzeń związanych ze składnikami w języku ActionScript 3.0:

- Wszystkie zdarzenia są rozgłaszane przez instancje klas składników. Instancja składnika jest *źródłem zdarzeń*.
- Kod użytkownika rejestruje *detektor* zdarzenia, wywołując metodę `addEventListener()` w instancji składnika. Na przykład poniższy kod dodaje detektor zdarzenia `MouseEvent.CLICK` do instancji składnika Button o nazwie `aButton`:

**Praca ze składnikami**

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

Drugim parametrem metody `addEventListener()` jest nazwa rejestrowanej funkcji, `clickHandler`; ta funkcja będzie wywoływana po zajściu zdarzenia. Funkcja ta jest także nazywana *funkcją wywołania zwrotnego*.

- Dla jednej instancji składnika można zarejestrować wiele detektorów zdarzeń.

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

- Jeden detektor zdarzeń można zarejestrować dla wielu instancji składnika.

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

- Do funkcji obsługi zdarzeń jest przekazywany obiekt zdarzenia zawierający informacje o typie zdarzenia i instancji, która to zdarzenie rozgłosiła. Więcej informacji zawiera sekcja „[Informacje o obiekcie zdarzenia](#)” na stronie 26.
- Detektor pozostaje aktywny, dopóki aplikacja nie zakończy działania lub dopóki nie zostanie jawnie usunięty przy użyciu metody `removeEventListener()`. Na przykład poniższy wiersz kodu usuwa detektor zdarzenia `MouseEvent.CLICK` z instancji `aButton`:

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

**Informacje o obiekcie zdarzenia**

Obiekt zdarzenia dziedziczy z klasy obiektów `Event` i ma właściwości zawierające informacje o zdarzeniu, które zaszło, między innymi właściwości `target` oraz `type` udostępniające najważniejsze informacje o zdarzeniu:

Właściwość	Opis
<code>type</code>	Ciąg znaków określający typ zdarzenia.
<code>target</code>	Odwwołanie do instancji składnika, która rozgłosiła zdarzenia.

Jeśli zdarzenie ma dodatkowe właściwości, są one wymienione w opisie klasy zdarzenia w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

Obiekt zdarzenia jest generowany automatycznie i przekazywany do funkcji obsługi zdarzenia, gdy zajdzie dane zdarzenie.

Obiektu zdarzenia można używać wewnątrz funkcji, aby uzyskać dostęp do nazwy rozgłoszonego zdarzenia lub nazwy instancji składnika, która rozgłosiła zdarzenia. Znając nazwę instancji, można uzyskać dostęp do pozostałych właściwości składnika. Na przykład w poniższym kodzie użyto właściwości `target` obiektu zdarzenia `evtObj` w celu uzyskania dostępu do właściwości `label` składnika `aButton` i wyświetlenia jej wartości w panelu Wyjście:

```
import fl.controls.Button;
import flash.events.MouseEvent;

var aButton:Button = new Button();
aButton.label = "Submit";
addChild(aButton);
aButton.addEventListener(MouseEvent.CLICK, clickHandler);

function clickHandler(evtObj:MouseEvent) {
    trace("The " + evtObj.target.label + " button was clicked");
}
```

## Praca z listą wyświetlania

Wszystkie składniki w języku ActionScript 3.0 dziedziczą z klasy `DisplayObject`, a tym samym mają dostęp do metod i właściwości tej klasy umożliwiających interakcje z listą wyświetlania. *Lista wyświetlania* to hierarchia obiektów wyświetlanych i elementów wizualnych w aplikacji. Hierarchia ta obejmuje następujące elementy:

- stół montażowy, który jest kontenerem najwyższego poziomu;
- obiekty wyświetlane, do których należą między innymi kształty, klipy filmowe i pola tekstowe;
- kontenery obiektów wyświetlanych, będące szczególnymi typami obiektów wyświetlanych, które mogą zawierać podrzędne obiekty wyświetlane.

Kolejność obiektów na liście wyświetlania decyduje o ich głębokości w kontenerze nadrzędnym. Głębokość obiektu to jego pozycja liczona od najwyższej do najniższej warstwy stołu montażowego, od pierwszego planu w głąb stołu montażowego lub pozycja w ramach kontenera obiektu wyświetlanego. Różnice w głębokościach obiektów są widoczne, gdy obiekty się nakładają, ale istnieją nawet wtedy, gdy obiekty nie nakładają się. Każdy obiekt na liście wyświetlania ma określoną głębokość na stole montażowym. Aby zmienić głębokość obiektu, umieszczając go przed lub za określonymi innymi obiektami, należy zmienić jego pozycję na liście wyświetlania. Domyślna kolejność obiektów na liście wyświetlania jest identyczna z kolejnością, w jakiej były one umieszczane na stole montażowym. Pozycję 0 ma na liście wyświetlania ten obiekt, który znajduje się najgłębiej.

### Dodawanie składnika do listy wyświetlania

Możliwe jest dodawanie obiektów do obiektu `DisplayObjectContainer` poprzez wywołanie metody `addChild()` lub `addChildAt()` kontenera. W przypadku stołu montażowego możliwe jest także dodawanie obiektów do jego listy wyświetlania podczas tworzenia treści. Wystarczy utworzyć obiekt na stole montażowym lub przeciągnąć składnik z panelu Składniki na stół montażowy. Aby dodać obiekt do kontenera przy użyciu kodu w języku ActionScript, należy najpierw utworzyć instancję obiektu, wywołując jego konstruktor za pomocą operatora `new`, a następnie wywołać metodę `addChild()` lub `addChildAt()` w celu umieszczenia instancji na stole montażowym i na liście wyświetlania. Metoda `addChild()` umieszcza obiekt w następnej pozycji na liście wyświetlania, natomiast metoda `addChildAt()` umożliwia określenie pozycji, w której obiekt ma zostać dodany. W wypadku określenia pozycji, która jest już zajęta, obiekt o tej pozycji i obiekty powyżej niego są przenoszone w górę o 1 pozycję. Właściwość `numChildren` obiektu `DisplayObjectContainer` zawiera liczbę obiektów wyświetlanych zawartych w kontenerze. Można pobrać obiekt z listy wyświetlania, wywołując metodę `getChildAt()` i określając pozycję, lub — jeśli znana jest nazwa obiektu — wywołując metodę `getChildByName()`.

**Uwaga:** Dodając składnik przy użyciu kodu ActionScript, należy przypisać nazwę jego właściwości `name`, jeśli ma być możliwe odwoływanie się do składnika na liście wyświetlania za pośrednictwem jego nazwy.

Poniższy przykładowy kod powoduje wyświetlenie nazw i pozycji trzech składników na liście wyświetlania. Najpierw należy przeciągnąć składniki `NumericStepper`, `Button` i `ComboBox` na stół montażowy w taki sposób, aby nakładały się, po czym nadać im odpowiednio nazwy instancji `aNs`, `aButton` oraz `aCb`. Następnie należy dodać poniższy kod do panelu Operacje w klatce nr 1 na osi czasu:

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

W panelu Wyjście powinny pojawić się następujące wiersze:

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

## Przenoszenie składnika na liście wyświetlania

Położenie składnika na liście wyświetlania, a tym samym głębokość, na jakiej jest wyświetlany, można zmienić, wywołując metodę `addChildAt()` z parametrami określającymi nazwę obiektu oraz pozycję, na której ma być umieszczony. Na przykład dodanie poniższego kodu do poprzedniego przykładu powoduje umieszczenie składnika `NumericStepper` na wierzchu i powtórzenie pętli w celu wyświetlenia nowych pozycji składników na liście wyświetlania:

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

W panelu Wyjście powinny pojawić się następujące informacje:

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

Ponadto składnik `NumericStepper` powinien pojawić się przed pozostałymi składnikami na ekranie.

Należy zwrócić uwagę, że właściwość `numChildren` określa liczbę obiektów (od 1 do  $n$ ) na liście wyświetlania, natomiast pierwsza pozycja na liście ma indeks 0. Jeśli zatem lista zawiera trzy obiekty, trzeci obiekt ma indeks 2. Oznacza to, że do ostatniej pozycji na liście wyświetlania lub do najwyżej położonego obiektu na ekranie można odwołać się, korzystając z indeksu `numChildren - 1`.

## Usuwanie składnika z listy wyświetlania

W celu usunięcia składnika z kontenera obiektów wyświetlanych i jego listy wyświetlania można użyć metody `removeChild()` lub `removeChildAt()`. Poniższy przykładowy kod powoduje umieszczenie trzech składników `Button`, tak aby pokrywały się na stole montażowym, i dodanie detektora zdarzeń dla każdego z nich. Kliknięcie każdego z przycisków sprawia, że podprogram obsługi zdarzeń usuwa odpowiedni składnik `Button` z listy wyświetlania i ze stołu montażowego.

- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij przycisk z panelu Składniki do panelu Biblioteka.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i dodaj następujący kod:



**Praca ze składnikami**

```
import fl.controls.Button;

var i:int = 0;
while(i++ < 3) {
    makeButton(i);
}
function removeButton(event:MouseEvent):void {
    removeChildAt(numChildren -1);
}
function makeButton(num) {
    var aButton:Button = new Button();
    aButton.name = "Button" + num;
    aButton.label = aButton.name;
    aButton.move(200, 200);
    addChild(aButton);
    aButton.addEventListener(MouseEvent.CLICK, removeButton);
}
```

Pełne wyjaśnienie działania listy wyświetlania zawiera sekcja „Programowanie wyświetlania” w podręczniku *Programowanie w języku ActionScript 3.0*.

## Praca z klasą FocusManager

Gdy użytkownik naciśnie klawisz Tab w celu nawigowania w aplikacji Flash lub kliknie wewnątrz aplikacji, klasa FocusManager określa, który składnik powinien stać się aktywny. Aby aktywować klasę FocusManager, nie trzeba dodawać do aplikacji instancji tej klasy ani pisać żadnego dodatkowego kodu. Wyjątkiem są przypadki samodzielnego tworzenia składników.

Gdy aktywny staje się obiekt RadioButton, klasa FocusManager analizuje ten obiekt i wszystkie obiekty o tej samej wartości groupName, po czym uaktywnia obiekt, którego właściwość selected jest ustawiona na true.

Każdy modalny składnik Window zawiera instancję klasy FocusManager, a zatem elementy sterujące tego okna stają się jego własnym zestawem obiektów aktywowanych klawiszem Tab. W efekcie użytkownik nie może, naciskając klawisz Tab, przypadkowo przejść do składników w innych oknach.

W charakterze domyślnego systemu nawigacji lub *pętli nawigacji klawiszem Tab*, klasa FocusManager używa hierarchii głębokości elementów w kontenerze (ich kolejności z). Użytkownik zazwyczaj nawiguje w pętli, korzystając z klawisza Tab, co powoduje przenoszenie aktywności od pierwszego składnika do ostatniego, a potem z powrotem do pierwszego. Głębokość zależy przede wszystkim od kolejności, w jakiej składniki są przeciągane na stół montażowy; możliwe jest jednak również użycie poleceń Modyfikuj > Ułóż > Przesuń na wierzch/Przesuń na spód do zdefiniowania ostatecznej kolejności z. Więcej informacji na temat poziomów głębokości zawiera sekcja „Praca z listą wyświetlania” na stronie 27.

Istnieje możliwość wywołania metody setFocus() w celu uaktywnienia określonej instancji składnika w aplikacji. Poniższy kod przykładowy tworzy instancję klasy FocusManager dla bieżącego kontenera (this) i uaktywnia instancję składnika Button o nazwie aButton.

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

**Praca ze składnikami**

Metoda `getFocus()` umożliwia sprawdzenie, który składnik jest w danej chwili aktywny, natomiast metoda `getNextFocusManagerComponent()` umożliwia sprawdzenie, który składnik w pętli Tab zostanie uaktywniony jako następny. W poniższym przykładzie składniki `CheckBox`, `RadioButton` i `Button` znajdują się na stole montażowym, a każdy składnik ma detektory zdarzeń `MouseEvent.CLICK` i `FocusEvent.MOUSE_FOCUS_CHANGE`. Gdy w wyniku kliknięcia składnika przez użytkownika zostanie wygenerowane zdarzenie `MouseEvent.CLICK`, funkcja `showFocus()` wywołuje metodę `getNextFocusManagerComponent()` w celu określenia, który składnik w pętli klawisza Tab zostałby uaktywniony jako następny. Następnie wywołuje metodę `setFocus()` w celu uaktywnienia tego właśnie składnika. Gdy zostanie wygenerowane zdarzenie `FocusEvent.MOUSE_FOCUS_CHANGE`, funkcja `fc()` wyświetla nazwę składnika, w którym zdarzenie to wystąpiło. To zdarzenie jest wywoływane, gdy użytkownik kliknie składnik inny niż następny składnik w pętli klawisza Tab.

```
// This example assumes a CheckBox (aCh), a RadioButton (aRb) and a Button
// (aButton) have been placed on the Stage.
```

```
import fl.managers.FocusManager;
import flash.display.InteractiveObject;

var fm:FocusManager = new FocusManager(this);

aCh.addEventListener(MouseEvent.CLICK, showFocus);
aRb.addEventListener(MouseEvent.CLICK, showFocus);
aButton.addEventListener(MouseEvent.CLICK, showFocus);
aCh.addEventListener(FocusEvent.MOUSE_FOCUS_CHANGE, fc);
aRb.addEventListener(FocusEvent.MOUSE_FOCUS_CHANGE, fc);
aButton.addEventListener(FocusEvent.MOUSE_FOCUS_CHANGE, fc);

function showFocus(event:MouseEvent):void {
    var nextComponent:InteractiveObject = fm.getNextFocusManagerComponent();
    trace("Next component in tab loop is: " + nextComponent.name);
    fm.setFocus(nextComponent);
}

function fc(fe:FocusEvent):void {
    trace("Focus Change: " + fe.target.name);
}
```

Aby utworzyć składnik `Button`, który będzie uaktywniany po naciśnięciu klawisza `Enter` (Windows) lub `Return` (Macintosh), należy przypisać właściwości `FocusManager.defaultButton` instancję składnika `Button`, która ma być przyciskiem domyślnym; ilustruje to poniższy kod:

```
import fl.managers.FocusManager;

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

Klasa `FocusManager` przesłania domyślny prostokąt aktywności programu Flash Player i rysuje niestandardowy prostokąt aktywności z zaokrąglonymi narożnikami.

Więcej informacji na temat tworzenia schematu aktywności w aplikacjach Flash zawiera opis klasy `FocusManager` w [dokumentacji języka ActionScript 3.0 dla programu Flash Professional](#). Aby utworzyć własnego menedżera aktywności, należy utworzyć klasę implementującą interfejs `IFocusManager`. Więcej informacji zawiera opis klasy `IFocusManager` w [dokumentacji języka ActionScript 3.0 dla programu Flash Professional](#).

## Praca ze składnikami opartymi na liście

Składniki List, DataGrid i TileList dziedziczą z klasy bazowej SelectableList. Z tego powodu zalicza się je do składników opartych na liście. Składnik ComboBox zawiera pole tekstowe oraz listę, dlatego również uznawany jest za składnik oparty na liście.

Lista składa się z wierszy. Składniki DataGrid i TileList są złożone z wierszy, które mogą być podzielone na wiele kolumn. Przecięcie wiersza i kolumny to komórka. W przypadku listy, która jest pojedynczą kolumną wierszy, każdy wiersz jest jednocześnie komórką. Należy wspomnieć o dwóch ważnych cechach komórek:

- Wartości danych zapisane w komórkach nazywane są elementami. *Element* jest to obiekt języka ActionScript służący do przechowywania jednostek informacji w ramach listy. Listę można traktować jak tablicę, w której każda indeksowana pozycja jest jednym elementem. Element listy to obiekt, który zwykle zawiera wyświetlaną właściwość `label` oraz właściwość `data` służącą do przechowywania danych. *Dostawca danych* jest to model danych zapisanych w elementach listy. Dostawca danych umożliwia wypełnienie składnika opartego na liście poprzez proste przypisanie do właściwości `dataProvider` składnika.
- Komórka może zawierać dane różnego typu, od tekstu po obrazy, klipy filmowe i obiekty należące do dowolnych klas utworzonych przez użytkownika. Dlatego komórka musi być rysowana, czyli renderowana, w sposób właściwy dla jej zawartości. W związku z tym składniki oparte na liście mają własne *mechanizmy renderujące* zawartość komórek. W przypadku składnika DataGrid każda kolumna jest obiektem `DataGridColumn`, który ma również właściwość `cellRenderer`, dzięki czemu każda kolumna może być renderowana inaczej, w sposób właściwy dla jej zawartości.

Wszystkie składniki oparte na liście mają właściwości `cellRenderer` i `dataProvider`, którym można przypisać mechanizmy renderujące komórki i dostawców danych. Informacje na temat tych właściwości i pracy ze składnikami opartymi na liście zawierają sekcje „Praca z dostawcą danych” na stronie 31 oraz „Praca z mechanizmem renderującym komórki” na stronie 39.

## Praca z dostawcą danych

Dostawca danych to źródło danych dla składników ComboBox, DataGrid, List i TileList. Każda z tych klas składników ma właściwość `dataProvider`, do której można przypisać obiekt `DataProvider` w celu wypełnienia komórek składnika danymi. Zazwyczaj dostawca danych jest kolekcją danych, np. obiektem `Array` lub `XML`.

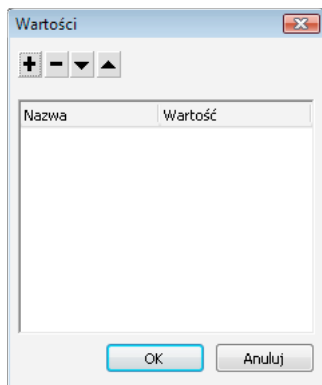
### Tworzenie dostawcy danych

Dostawcę danych dla składników ComboBox, List i TileList można utworzyć, korzystając z parametru `dataProvider` w środowisku tworzenia treści. Składnik DataGrid nie udostępnia parametru `dataProvider` w Inspektorze właściwości, ponieważ może mieć kilka kolumn i jego dostawca danych jest odpowiednio bardziej skomplikowany. Obiekty dostawców danych dla powyższych składników oraz dla składnika DataGrid można tworzyć także za pomocą kodu ActionScript.

### Korzystanie z parametru `dataProvider`

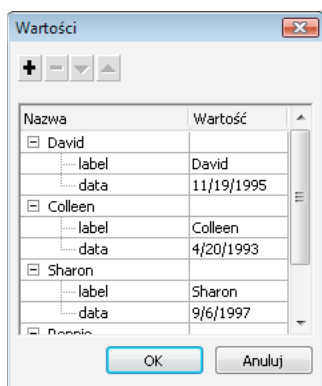
Prosty obiekt dostawcy danych dla składników ComboBox, List i TileList można utworzyć, klikając parametr `dataProvider` na karcie Parametry w Inspektorze właściwości lub Inspektorze składników.

Dwukrotne kliknięcie wartości w komórce, która początkowo zawiera pustą tabelę, spowoduje otwarcie okna dialogowego Wartości umożliwiającego utworzenie dostawcy danych poprzez wprowadzenie wielu wartości etykiet i danych.

**Praca ze składnikami**

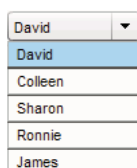
Okno dialogowe Wartości dostawcy danych

Klikając znak plus, można dodać element do dostawcy danych. Kliknięcie znaku minus powoduje usunięcie elementu. Kliknięcie strzałki w górę powoduje przeniesienie zaznaczonego elementu bliżej początku listy, a kliknięcie strzałki w dół — bliżej końca listy. Na poniższej ilustracji przedstawiono okno dialogowe Wartości, w którym tworzona jest lista imion dzieci oraz ich dat urodzenia.



Okno dialogowe Wartości z danymi

Utworzona tablica (obiekt Array) zawiera pary pól etykiet i wartości. Pola etykiet to `label` i `data`, natomiast pola wartości to imiona i daty urodzenia dzieci. Pole etykiety identyfikuje zawartość komórek wyświetlaną na liście, czyli w tym przypadku imiona dzieci. Wynikowy składnik ComboBox wygląda następująco:



Składnik ComboBox wypełniony przez dostawcę danych

Po zakończeniu dodawania danych należy kliknąć przycisk OK, aby zamknąć okno dialogowe. Tablica w parametrze `dataProvider` jest teraz wypełniona elementami utworzonymi przez użytkownika.

**Praca ze składnikami**

allowMultipleSelection	false
dataProvider	[(label:David,data:11/19/1995), (label:Colleen,data:4/20/1993), (label:Sharon,data:9/6/1997)];
horizontalLineScrollSize	1
horizontalPageScrollSize	0
horizontalScrollPolicy	auto
verticalLineScrollSize	1

Parametr `dataProvider` z danymi

Dostęp do utworzonych wartości etykiet i danych można uzyskać w języku ActionScript za pośrednictwem właściwości `dataProvider` składnika.

**Tworzenie dostawcy danych przy użyciu kodu ActionScript**

Dostawcę danych można utworzyć, tworząc dane w obiekcie Array lub XML i przekazując ten obiekt jako parametr `value` do konstruktora obiektu `DataProvider`.

**Uwaga:** W języku ActionScript 3.0 nie jest możliwe bezpośrednie przypisanie obiektu Array lub XML do właściwości `dataProvider`, ponieważ właściwość jest zdefiniowana jako obiekt typu `DataProvider` i można jej przypisywać tylko obiekty tego typu.

W poniższym przykładzie składnik `List`, będący pojedynczą kolumną wierszy, jest wypełniany imionami i datami urodzin kilkorga dzieci. W przykładowym kodzie lista jest definiowana w tablicy `items` i przekazywana jako parametr podczas tworzenia instancji obiektu `DataProvider` (`new DataProvider(items)`); następnie instancja ta jest przypisywana właściwości `dataProvider` składnika `List`.

```
import fl.controls.List;
import fl.data.DataProvider;

var aList:List = new List();
var items:Array = [
    {label:"David", data:"11/19/1995"},
    {label:"Colleen", data:"4/20/1993"},
    {label:"Sharon", data:"9/06/1997"},
    {label:"Ronnie", data:"7/6/1993"},
    {label:"James", data:"2/15/1994"},
];
aList.dataProvider = new DataProvider(items);
addChild(aList);
aList.move(150,150);
```

Obiekt `Array` składa się z par pól etykiet i wartości. Pola etykiet to `label` i `data`, natomiast pola wartości to imiona i daty urodzenia dzieci. Pole etykiety identyfikuje zawartość komórek wyświetlaną na liście, czyli w tym przypadku imiona dzieci. Wynikowy składnik `List` wygląda tak:

David
Colleen
Sharon
Ronnie
James

Lista wypełniona przez dostawcę danych

Wartość pola danych jest dostępna, gdy użytkownik wybierze element na liście, klikając go i wywołując tym samym zdarzenie `change`. W poniższym przykładzie do poprzedniego przykładowego kodu dodano składnik `TextArea` (`aTa`) oraz funkcję obsługi zdarzeń (`changeHandler`), które wyświetlają datę urodzin dziecka, gdy użytkownik wybierze imię na liście.

**Praca ze składnikami**

```

import fl.controls.List;
import fl.controls.TextArea;
import flash.events.Event;
import fl.data.DataProvider;

var aList:List = new List();
var aTa:TextArea = new TextArea();
var items:Array = [
    {label:"David", data:"1/19/1995"},
    {label:"Colleen", data:"4/20/1993"},
    {label:"Sharon", data:"9/06/1994"},
    {label:"Ronnie", data:"7/6/1993"},
    {label:"James", data:"2/15/1994"},
];
aList.dataProvider = new DataProvider(items);

addChild(aList);
addChild(aTa);

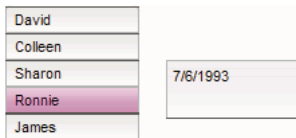
aList.move(150,150);
aTa.move(150, 260);

aList.addEventListener(Event.CHANGE, changeHandler);

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

```

Teraz, gdy użytkownik wybierze imię dziecka na liście, w obszarze tekstowym pojawi się data urodzin dziecka, tak jak przedstawiono to na poniższej ilustracji. Zadanie to realizuje funkcja `changeHandler()`, gdy przypisuje właściwość `text` składnika `TextArea` (`aTa.text`) wartość elementu danych (`event.target.selectedItem.data`). Właściwość `event.target` to obiekt, który wywołał zdarzenie, czyli w tym przypadku obiekt `List`.



Wyświetlanie pola danych z dostawcy danych składnika `List`

Dostawca danych może zawierać dane inne niż tekst. W poniższym przykładzie obiekt `DataProvider`, dostarczający danych dla składnika `TitleList`, zawiera klipy filmowe. Kod buduje obiekt `DataProvider`, wywołując metodę `addItem()` w celu dodania kolejnego elementu po utworzeniu każdego kolejnego obiektu `MovieClip` (kolorowego prostokąta).

**Praca ze składnikami**

```

import fl.data.DataProvider;
import flash.display.DisplayObject;

var aBox:MovieClip = new MovieClip();
var i:uint = 0;
var colors:Array = new Array(0x000000, 0xFF0000, 0x0000CC, 0x00CC00, 0xFFFF00);
var colorNames:Array = new Array("Midnight", "Cranberry", "Sky", "Forest", "July");
var dp:DataProvider = new DataProvider();
for(i=0; i < colors.length; i++) {
    drawBox(aBox, colors[i]); // draw box w next color in array
    dp.addItem( {label:colorNames[i], source:aBox} );
}
aTl.dataProvider = dp;
aTl.columnWidth = 110;
aTl.rowHeight = 130;
aTl.setSize(280,150);
aTl.move(150, 150);
aTl.setStyle("contentPadding", 5);

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1.0);
    box.graphics.drawRect(0, 0, 100, 100);
    box.graphics.endFill();
}

```

Obiekt DataProvider można również zapełnić danymi XML (a nie tablicą). Poniższy przykładowy kod zapisuje dane w obiekcie XML o nazwie employeesXML, a następnie przekazuje ten obiekt jako parametr value do funkcji-konstruktor DataProvider():

```

import fl.controls.DataGrid;
import fl.data.DataProvider;

var aDg:DataGrid = new DataGrid();
addChild(aDg);

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

var myDP:DataProvider = new DataProvider(employeesXML);

aDg.columns = ["Name", "ID"];
aDg.dataProvider = myDP;

```

Dane mogą być zapisane jako atrybuty XML (tak jak w poprzednim przykładzie) lub jako właściwości XML, co ilustruje poniższy przykład:

**Praca ze składnikami**

```

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

```

Obiekt `DataProvider` ma także zestaw metod i właściwości umożliwiających dostęp do niego i manipulowanie nim. Interfejs API dostawcy danych umożliwia dodawanie, usuwanie, zastępowanie, sortowanie i scalanie elementów w obiekcie `DataProvider`.

**Manipulowanie dostawcą danych**

Istnieje możliwość dodawania elementów do obiektu `DataProvider` za pomocą metod `addItem()` i `addItemAt()`. W poniższym przykładzie dodawane są elementy wprowadzane przez użytkownika w polu tekstowym edytowalnego składnika `ComboBox`. Przyjęto, że składnik `ComboBox` został już przeciągnięty na stół montażowy i że nadano mu nazwę instancji `aCb`.

```

import fl.data.DataProvider;
import fl.events.ComponentEvent;

var items:Array = [
    {label:"Roger"},
    {label:"Carolyn"},
    {label:"Darrell"},
    {label:"Rebecca"},
    {label:"Natalie"},
    {label:"Mitchell"},
];
aCb.dataProvider = new DataProvider(items);

aCb.addEventListener(ComponentEvent.ENTER, newItemHandler);

function newItemHandler(event:ComponentEvent):void {
    var newRow:int = event.target.length + 1;
    event.target.addItemAt({label:event.target.selectedLabel},
        event.target.length);
}

```

Możliwe jest także zastępowanie i usuwanie elementów w składniku za pośrednictwem jego dostawcy danych. W poniższym przykładzie zaimplementowano dwa odrębne składniki `List`, `listA` i `listB`, oraz udostępniono składnik `Button` z etykietą `Sync`. Gdy użytkownik kliknie przycisk (składnik `Button`), wywoływana jest metoda `replaceItemAt()` zastępująca elementy listy `listB` elementami listy `listA`. Jeśli lista `listA` jest dłuższa niż `listB`, wywoływana jest metoda `addItem()` w celu dodania dodatkowych elementów do listy `listB`. Jeśli lista `listB` jest dłuższa niż `listA`, wywoływana jest metoda `removeItemAt()` w celu usunięcia nadmiarowych elementów z listy `ListB`.



```
// Requires the List and Button components to be in the library

import fl.controls.List;
import fl.controls.Button;
import flash.events.Event;
import fl.data.DataProvider;

var listA:List = new List();
var listB:List = new List();
var syncButton:Button = new Button();
syncButton.label = "Sync";

var itemsA:Array = [
    {label:"David"},
    {label:"Colleen"},
    {label:"Sharon"},
    {label:"Ronnie"},
    {label:"James"},
];
var itemsB:Array = [
    {label:"Roger"},
    {label:"Carolyn"},
    {label:"Darrell"},
    {label:"Rebecca"},
    {label:"Natalie"},
    {label:"Mitchell"},
];
listA.dataProvider = new DataProvider(itemsA);
listB.dataProvider = new DataProvider(itemsB);

addChild(listA);
addChild(listB);
addChild(syncButton);

listA.move(100, 100);
listB.move(250, 100);
syncButton.move(175, 220);

syncButton.addEventListener(MouseEvent.CLICK, syncHandler);

function syncHandler(event:MouseEvent):void {
    var i:uint = 0;
    if(listA.length > listB.length) { //if listA is longer, add items to B
        while(i < listB.length) {
            listB.dataProvider.replaceItemAt(listA.dataProvider.getItemAt(i), i);
        }
    }
}
```

**Praca ze składnikami**

```

        ++i;
    }
    while(i < listA.length) {
        listB.dataProvider.addItem(listA.dataProvider.getItemAt(i++));
    }
} else if(listA.length == listB.length) { //if listA and listB are equal length
    while(i < listB.length) {
        listB.dataProvider.replaceItemAt(listA.dataProvider.getItemAt(i), i);
        ++i;
    }
} else { //if listB is longer, remove extra items from B
    while(i < listA.length) {
        listB.dataProvider.replaceItemAt(listA.dataProvider.getItemAt(i), i);
        ++i;
    }
    while(i < listB.length) {
        listB.dataProvider.removeItemAt(i++);
    }
}
}
}
}

```

Metody `merge()`, `sort()` i `sortOn()` obiektu `DataProvider` umożliwiają scalanie i sortowanie dostawców danych. W poniższym przykładzie dwie instancje klasy `DataGrid` (`aDg` i `bDg`) wypełniane są niepełnymi składami drużyn softballowych. Dodawany jest składnik `Button` z etykietą `Merge (Scal)`, a gdy użytkownik kliknie ten składnik, funkcja obsługi zdarzenia (`mrgHandler`) scala skład `bDg` ze składem `aDg` i sortuje wynikowy składnik `DataGrid` według zawartości kolumny `Name`.

```

import fl.data.DataProvider;
import fl.controls.DataGrid;
import fl.controls.Button;

var aDg:DataGrid = new DataGrid();
var bDg:DataGrid = new DataGrid();
var mrgButton:Button = new Button();
addChild(aDg);
addChild(bDg);
addChild(mrgButton);
bldRosterGrid(aDg);
bldRosterGrid(bDg);
var aRoster:Array = new Array();
var bRoster:Array = new Array();
aRoster = [
    {Name:"Wilma Carter", Bats:"R", Throws:"R", Year:"So", Home: "Redlands, CA"},
    {Name:"Sue Pennypacker", Bats:"L", Throws:"R", Year:"Fr", Home: "Athens, GA"},
    {Name:"Jill Smithfield", Bats:"R", Throws:"L", Year:"Sr", Home: "Spokane, WA"},
    {Name:"Shirley Goth", Bats:"R", Throws:"R", Year:"Sr", Home: "Carson, NV"}
];
bRoster = [
    {Name:"Angelina Davis", Bats:"R", Throws:"R", Year:"So", Home: "Odessa, TX"},
    {Name:"Maria Santiago", Bats:"L", Throws:"L", Year:"Sr", Home: "Tacoma, WA"},
    {Name:"Debbie Ferguson", Bats:"R", Throws:"R", Year: "Jr", Home: "Bend, OR"}
];
aDg.dataProvider = new DataProvider(aRoster);
bDg.dataProvider = new DataProvider(bRoster);
aDg.move(50,50);
aDg.rowCount = aDg.length;

```

**Praca ze składnikami**

```
bDg.move(50,200);
bDg.rowCount = bDg.length;
mrgButton.label = "Merge";
mrgButton.move(200, 315);
mrgButton.addEventListener(MouseEvent.CLICK, mrgHandler);

function bldRosterGrid(dg:DataGrid) {
    dg.setSize(400, 300);
    dg.columns = ["Name", "Bats", "Throws", "Year", "Home"];
    dg.columns[0].width = 120;
    dg.columns[1].width = 50;
    dg.columns[2].width = 50;
    dg.columns[3].width = 40;
    dg.columns[4].width = 120;
};

function mrgHandler(event:MouseEvent):void {
    aDg.dataProvider.merge(bDg.dataProvider);
    aDg.dataProvider.sortOn("Name");
}
```

Więcej informacji zawiera opis klasy `DataProvider` w [dokumentacji języka ActionScript 3.0](#).

## Praca z mechanizmem renderującym komórki

`CellRenderer` — mechanizm renderujący komórki — to klasa, której składniki oparte na liście, takie jak `List`, `DataGrid`, `TileList` i `ComboBox`, używają do wyświetlania niestandardowej zawartości komórek wierszy oraz manipulowania tą zawartością. Niestandardowa komórka może zawierać tekst, gotowy składnik, np. `CheckBox`, lub dowolny obiekt wyświetlany należący do klasy zdefiniowanej przez użytkownika. Aby renderować dane przy użyciu niestandardowego mechanizmu renderującego komórki, można rozszerzyć klasę `CellRenderer` lub zaimplementować interfejs `ICellRenderer` w celu utworzenia własnego niestandardowego mechanizmu renderującego.

Klasy `List`, `DataGrid`, `TileList` i `ComboBox` są podklasami klasy `SelectableList`. Klasa `SelectableList` zawiera styl `cellRenderer`. Styl ten definiuje obiekt wyświetlany używany przez składnik do renderowania komórek.

Formatowanie stylów używanych przez mechanizm renderujący komórki można dostosować, wywołując metodę `setRendererStyle()` obiektu `List` (patrz „[Formatowanie komórek](#)” na stronie 39). Alternatywnym rozwiązaniem jest zdefiniowanie klasy niestandardowej, która posłuży za mechanizm renderujący komórki (patrz „[Definiowanie niestandardowej klasy mechanizmu renderującego komórki](#)” na stronie 40).

### Formatowanie komórek

Klasa `CellRenderer` zawiera szereg stylów umożliwiających sterowanie formatem komórek.

Poniższe style umożliwiają definiowanie karnacji obowiązujących dla różnych stanów komórki (wyłączona, naciśnięta, pod wskaźnikiem myszy i nienaciśnięta):

- `disabledSkin` i `SelectedDisabledSkin`
- `downSkin` i `selectedDownSkin`
- `overSkin` i `selectedOverSkin`
- `upSkin` i `selectedUpSkin`

Następujące style mają zastosowanie do formatowania tekstu:

- disabledTextFormat
- textFormat
- textPadding

Style te można ustawiać, wywołując metodę `setRendererStyle()` obiektu `List` lub metodę `setStyle()` obiektu `CellRenderer`. Do odczytywania tych stylów służy metoda `getRendererStyle()` obiektu `List` oraz metoda `getStyle()` obiektu `CellRenderer`. Można również uzyskać dostęp do obiektu, który definiuje wszystkie style mechanizmu renderującego (jako nazwane właściwości obiektu) za pośrednictwem właściwości `rendererStyles` obiektu `List` lub metody `getStyleDefinition()` obiektu `CellRenderer`.

Metoda `clearRendererStyle()` przywraca wartości domyślne wszystkich stylów.

Aby odczytać lub ustawić wysokość wierszy listy, należy skorzystać z właściwości `rowHeight` obiektu `List`.

## Definiowanie niestandardowej klasy mechanizmu renderującego komórki

### Tworzenie klasy rozszerzającej klasę `CellRenderer` w celu zdefiniowania niestandardowego mechanizmu renderującego

Poniższy przykładowy kod zawiera dwie klasy. Klasa `ListSample` tworzy instancję składnika `List` i wykorzystuje drugą klasę, `CustomRenderer`, do zdefiniowania mechanizmu renderującego komórki używanego ze składnikiem `List`. Klasa `CustomRenderer` rozszerza klasę `CellRenderer`.

- 1 Wybierz polecenie Plik > Nowy.
- 2 W wyświetlonym oknie dialogowym Nowy dokument wybierz typ Plik Flash (ActionScript 3.0), a następnie kliknij przycisk OK.
- 3 Wybierz polecenie Okno > Składniki, aby wyświetlić panel Składniki.
- 4 Przeciągnij składnik `List` z panelu Składniki na stół montażowy.
- 5 Jeśli w programie Flash nie zostanie wyświetlony Inspektor właściwości, wybierz polecenie Okno > Właściwości > Właściwości.
- 6 Po zaznaczeniu składnika `List` ustaw właściwości w Inspektorze właściwości:
  - Nazwa instancji: `myList`
  - Sz (szerokość): 200
  - W (wysokość): 300
  - X: 20
  - Y: 20
- 7 Na osi czasu wybierz klatkę nr 1 warstwy nr 1 i wybierz polecenie Okno > Operacje.
- 8 W panelu Operacje wpisz następujący skrypt:

```
myList.setStyle("cellRenderer", CustomCellRenderer);
myList.addItem({label:"Burger -- $5.95"});
myList.addItem({label:"Fries -- $1.95"});
```
- 9 Wybierz polecenie Plik > Zapisz. Nadaj plikowi nazwę i kliknij przycisk OK.
- 10 Wybierz polecenie Plik > Nowy.
- 11 W wyświetlonym oknie dialogowym Nowy dokument wybierz typ Plik ActionScript i kliknij przycisk OK.
- 12 W oknie skryptu wprowadź następujący kod w celu zdefiniowania klasy `CustomCellRenderer`:

**Praca ze składnikami**

```

package {
    import fl.controls.listClasses.CellRenderer;
    import flash.text.TextFormat;
    import flash.filters.BevelFilter;
    public class CustomCellRenderer extends CellRenderer {
        public function CustomCellRenderer() {
            var format:TextFormat = new TextFormat("Verdana", 12);
            setStyle("textFormat", format);
            this.filters = [new BevelFilter()];
        }
    }
}

```

13 Wybierz polecenie Plik > Zapisz. Nadaj plikowi nazwę CustomCellRenderer.as, umieść go w tym samym katalogu, co plik FLA, a następnie kliknij przycisk OK.

14 Wybierz polecenie Sterowanie > Testuj film.

### Użycie klasy implementującej interfejs ICellRenderer w celu zdefiniowania niestandardowego mechanizmu renderującego

Mechanizm renderujący komórki można również zdefiniować, używając dowolnej klasy, która dziedziczy z klasy DisplayObject i implementuje interfejs ICellRenderer. Poniższy przykładowy kod definiuje dwie klasy. Klasa ListSample2 dodaje obiekt List do listy wyświetlania i definiuje własny mechanizm renderujący komórki wykorzystujący klasę CustomRenderer. Klasa CustomRenderer rozszerza klasę CheckBox (która z kolei rozszerza klasę DisplayObject) i implementuje interfejs ICellRenderer. Należy zwrócić uwagę, że klasa CustomRenderer definiuje metody pobierające i ustawiające dla właściwości data i listData zdefiniowanych w interfejsie ICellRenderer. Pozostałe właściwości i metody zdefiniowane w interfejsie ICellRenderer (właściwość selected i metoda setSize()) są już zdefiniowane w klasie CheckBox:

- 1 Wybierz polecenie Plik > Nowy.
- 2 W wyświetlonym oknie dialogowym Nowy dokument wybierz typ Plik Flash (ActionScript 3.0), a następnie kliknij przycisk OK.
- 3 Wybierz polecenie Okno > Składniki, aby wyświetlić panel Składniki.
- 4 Przeciągnij składnik List z panelu Składniki na stół montażowy.
- 5 Jeśli w programie Flash nie zostanie wyświetlony Inspektor właściwości, wybierz polecenie Okno > Właściwości > Właściwości.
- 6 Po zaznaczeniu składnika List ustaw właściwości w Inspektorze właściwości:
  - Nazwa instancji: myList
  - Sz (szerokość): 100
  - W (wysokość): 300
  - X: 20
  - Y: 20
- 7 Na osi czasu wybierz klatkę nr 1 warstwy nr 1 i wybierz polecenie Okno > Operacje.
- 8 W panelu Operacje wpisz następujący skrypt.

```

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

```

- 9 Wybierz polecenie Plik > Zapisz. Nadaj plikowi nazwę i kliknij przycisk OK.

10 Wybierz polecenie Plik > Nowy.

11 W wyświetlonym oknie dialogowym Nowy dokument wybierz typ Plik ActionScript i kliknij przycisk OK.

12 W oknie skryptu wprowadź następujący kod w celu zdefiniowania klasy CustomCellRenderer:

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

13 Wybierz polecenie Plik > Zapisz. Nadaj plikowi nazwę CustomCellRenderer.as, umieść go w tym samym katalogu, co plik FLA, a następnie kliknij przycisk OK.

14 Wybierz polecenie Sterowanie > Testuj film.

### Definiowanie mechanizmu renderującego komórki przy użyciu symbolu

Do zdefiniowania mechanizmu renderującego komórki można także użyć symbolu z biblioteki. Symbol musi być wyeksportowany do użycia w kodzie ActionScript, a z nazwą klasy symbolu bibliotecznego musi być skojarzony plik klasy, która albo implementuje interfejs ICellRenderer, albo rozszerza klasę CellRenderer (bądź jedną z jej podklas).

W poniższym przykładzie niestandardowy mechanizm renderujący komórki jest definiowany przy użyciu symbolu z biblioteki.

1 Wybierz polecenie Plik > Nowy.

2 W wyświetlonym oknie dialogowym Nowy dokument wybierz typ Plik Flash (ActionScript 3.0), a następnie kliknij przycisk OK.

3 Wybierz polecenie Okno > Składniki, aby wyświetlić panel Składniki.

4 Przeciągnij składnik List z panelu Składniki na stół montażowy.

5 Jeśli w programie Flash nie zostanie wyświetlony Inspektor właściwości, wybierz polecenie Okno > Właściwości > Właściwości.

6 Po zaznaczeniu składnika List ustaw właściwości w Inspektorze właściwości:

- Nazwa instancji: myList

- Sz (szerokość): 100
  - W (wysokość): 400
  - X: 20
  - Y: 20
- 7 Kliknij panel Parametry, a następnie kliknij dwukrotnie drugą kolumnę w wierszu dataProvider.
- 8 W wyświetlonym oknie dialogowym Wartości kliknij dwa razy znak plus, aby dodać dwa elementy danych (z etykietami label0 i label1), a następnie kliknij przycisk OK.
- 9 Za pomocą narzędzia Tekst narysuj pole tekstowe na stole montażowym.
- 10 Po zaznaczeniu pola tekstowego ustaw właściwości w Inspektorze właściwości:
- Typ tekstu: Tekst dynamiczny
  - Nazwa instancji: textField
  - Sz (szerokość): 100
  - Rozmiar czcionki: 24
  - X: 0
  - Y: 0
- 11 Po zaznaczeniu pola tekstowego wybierz polecenie Modyfikuj > Konwertuj na symbol.
- 12 W oknie dialogowym Konwertuj na symbol wybierz następujące ustawienia i kliknij przycisk OK.
- Nazwa: MyCellRenderer
  - Typ: MovieClip
  - Eksportuj dla ActionScript: zaznaczone
  - Eksportuj w pierwszej klatce: zaznaczone
  - Klasa: MyCellRenderer
  - Klasa bazowa: flash.display.MovieClip
- Jeśli zostanie wyświetlone ostrzeżenie dotyczące klasy ActionScript, kliknij przycisk OK w oknie ostrzeżenia.
- 13 Usuń instancję nowego symbolu klipu filmowego ze stołu montażowego.
- 14 Na osi czasu wybierz klatkę nr 1 warstwy nr 1 i wybierz polecenie Okno > Operacje.
- 15 W panelu Operacje wpisz następujący skrypt.
- ```
myList.setStyle("cellRenderer", MyCellRenderer);
```
- 16 Wybierz polecenie Plik > Zapisz. Nadaj plikowi nazwę i kliknij przycisk OK.
- 17 Wybierz polecenie Plik > Nowy.
- 18 W wyświetlonym oknie dialogowym Nowy dokument wybierz typ Plik ActionScript i kliknij przycisk OK.
- 19 W oknie skryptu wprowadź następujący kod w celu zdefiniowania klasy MyCellRenderer:

```
package {
    import flash.display.MovieClip;
    import flash.filters.GlowFilter;
    import flash.text.TextField;
    import fl.controls.listClasses.ICellRenderer;
    import fl.controls.listClasses.ListData;
    import flash.utils.setInterval;
    public class MyCellRenderer extends MovieClip implements ICellRenderer {
        private var _listData:ListData;
        private var _data:Object;
        private var _selected:Boolean;
        private var glowFilter:GlowFilter;
        public function MyCellRenderer() {
            glowFilter = new GlowFilter(0xFFFF00);
            setInterval(toggleFilter, 200);
        }
        public function set data(d:Object):void {
            _data = d;
            textField.text = d.label;
        }
        public function get data():Object {
            return _data;
        }
        public function set listData(ld:ListData):void {
            _listData = ld;
        }
        public function get listData():ListData {
            return _listData;
        }
        public function set selected(s:Boolean):void {
            _selected = s;
        }
        public function get selected():Boolean {
            return _selected;
        }
        public function setSize(width:Number, height:Number):void {
        }
        public function setStyle(style:String, value:Object):void {
        }
        public function setMouseState(state:String):void{
        }
        private function toggleFilter():void {
            if (textField.filters.length == 0) {
                textField.filters = [glowFilter];
            } else {
                textField.filters = [];
            }
        }
    }
}
```

20 Wybierz polecenie Plik > Zapisz. Nadaj plikowi nazwę MyCellRenderer.as, umieść go w tym samym katalogu, co plik FLA, a następnie kliknij przycisk OK.

21 Wybierz polecenie Sterowanie > Testuj film.



## Właściwości klasy CellRenderer

Właściwość `data` to obiekt zawierający wszystkie właściwości mechanizmu renderującego komórki. Na przykład w poniższej klasie, która definiuje niestandardowy mechanizm renderujący komórki i rozszerza klasę `CheckBox`, funkcja ustawiająca właściwość `data` przekazuje wartość `data.label` do właściwości `label` odziedziczonej z klasy `CheckBox`:

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

Właściwość `selected` określa, czy komórka jest zaznaczona na liście.

## Stosowanie mechanizmu renderującego komórki do kolumny obiektu DataGrid

Obiekt `DataGrid` może zawierać wiele kolumn i możliwe jest określenie różnych mechanizmów renderujących komórki dla poszczególnych kolumn. Każda kolumna obiektu `DataGrid` ma odzwierciedlenie w postaci obiektu `DataGridColumn`, a klasa `DataGridColumn` zawiera właściwość `cellRenderer` umożliwiającą zdefiniowanie mechanizmu renderującego komórki danej kolumny.

## Definiowanie mechanizmu renderującego dla komórki edytowalnej

Klasa `DataGridCellEditor` definiuje mechanizm renderujący używany dla komórek edytowalnych w obiekcie `DataGrid`. Staje się ona mechanizmem renderującym komórki, gdy właściwość `editable` obiektu `DataGrid` jest ustawiona na `true`, a użytkownik kliknie komórkę przeznaczoną do edycji. Aby zdefiniować mechanizm renderujący dla komórki edytowalnej, należy ustawić właściwość `itemEditor` każdego elementu tablicy `columns` obiektu `DataGrid`.

## Użycie obrazu, pliku SWF lub klipu filmowego jako mechanizmu renderującego

Klasa `ImageCell`, będąca podklasą klasy `CellRenderer`, definiuje obiekt służący do renderowania komórek, których zasadniczą zawartością jest obraz, plik SWF lub klip filmowy. Klasa `ImageCell` zawiera następujące style służące do definiowania wyglądu komórki:

- `imagePadding` — szerokość wypełnienia oddzielającego krawędź komórki od krawędzi obrazu, w pikselach.
- `selectedSkin` — karnacja używana do sygnalizacji stanu zaznaczenia.

- `textOverlayAlpha` — nieprzezroczystość (krycie) nakładki za etykietą komórki.
- `textPadding` — szerokość wypełnienia oddzielającego krawędź komórki od krawędzi tekstu, w pikselach.

Klasa `ImageCell` jest domyślnym mechanizmem renderującym komórki dla klasy `TileList`.

## Ułatwienia dostępu do składników

Czytniki ekranu, które generują dźwiękowe opisy treści ekranu, ułatwiają użytkownikom niewidzącym i niedowidzącym dostęp do treści wizualnych w aplikacjach Flash. Informacje na temat przygotowania aplikacji Flash do współpracy z czytnikiem ekranu zawiera Rozdział 18, „Tworzenie treści dostępnych dla użytkowników niepełnosprawnych”, w podręczniku *Korzystanie z programu Flash*.

Aby składnik w języku ActionScript 3.0 był dostępny dla czytnika ekranu, konieczne jest także zaimportowanie odpowiedniej dla składnika klasy ułatwień dostępu oraz wywołanie metody `enableAccessibility()` tej klasy. Dla następujących składników w języku ActionScript 3.0 można włączyć ułatwienia dostępu zapewniające współpracę z czytnikiem ekranu:

| Składnik          | Klasa ułatwień dostępu          |
|-------------------|---------------------------------|
| Przycisk          | <code>ButtonAccImpl</code>      |
| Pole wyboru       | <code>CheckBoxAccImpl</code>    |
| Lista kombinowana | <code>ComboBoxAccImpl</code>    |
| Lista             | <code>ListAccImpl</code>        |
| Przycisk opcji    | <code>RadioButtonAccImpl</code> |
| Lista tytułów     | <code>TileListAccImpl</code>    |

Klasy ułatwień dostępu odpowiadające poszczególnym składnikom znajdują się w pakiecie `fl.accessibility`. Na przykład, aby udostępnić składnik `CheckBox` dla czytnika ekranu, należałoby dodać do aplikacji następujące instrukcje:

```
import fl.accessibility.CheckBoxAccImpl;

CheckBoxAccImpl.enableAccessibility();
```

Ułatwienia dostępu są włączane dla danej klasy składników tylko raz, niezależnie od liczby utworzonych wystąpień.

**Uwaga:** Włączenie ułatwień dostępu nieznacznie zwiększa objętość pliku, ponieważ powoduje dodanie do niego wymaganych klas podczas kompilacji.

Większość składników umożliwia także nawigację za pomocą klawiatury. Więcej informacji o włączaniu ułatwień dostępu do składników oraz nawigacji za pomocą klawiatury zawierają sekcje Interakcja z użytkownikiem w rozdziale „Korzystanie ze składników interfejsu użytkownika” na stronie 47 oraz opisy klas ułatwień dostępu w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

# Rozdział 4: Korzystanie ze składników interfejsu użytkownika

W tym rozdziale wyjaśniony został sposób, w jaki należy używać składników interfejsu użytkownika języka ActionScript 3.0 zawartych w programie Flash.

## Korzystanie ze składnika Button

Składnik Button jest prostokątnym przyciskiem, którego rozmiar można zmienić i który można nacisnąć myszą lub spacją w celu zainicjowania operacji aplikacji. Do składnika Button można dodać niestandardową ikonę. Można również zmienić zachowanie składnika Button z naciskania na przełączanie. Składnik Button typu przełącznik pozostaje wciśnięty po kliknięciu i powraca do stanu up po ponownym kliknięciu.

Składnik Button jest podstawową częścią wielu formularzy i aplikacji sieci Web. Przycisków można używać zawsze, gdy użytkownik ma inicjować zdarzenie. Na przykład, większość formularzy zawiera przycisk Prześlij. Do prezentacji można dodać przyciski Poprzedni i Następny.

## Posługiwanie się składnikiem Button

W aplikacji przycisk można włączyć lub wyłączyć. W stanie disabled przycisk nie odbiera danych wprowadzanych za pomocą klawiatury lub myszy. Włączony przycisk jest aktywowany po jego kliknięciu lub przejściu do niego za pomocą klawisza Tab. Po aktywowaniu instancji Button można nią sterować za pomocą następujących klawiszy:

| Klucz        | Opis                                                                                                                                     |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Shift+Tab    | Przesuwa aktywność do poprzedniego obiektu.                                                                                              |
| Spacja       | Powoduje naciśnięcie lub zwolnienie przycisku i wywołanie zdarzenia <code>click</code> .                                                 |
| Karta        | Przesuwa aktywność do kolejnego obiektu.                                                                                                 |
| Enter/Return | Przesuwa aktywność do kolejnego obiektu, jeśli przycisk został ustawiony jako domyślny składnik Button klasy <code>FocusManager</code> . |

Więcej informacji na temat sterowania aktywnością zawiera opis interfejsu `IFocusManager` oraz opis klasy `FocusManager` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#) oraz w sekcji „Praca z klasą `FocusManager`” na stronie 29.

Podgląd aktywny każdej z instancji Button odzwierciedla zmiany dokonane w parametrach w Inspektorze właściwości oraz Inspektorze składników podczas tworzenia.

**Uwaga:** Jeśli ikona jest większa niż przycisk, zostanie ona rozszerzona poza krawędzie przycisku.

Aby wskazać przycisk jako domyślny przycisk aplikacji (przycisk, który otrzymuje zdarzenie `click` po naciśnięciu klawisza Enter), należy ustawić właściwość `FocusManager.defaultButton`. Na przykład poniższy kod ustawia jako przycisk domyślny instancję składnika Button o nazwie `submitButton`.

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

Po dodaniu składnika Button do aplikacji można go udostępnić dla czytnika ekranu, dodając do kodu ActionScript następujące wiersze:

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

Ułatwienia dostępu włącza się dla danej klasy składników tylko raz, niezależnie od liczby utworzonych instancji.

## Parametry składnika Button

Dla każdej instancji klasy Button można w Inspektorze właściwości (Okno > Właściwości > Właściwości) lub w Inspektorze składników (Okno > Inspektor składników) ustawić następujące parametry tworzenia treści: `emphasized`, `label`, `labelPlacement`, `selected` i `toggle`. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Przypisanie wartości dla tych parametrów jest równoważne z ustawieniem wstępnego stanu właściwości w aplikacji. Ustawienie właściwości w ActionScript powoduje zastąpienie wartości ustawionej w parametrze. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy Button w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

## Tworzenie aplikacji ze składnikiem Button

Poniższa procedura wyjaśnia jak podczas tworzenia dodać składnik Button do aplikacji. W tym przykładzie kliknięcie przycisku powoduje zmianę stanu składnika ColorPicker.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik Button z panelu Składniki na stół montażowy i wprowadź dla niego w Inspektorze właściwości następujące wartości:
  - Wprowadź dla instancji nazwę **aButton**.
  - Dla parametru label wprowadź wartość **Show**.
- 3 Dodaj składnik ColorPicker na stół montażowy i nadaj instancji nazwę **aCp**.
- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
aCp.visible = false;

aButton.addEventListener(MouseEvent.CLICK, clickHandler);

function clickHandler(event:MouseEvent):void {

    switch(event.currentTarget.label) {
        case "Show":
            aCp.visible = true;
            aButton.label = "Disable";
            break;
        case "Disable":
            aCp.enabled = false;
            aButton.label = "Enable";
            break;
        case "Enable":
            aCp.enabled = true;
            aButton.label = "Hide";
            break;
        case "Hide":
            aCp.visible = false;
            aButton.label = "Show";
            break;
    }
}
```

Drugi wiersz kodu rejestruje funkcję `clickHandler()` jako funkcję obsługi zdarzenia `MouseEvent.CLICK`. Zdarzenie następuje po kliknięciu składnika `Button`, powodując, w zależności od wartości składnika `Button`, wykonanie przez funkcję `clickHandler()` jednej z następujących operacji:

- Show - powoduje wyświetlenie składnika `ColorPicker` i zmienia etykietę składnika `Button` na `Disable`.
- Disable - wyłącza składnik `ColorPicker` i zmienia etykietę składnika `Button` na `Enable`.
- Enable - włącza składnik `ColorPicker` i zmienia etykietę składnika `Button` na `Hide`.
- Hide - powoduje ukrycie składnika `ColorPicker` i zmienia etykietę składnika `Button` na `Show`.

5 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

## Tworzenie aplikacji ze składnikiem Button

Poniższa procedura tworzy składnik `Button` typu przełącznik z użyciem kodu `ActionScript` i wyświetla typ zdarzenia w panelu `Wyjście` i po naciśnięciu składnika `Button`. W przykładzie instancja `Button` tworzona jest przez wywołanie konstruktora klasy i dodanie go na stół montażowy przez wywołanie metody `addChild()`.

- 1 Utwórz nowy dokument `Flash (ActionScript 3.0)`.
- 2 Przeciągnij składnik `Button` z panelu `Składniki` do panelu `Biblioteka bieżącego dokumentu`.  
Powoduje to dodanie składnika do biblioteki, ale nie jest on wyświetlany w aplikacji.
- 3 Otwórz panel `Operacje`, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod w celu utworzenia instancji `Button`:

```
import fl.controls.Button;

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

Metoda `move()` powoduje ustawienie przycisku w lokalizacji 50 (współrzędna x), 50 (współrzędna y) na stole montażowym.

- 4 Dodaj następujący kod ActionScript, aby utworzyć detektor zdarzeń i funkcję obsługi zdarzenia:

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

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

- 5 Wybierz polecenie Sterowanie > Testuj film.

Po kliknięciu przycisku program Flash wyświetli w panelu Wyjście komunikat "Typ zdarzenia: kliknięcie".

## Korzystanie ze składnika CheckBox

Obiekt `CheckBox` jest kwadratowym polem, które można zaznaczyć lub z którego można usunąć zaznaczenie. Po jego zaznaczeniu w polu wyświetlone zostanie oznaczone znacznikiem wyboru. Do składnika `CheckBox` można dodać etykietę tekstową i umieścić ją po lewej/prawej stronie lub poniżej/powyżej składnika.

Składników `CheckBox` można używać w celu zgromadzenia zestawu wartości `true` i `false`, które się wzajemnie nie wykluczają. Na przykład aplikacja zbierająca informacje o rodzaju samochodu, który użytkownik chce nabyć może użyć składników `CheckBox`, aby umożliwić wybieranie jego cech.

### Posługiwanie się składnikiem CheckBox

Składnika `CheckBox` w aplikacji można włączyć lub wyłączyć. Jeśli składnik `CheckBox` jest włączony i zostanie kliknięty lub kliknięta zostanie jego etykieta, składnik zostanie aktywowany i wyświetlony zostanie jako naciśnięty. Jeśli wskaźnik z wciśniętym przyciskiem myszy przesunięty zostanie poza obszar powiązany ze składnikiem `CheckBox` lub jego etykietą wygląd składnika powróci do stanu pierwotnego i zachowa aktywność do wprowadzania. Stan składnika `CheckBox` nie zmienia się do momentu zwolnienia przycisku myszy nad składnikiem. Ponadto składnik `CheckBox` zawiera dwa stany `disabled` (`selected` i `deselected`), które używają odpowiednio karnacji `selectedDisabledSkin` i `disabledSkin`, które nie zezwalają na wykonywanie działań za pomocą myszy lub klawiatury.

Jeśli składnik `CheckBox` jest wyłączony, karnacja `disabled` jest wyświetlana niezależnie od działania użytkownika. W stanie `disabled` składnik `CheckBox` nie odbiera danych wprowadzanych za pomocą klawiatury lub myszy.

Instancja `CheckBox` jest aktywowana po jej kliknięciu lub przejściu do niej za pomocą klawisza `Tab`. Po aktywowaniu instancji `CheckBox` można nią sterować za pomocą następujących klawiszy:

| Klucz     | Opis                                                                                |
|-----------|-------------------------------------------------------------------------------------|
| Shift+Tab | Przesuwa aktywność do poprzedniego elementu.                                        |
| Spacja    | Zaznacza lub usuwa zaznaczenie składnika i wywołuje zdarzenie <code>change</code> . |
| Karta     | Przesuwa aktywność do kolejnego elementu.                                           |

Więcej informacji na temat sterowania aktywnością zawiera sekcja „Praca z klasą `FocusManager`” na stronie 29 oraz opis klasy `FocusManager` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

Podgląd aktywny każdej z instancji `CheckBox` odzwierciedla zmiany dokonane w parametrach w Inspektorze właściwości oraz Inspektorze składników podczas tworzenia.

Po dodaniu składnika `CheckBox` do aplikacji można go udostępnić dla czytnika ekranu, dodając do kodu ActionScript następujące wiersze:

```
import fl.accessibility.CheckBoxAccImpl;

CheckBoxAccImpl.enableAccessibility();
```

Ułatwienia dostępu włącza się dla danej klasy składników tylko raz, niezależnie od liczby posiadanych instancji.

## Parametry składnika CheckBox

Dla każdej instancji składnika `CheckBox` w Inspektorze właściwości lub Inspektorze składników można ustawić następujące parametry tworzenia treści: `label`, `labelPlacement` i `selected`. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy `CheckBox` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

## Tworzenie aplikacji za pomocą składnika CheckBox

Poniższa procedura wyjaśnia za pomocą fragmentu z formularza aplikacji do udzielania kredytów jak podczas tworzenia dodać składnik `CheckBox` do aplikacji. W formularzu zawarte jest pytanie, czy zgłaszający się jest właścicielem domu i zapewniany jest składnik `CheckBox` w celu umożliwienia odpowiedzi “tak”. Jeśli taka odpowiedź zostanie zaznaczona, w formularzu wyświetlone zostaną dwa przyciski opcji, które umożliwią wskazanie względnej wartości domu.

### Tworzenie aplikacji za pomocą składnika Checkbox

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik `Checkbox` z panelu Składniki na stół montażowy.
- 3 W Inspektorze właściwości wykonaj następujące kroki:
  - Dla nazwy instancji wprowadź `homeCh`.
  - Wprowadź `140` dla wartości szerokości (W).
  - Wprowadź “Own your home?” dla parametru `label`.
- 4 Przeciągnij dwa składniki `RadioButton` z panelu Składniki na stół montażowy i umieść je po prawej stronie poniżej składnika `CheckBox`. W Inspektorze właściwości wprowadź dla nich następujące wartości:
  - Dla nazw instancji wprowadź `underRb` i `overRb`.
  - W obu składnikach `RadioButton` wprowadź wartość `120` dla parametru `Sz` (szerokość).
  - Wprowadź `Under $500,000?` dla parametru `label` instancji `underRb`.

- Wprowadź wartość **Over \$500,000?** dla parametru label instancji overRb.
- Wprowadź **valueGrp** dla parametru groupName obu składników RadioButton.

5 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
homeCh.addEventListener(MouseEvent.CLICK, clickHandler);
underRb.enabled = false;
overRb.enabled = false;

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

Kod ten tworzy moduł obsługi zdarzenia CLICK, który włącza składniki RadioButton underRb i overRb, jeśli zaznaczony jest składnik CheckBox homeCh, a wyłącza je, jeśli homeCh nie jest zaznaczony. Więcej informacji zawiera opis klasy MouseEvent w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

6 Wybierz polecenie Sterowanie > Testuj film.

W poniższym przykładzie powielana jest poprzednia aplikacja i tworzone są za pomocą ActionScript składnik CheckBox i składniki RadioButton.

## Tworzenie składnika Checkbox przy użyciu kodu ActionScript

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przecignij składnik CheckBox i składnik RadioButton z panelu Składniki do panelu Biblioteka bieżącego dokumentu. Jeśli panel Biblioteka jest zamknięty, naciśnij klawisze Ctrl+L lub wybierz opcje Okno > Biblioteka, aby otworzyć panel Biblioteka.

Spowoduje to udostępnienie składników w aplikacji użytkownika, ale nie zostaną one wstawione na stół montażowy.

- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod w celu utworzenia i ustawienia instancji składnika:

```
import fl.controls.CheckBox;
import fl.controls.RadioButton;

var homeCh:CheckBox = new CheckBox();
var underRb:RadioButton = new RadioButton();
var overRb:RadioButton = new RadioButton();
addChild(homeCh);
addChild(underRb);
addChild(overRb);
underRb.groupName = "valueGrp";
overRb.groupName = "valueGrp";
homeCh.move(200, 100);
homeCh.width = 120;
homeCh.label = "Own your home?";
underRb.move(220, 130);
underRb.enabled = false;
underRb.width = 120;
underRb.label = "Under $500,000?";
overRb.move(220, 150);
overRb.enabled = false;
overRb.width = 120;
overRb.label = "Over $500,000?";
```



Kod ten używa konstruktorów `CheckBox()` i `RadioButton()` w celu utworzenia składników oraz metody `addChild()` w celu umieszczenia ich na stole montażowym. Metoda `move()` używana jest do ustawienia składników na stole montażowym.

- 4 Dodaj następujący kod ActionScript, aby utworzyć detektor zdarzeń i funkcję obsługi zdarzenia:

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

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

Kod ten tworzy moduł obsługi zdarzenia `CLICK`, który włącza przyciski opcji `underRb` i `overRb`, jeśli zaznaczony jest składnik `CheckBox` `homeCh`, a wyłącza je, jeśli `homeCh` nie jest zaznaczony. Więcej informacji zawiera opis klasy `MouseEvent` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

- 5 Wybierz polecenie Sterowanie > Testuj film.

## Korzystanie ze składnika ColorPicker

Składnik `ColorPicker` umożliwia wybieranie koloru z listy próbek. Domyślny tryb składnika `ColorPicker` wyświetla pojedynczy kolor w kwadratowym przycisku. Po kliknięciu przycisku w panelu próbek pojawia się lista dostępnych kolorów razem z polem tekstowym, w którym wyświetlana jest heksadecymalna wartość dla bieżącego zaznaczenia koloru.

Kolory składnika `ColorPicker` można ustawić przez ustawienie ich właściwości `colors` na wartości kolorów, które mają być wyświetlane.

### Posługiwanie się składnikiem ColorPicker

Składnik `ColorPicker` umożliwia wybranie koloru i zastosowanie go w innym obiekcie aplikacji. Na przykład, aby zezwolić na personalizację elementów aplikacji np. koloru tła lub tekstu, można dołączyć składnik `ColorPicker` i zastosować wybrany przez użytkownika kolor.

Użytkownik wybiera kolor przez kliknięcie jego próbki w panelu lub przez wprowadzenie do pola tekstowego jego wartości heksadecymalnej. Po wybraniu koloru właściwość `selectedColor` składnika `ColorPicker` można użyć w celu zastosowania koloru do tekstu lub innego obiektu w aplikacji.

Instancja `ColorPicker` jest aktywowana po przesunięciu nad nią wskaźnika lub przejściu do niej za pomocą klawisza `Tab`. Jeśli panel próbek składnika `ColorPicker` jest otwarty, można nim sterować za pomocą następujących klawiszy:

| Klucz            | Opis                                                        |
|------------------|-------------------------------------------------------------|
| Home             | Przesuwa zaznaczenie do pierwszego koloru w panelu próbek.  |
| Strzałka w górę  | Przesuwa zaznaczenie w górę o jeden wiersz w panelu próbek. |
| Strzałka w dół   | Przesuwa zaznaczenie w dół o jeden wiersz w panelu próbek.  |
| Strzałka w prawo | Przesuwa zaznaczenie w panelu próbek o jeden kolor w prawo. |
| Strzałka w lewo  | Przesuwa zaznaczenie w panelu próbek o jeden kolor w lewo.  |
| End              | Przesuwa zaznaczenie do ostatniego koloru w panelu próbek.  |

## Parametry składnika ColorPicker

W Inspektorze właściwości lub w Inspektorze składników można ustawić następujące parametry tworzenia dla każdej instancji składnika ColorPicker: `selectedColor` i `showTextField`. Każdy z parametrów posiada odpowiadającą mu właściwość ActionScript o tej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy ColorPicker w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

## Tworzenie aplikacji ze składnikiem ColorPicker

W poniższym przykładzie podczas tworzenia do aplikacji dodawany jest składnik ColorPicker. W tym przykładzie za każdym razem, gdy zmieniany jest kolor w składniku ColorPicker, w funkcji `changeHandler()` wywoływana jest funkcja `drawBox()` w celu narysowania nowego pola z kolorem wybranym w składniku ColorPicker.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ColorPicker z panelu Składniki na środek stołu montażowego i nadaj instancji nazwę **aCp**.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.events.ColorPickerEvent;

var aBox:MovieClip = new MovieClip();
drawBox(aBox, 0xFF0000); //draw a red box
addChild(aBox);

aCp.addEventListener(ColorPickerEvent.CHANGE, changeHandler);

function changeHandler(event:ColorPickerEvent):void {
    drawBox(aBox, event.target.selectedColor);
}

function drawBox(box:MovieClip, color:uint):void {
    box.graphics.beginFill(color, 1);
    box.graphics.drawRect(100, 150, 100, 100);
    box.graphics.endFill();
}
```

- 4 Wybierz polecenie Sterowanie > Testuj film.
- 5 Kliknij składnik ColorPicker i wybierz kolor, aby wypełnić pole kolorem.

## Tworzenie składnika ColorPicker przy użyciu kodu ActionScript

W tym przykładzie do utworzenia na stole montażowym składnika ColorPicker używany jest konstruktor `ColorPicker()` oraz metoda `addChild()`. Właściwość `colors` ustawiana jest na wartości kolorów: dla czerwonego (0xFF0000), zielonego (0x00FF00) i niebieskiego (0x0000FF) w celu określenia kolorów, które będą wyświetlane w składniku ColorPicker. Tworzony jest także składnik TextArea i za każdym razem, gdy w składniku ColorPicker wybrany zostanie inny kolor, w celu uzgodnienia w przykładzie zmieniony zostanie kolor tekstu składnika TextArea.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ColorPicker z panelu Składniki do panelu Biblioteka.
- 3 Przeciągnij składnik TextArea z panelu Składniki do panelu Biblioteka.
- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

**Korzystanie ze składników interfejsu użytkownika**

```
import fl.controls.ColorPicker;
import fl.controls.TextArea;
import fl.events.ColorPickerEvent;

var aCp:ColorPicker = new ColorPicker();
var aTa:TextArea = new TextArea();
var aTf:TextFormat = new TextFormat();

aCp.move(100, 100);
aCp.colors = [0xff0000, 0x00ff00, 0x0000ff];
aCp.addEventListener(ColorPickerEvent.CHANGE, changeHandler);

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

addChild(aCp);
addChild(aTa);

function changeHandler(event:ColorPickerEvent):void {
    if(TextFormat(aTa.getStyle("textFormat"))){
        aTf = TextFormat(aTa.getStyle("textFormat"));
    }
    aTf.color = event.target.selectedColor;
    aTa.setStyle("textFormat", aTf);
}
```

5 Wybierz polecenie Sterowanie > Testuj film.

## Korzystanie ze składnika ComboBox

Składnik ComboBox umożliwia wybranie pojedynczej pozycji z listy rozwijanej. ComboBox może być składnikiem statycznym lub edytowalnym. Edytowalny składnik ComboBox umożliwia bezpośrednie wprowadzanie tekstu do pola tekstowego na górze listy. Jeśli możliwe jest zetknięcie się listy rozwijanej z dolną częścią dokumentu, zostanie ona rozwinięta w górę, zamiast w dół. Składnik ComboBox jest zbudowany z trzech składników podrzędnych: BaseButton, TextInput i List.

Jeśli ComboBox jest składnikiem edytowalnym, obszarem klikania jest tylko przycisk — nie pole tekstowe. Dla statycznego składnika ComboBox obszar klikania stanowi przycisk i pole tekstowe. W odpowiedzi na naciśnięcie obszaru klikania składnika lista rozwijana jest otwierana lub zamykana.

Po dokonaniu wyboru na liście za pomocą myszy lub klawiatury etykieta zaznaczenia kopiowana jest do pola tekstowego u góry składnika ComboBox.

## Posługiwanie się składnikiem ComboBox

Składnika ComboBox można użyć w dowolnym formularzu lub aplikacji, która wymaga pojedynczego wyboru z listy opcji. Na przykład można udostępnić rozwijaną listę województw w formularzu adresu klienta. Edytowalnego składnika ComboBox można użyć dla bardziej złożonych scenariuszy. Na przykład w aplikacji, która podaje kierunki jazdy, można użyć edytowalnego składnika ComboBox, aby umożliwić użytkownikowi wprowadzenie adresu początkowego i docelowego. Lista rozwijana zawierać będzie poprzednio wprowadzone adresy.

Jeśli składnik ComboBox jest edytowalny, właściwość `editable` ma wartość `true`, poniższe klawisze usuwają aktywność z wejściowego pola tekstowego i pozostawiają w nim poprzednią wartość. Wyjątek stanowi klawisz `Enter`, który najpierw stosuje nową wartość, jeśli użytkownik wprowadził tekst.

| Klucz          | Opis                                                                                                                                                                                                                             |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Shift+Tab      | Przesuwa aktywność do poprzedniej pozycji. Jeśli zaznaczona zostanie nowa pozycja, wywołane zostanie zdarzenie <code>change</code> .                                                                                             |
| Karta          | Przesuwa aktywność do kolejnej pozycji. Jeśli zaznaczona zostanie nowa pozycja, wywołane zostanie zdarzenie <code>change</code> .                                                                                                |
| Strzałka w dół | Przesuwa zaznaczenie w dół o jedną pozycję.                                                                                                                                                                                      |
| End            | Przesuwa zaznaczenie w dół listy.                                                                                                                                                                                                |
| Esc            | Zamyka listę rozwijaną i zwraca aktywność do składnika ComboBox.                                                                                                                                                                 |
| Enter          | Zamyka listę rozwijaną i zwraca aktywność do składnika ComboBox. Jeśli składnik ComboBox jest edytowalny, a użytkownik wprowadzi tekst, naciśnięcie klawisza <code>Enter</code> ustawi w składniku wartość wprowadzonego tekstu. |
| Home           | Przesuwa zaznaczenie na górę listy.                                                                                                                                                                                              |
| Page Up        | Przesuwa zaznaczenie w górę o jedną stronę.                                                                                                                                                                                      |
| Page Down      | Przesuwa zaznaczenie w dół o jedną stronę.                                                                                                                                                                                       |

Po dodaniu składnika ComboBox do aplikacji można go udostępnić dla czytnika ekranu, dodając do kodu ActionScript następujące wiersze:

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

Ułatwienia dostępu włącza się dla danej klasy składników tylko raz, niezależnie od liczby posiadanych instancji.

## Parametry składnika ComboBox

W Inspektorze właściwości lub w Inspektorze składników można ustawić następujące parametry tworzenia dla każdej instancji składnika: `dataProvider`, `editable`, `prompt` oraz `rowCount`. Każdy z parametrów posiada odpowiadającą mu właściwość ActionScript o tej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy ComboBox w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#). Informacje o używaniu parametru `dataProvider` zawiera sekcja „Korzystanie z parametru `dataProvider`” na stronie 31.

## Tworzenie aplikacji ze składnikiem ComboBox

Poniższa procedura opisuje jak podczas tworzenia dodać składnik ComboBox do aplikacji. Składnik ComboBox jest edytowalny, dlatego jeśli do pola tekstowego wpisana zostanie wartość **Add**, do listy rozwijanej w przykładzie dodana zostanie pozycja.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ComboBox na stół montażowy i nadaj instancji nazwę **aCb**. Na karcie Parametry ustaw parametr `editable` na wartość `true`.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod:

```
import fl.data.DataProvider;
import fl.events.ComponentEvent;

var items:Array = [
    {label:"screen1", data:"screenData1"},
    {label:"screen2", data:"screenData2"},
    {label:"screen3", data:"screenData3"},
    {label:"screen4", data:"screenData4"},
    {label:"screen5", data:"screenData5"},
];
aCb.dataProvider = new DataProvider(items);

aCb.addEventListener(ComponentEvent.ENTER, onAddItem);

function onAddItem(event:ComponentEvent):void {
    var newRow:int = 0;
    if (event.target.text == "Add") {
        newRow = event.target.length + 1;
        event.target.addItemAt({label:"screen" + newRow, data:"screenData" + newRow},
            event.target.length);
    }
}
```

- 4 Wybierz polecenie Sterowanie > Testuj film.

## Tworzenie składnika ComboBox przy użyciu kodu ActionScript

W poniższym przykładzie z użyciem kodu ActionScript tworzony jest składnik ComboBox i wypełniany jest listą uniwersytetów z obszaru stanu San Francisco i Kalifornia. Właściwość `width` składnika ComboBox dopasowywana jest do szerokości tekstu monitu, a właściwość `dropdownWidth` jest nieznacznie poszerzana w celu dopasowania do najdłuższej nazwy uniwersytetu.

Lista uniwersytetów w przykładzie tworzona jest w instancji Array z użyciem właściwości `label` w celu zapisania nazw szkół i właściwości `data` do zapisania adresów URL stron sieci Web dla każdej z uczelni. Instancja Array przypisywana jest do składnika ComboBox przez ustawienie właściwości `dataProvider`.

Po wybraniu z listy uniwersytetu wywoływane jest zdarzenie `Event.Zdarzenie CHANGE` oraz funkcja `changeHandler()`, która ładuje do żądania URL właściwość `data` w celu uzyskania dostępu do strony sieci Web.

Należy zauważyć, że ostatni wiersz instancji ComboBox ustawia właściwość `selectedIndex` na wartość `-1` w celu ponownego wyświetlenia monitu po zamknięciu listy. W innym wypadku monit zostanie zastąpiony nazwą wybranej uczelni.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ComboBox z panelu Składniki do panelu Biblioteka.

**3** Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.ComboBox;
import fl.data.DataProvider;
import flash.net.navigateToURL;

var sfUniversities:Array = new Array(
    {label:"University of California, Berkeley",
      data:"http://www.berkeley.edu/"},
    {label:"University of San Francisco",
      data:"http://www.usfca.edu/"},
    {label:"San Francisco State University",
      data:"http://www.sfsu.edu/"},
    {label:"California State University, East Bay",
      data:"http://www.csu Hayward.edu/"},
    {label:"Stanford University", data:"http://www.stanford.edu/"},
    {label:"University of Santa Clara", data:"http://www.scu.edu/"},
    {label:"San Jose State University", data:"http://www.sjsu.edu/" }
);

var aCb:ComboBox = new ComboBox();
aCb.dropdownWidth = 210;
aCb.width = 200;
aCb.move(150, 50);
aCb.prompt = "San Francisco Area Universities";
aCb.dataProvider = new DataProvider(sfUniversities);
aCb.addEventListener(Event.CHANGE, changeHandler);

addChild(aCb);

function changeHandler(event:Event):void {
    var request:URLRequest = new URLRequest();
    request.url = ComboBox(event.target).selectedItem.data;
    navigateToURL(request);
    aCb.selectedIndex = -1;
}
```

**4** Wybierz polecenie Sterowanie > Testuj film.

Przykład można zaimplementować i uruchomić w środowisku tworzenia programu Flash, jednak przy próbie dostępu do stron sieci Web uczelni, przez kliknięcie pozycji w składniku ComboBox, wyświetlone zostaną ostrzeżenia. Aby uzyskać dostęp do składnika ComboBox w pełni funkcjonalnego w Internecie, należy odwiedzić następującą stronę:

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

## Korzystanie ze składnika DataGrid

Składnik DataGrid umożliwia wyświetlanie danych w siatce wierszy i kolumn oraz pobieranie danych z tablicy lub zewnętrznego pliku XML, który można przetworzyć na tablicę dla obiektu DataProvider. Składnik DataGrid obsługuje poziome i pionowe przewijanie, zdarzenia (w tym edytowalne komórki) i posiada możliwości sortowania.

Cechy takie jak czcionka, kolor oraz krawędzie kolumn w siatce można dostosować i zmienić ich rozmiar. Jako mechanizmu renderującego komórki dla dowolnej kolumny w siatce można użyć niestandardowego klipu filmowego. (Mechanizm renderujący komórki wyświetla treść komórki). W celu utworzenia ekranu ze stylem widoku strony można wyłączyć paski przewijania i użyć metod DataGrid. Więcej informacji na temat dostosowywania zawiera opis klasy DataGridColumn w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

## Więcej tematów Pomocy

[Tworzenie, zapełnianie i zmiana rozmiaru składnika DataGrid](#)

[Dostosowywanie i sortowanie składnika DataGrid](#)

[Filtrowanie i formatowanie danych w składniku DataGrid](#)

## Posługiwanie się składnikiem DataGrid

Do obsługi składnika DataGrid można użyć myszy lub klawiatury.

Jeśli właściwość `sortableColumns` i właściwość `sortable` kolumny mają wartość `true`, kliknięcie nagłówka kolumny powoduje posortowanie danych w oparciu o wartości kolumny. Sortowanie dla pojedynczej kolumny można wyłączyć, ustawiając właściwość `sortable` na wartość `false`.

Jeśli właściwość `resizableColumns` ma wartość `true`, rozmiar kolumn można zmienić, przeciągając suwak podziału kolumn w wierszu nagłówka.

Kliknięcie edytowalnej komórki powoduje jej aktywowanie; kliknięcie nieedytowalnej komórki nie wpływa na zmianę aktywności na ekranie. Pojedyncza komórka jest edytowalna, jeśli obie właściwości `DataGrid.editable` i `DataGridColumn.editable` komórki mają wartość `true`.

Więcej informacji zawierają opisy klas `DataGrid` i `DataGridColumn` w [Skorowidzu języka ActionScript 3.0](#).

Po aktywowaniu instancji `DataGrid` (poprzez jej kliknięcie lub przejście do niej klawiszem `Tab`) można nią sterować za pomocą następujących klawiszy:

| Klucz                       | Opis                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Strzałka w dół              | Podczas edycji komórki punkt wstawiania przesuwany jest na koniec tekstu komórki. Jeśli komórka nie jest edytowalna, klawisz strzałki w dół obsługuje wybieranie tak, jak składnik <code>List</code> .                                                                                                                                                          |
| Strzałka w górę             | Podczas edycji komórki punkt wstawiania przesuwany jest na początek tekstu komórki. Jeśli komórka nie jest edytowalna, klawisz strzałki w górę obsługuje wybieranie tak, jak składnik <code>List</code> .                                                                                                                                                       |
| Shift+strzałka w górę/w dół | Jeśli składnik <code>DataGrid</code> nie jest edytowalny, a właściwość <code>allowMultipleSelection</code> ma wartość <code>true</code> , wybierane są sąsiednie wiersze. Zaznaczenie w przeciwnym kierunku dla drugiej ze strzałek powoduje odznaczenie wierszy, aż do osiągnięcia pierwszego wiersza; w tym punkcie wiersze w danym kierunku będą zaznaczone. |
| Shift+kliknięcie            | Jeśli parametr <code>allowMultipleSelection</code> ma wartość <code>true</code> , wybrane zostaną wszystkie między wybranym wierszem a bieżącą pozycją karetki (podświetloną komórką).                                                                                                                                                                          |
| Ctrl+kliknięcie             | Jeśli parametr <code>allowMultipleSelection</code> ma wartość <code>true</code> , wybrane zostaną dodatkowe wiersze, które nie muszą być wierszami sąsiednimi.                                                                                                                                                                                                  |
| Strzałka w prawo            | Podczas edycji komórki punkt wstawiania przesuwany jest o jeden znak w prawo. Jeśli komórka nie jest edytowalna, klawisz strzałki w prawo nie robi nic.                                                                                                                                                                                                         |
| Strzałka w lewo             | Podczas edycji komórki punkt wstawiania przesuwany jest o jeden znak w lewo. Jeśli komórka nie jest edytowalna, klawisz strzałki w lewo nie robi nic.                                                                                                                                                                                                           |
| Home                        | Zaznacza pierwszy wiersz w składniku <code>DataGrid</code> .                                                                                                                                                                                                                                                                                                    |

| Klucz                    | Opis                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| End                      | Zaznacza ostatni wiersz w składniku DataGrid.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| PageUp                   | Zaznacza pierwszy wiersz na stronie w składniku DataGrid. Strona składa się z wierszy, które składnik DataGrid może wyświetlić bez przewijania.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| PageDown                 | Zaznacza ostatni wiersz na stronie w składniku DataGrid. Strona składa się z wierszy, które składnik DataGrid może wyświetlić bez przewijania.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Return/Enter/Shift+Enter | Jeśli komórka jest edytowalna, dokonywana jest zmiana, a punkt wstawiania jest przesuwany do komórki znajdującej się w następnym wierszu w tej samej kolumnie (do góry lub w dół w zależności od stanu klawisza Shift).                                                                                                                                                                                                                                                                                                                                                                              |
| Shift+Tab/Tab            | Jeśli składnik jest edytowalny aktywność przesuwana jest do poprzedniej/kolejnej pozycji aż do osiągnięcia końca kolumny, a następnie do poprzedniego/kolejnego wiersza aż do osiągnięcia pierwszej lub ostatniej komórki. Jeśli zaznaczona została pierwsza komórka, klawisze Shift+Tab przesuną aktywność do poprzedzającego elementu sterującego. Jeśli zaznaczona została ostatnia komórka, klawisz Tab przesuwa aktywność do kolejnego elementu sterującego.<br><br>Jeśli składnik DataGrid nie jest edytowalny, aktywność zostanie przesunięta do poprzedniego/kolejnego elementu sterującego. |

Składnika DataGrid można używać jako podstawy dla wielu typów aplikacji sterowania danymi. Istnieje możliwość łatwego wyświetlenia sformatowanego tabelarycznego widoku danych, ale możliwe jest również użycie mechanizmów renderujących komórki w celu utworzenia bardziej wyrafinowanych i dostępnych dla edycji elementów interfejsu użytkownika. Składnik DataGrid w praktyce używany jest w:

- Klientach poczty uruchamianych w przeglądarkach sieci Web
- Stronach z wynikami wyszukiwania
- Aplikacjach z arkuszami (np. kalkulatory kredytowe, aplikacje z formularzami podatkowymi)

Podczas projektowania aplikacji ze składnikiem DataGrid pomocne jest zrozumienie konstrukcji składnika List, ponieważ klasa DataGrid rozszerza klasę SelectableList. Więcej informacji na temat klasy SelectableList i składnika List zawiera [Skorowidz języka ActionScript 3.0](#).

Po dodaniu składnika DataGrid do aplikacji można go udostępnić dla czytnika ekranu, dodając do kodu ActionScript następujące wiersze:

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

Ułatwienia dostępu włącza się dla danej klasy składników tylko raz, niezależnie od liczby instancji składnika. Więcej informacji zawiera rozdział 18, "Tworzenie treści dostępnych dla użytkowników niepełnosprawnych", w podręczniku *Korzystanie z programu Flash*.

## Parametry składnika DataGrid

W Inspektorze właściwości lub w Inspektorze składników można ustawić następujące parametry tworzenia dla każdej instancji składnika DataGrid: `allowMultipleSelection`, `editable`, `headerHeight`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `resizableColumns`, `rowHeight`, `showHeaders`, `verticalLineScrollSize`, `verticalPageScrollSize` oraz `verticalScrollPolicy`. Każdy z parametrów posiada odpowiadającą mu właściwość ActionScript o tej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy ComboBox w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).



## Tworzenie aplikacji ze składnikiem DataGrid

Aby utworzyć aplikację ze składnikiem DataGrid, należy najpierw określić skąd pochodzą dane. Dane zazwyczaj pochodzą z obiektu Array, które można umieścić w siatce za pomocą właściwości `dataProvider`. Aby dodać dane do siatki można także użyć metod klas DataGrid i DataGridColumn.

### Użycie lokalnego dostawcy danych ze składnikiem DataGrid:

W tym przykładzie składnik DataGrid tworzony jest w celu wyświetlenia wykazu zawodników drużyny softball. Wykaz zdefiniowany został w obiekcie Array (`aRoster`) i przypisany do właściwości `dataProvider` składnika DataGrid.

- 1 W programie Flash wybierz opcje Plik > Nowy, a następnie wybierz plik Flash (ActionScript 3.0).
- 2 Przeciągnij składnik DataGrid z panelu Składniki na stół montażowy.
- 3 W Inspektorze właściwości wprowadź dla instancji nazwę **aDg**.
- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.data.DataProvider;

bldRosterGrid(aDg);
var aRoster:Array = new Array();
aRoster = [
    {Name:"Wilma Carter", Bats:"R", Throws:"R", Year:"So", Home: "Redlands, CA"},
    {Name:"Sue Pennypacker", Bats:"L", Throws:"R", Year:"Fr", Home: "Athens, GA"},
    {Name:"Jill Smithfield", Bats:"R", Throws:"L", Year:"Sr", Home: "Spokane, WA"},
    {Name:"Shirley Goth", Bats:"R", Throws:"R", Year:"Sr", Home: "Carson, NV"},
    {Name:"Jennifer Dunbar", Bats:"R", Throws:"R", Year:"Fr", Home: "Seaside, CA"},
    {Name:"Patty Crawford", Bats:"L", Throws:"L", Year:"Jr", Home: "Whittier, CA"},
    {Name:"Angelina Davis", Bats:"R", Throws:"R", Year:"So", Home: "Odessa, TX"},
    {Name:"Maria Santiago", Bats:"L", Throws:"L", Year:"Sr", Home: "Tacoma, WA"},
    {Name:"Debbie Ferguson", Bats:"R", Throws:"R", Year: "Jr", Home: "Bend, OR"},
    {Name:"Karen Bronson", Bats:"R", Throws:"R", Year: "Sr", Home: "Billings, MO"},
    {Name:"Sylvia Munson", Bats:"R", Throws:"R", Year: "Jr", Home: "Pasadena, CA"},
    {Name:"Carla Gomez", Bats:"R", Throws:"L", Year: "Sr", Home: "Corona, CA"},
    {Name:"Betty Kay", Bats:"R", Throws:"R", Year: "Fr", Home: "Palo Alto, CA"},
];
aDg.dataProvider = new DataProvider(aRoster);
aDg.rowCount = aDg.length;

function bldRosterGrid(dg:DataGrid){
    dg.setSize(400, 300);
    dg.columns = ["Name", "Bats", "Throws", "Year", "Home"];
    dg.columns[0].width = 120;
    dg.columns[1].width = 50;
    dg.columns[2].width = 50;
    dg.columns[3].width = 40;
    dg.columns[4].width = 120;
    dg.move(50,50);
};
```

Funkcja `bldRosterGrid()` ustawia wielkość składnika DataGrid i ustawia kolejność kolumn oraz ich wielkości.

- 5 Wybierz polecenie Sterowanie > Testuj film.

### Wybierz kolumny i dodaj sortowanie dla składnika DataGrid w aplikacji

Nagłówek każdej kolumny można kliknąć w celu posortowania treści składnika DataGrid w porządku malejącym wg wartości w wybranej kolumnie.

W poniższym przykładzie wykorzystano metodę `addColumn()` do dodawania instancji klasy `DataGridColumn` do składnika `DataGrid`. Kolumny reprezentują nazwy odtwarzaczy oraz wyniki. W tym przykładzie przedstawiono również wybór właściwości `sortOptions` w celu określenia opcji sortowania dla każdej kolumny: `Array.CASEINSENSITIVE` dla kolumny `Name` i `Array.NUMERIC` dla kolumny `Score`. W przykładzie wielkość składnika `DataGrid` jest odpowiednia, ponieważ jego długość została ustawiona jako liczba wierszy, a szerokość na 200.

- 1 W programie Flash wybierz opcje Plik > Nowy, a następnie wybierz plik Flash (ActionScript 3.0).
- 2 Przeciągnij składnik `DataGrid` z panelu Składniki na stół montażowy.
- 3 W Inspektorze właściwości wprowadź dla instancji nazwę `aDg`.
- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.dataGridClasses.DataGridColumn;
import fl.events.DataGridEvent;
import fl.data.DataProvider;
// Create columns to enable sorting of data.
var nameDGC:DataGridColumn = new DataGridColumn("name");
nameDGC.sortOptions = Array.CASEINSENSITIVE;
var scoreDGC:DataGridColumn = new DataGridColumn("score");
scoreDGC.sortOptions = Array.NUMERIC;
aDg.addColumn(nameDGC);
aDg.addColumn(scoreDGC);
var aDP_array:Array = new Array({name:"clark", score:3135}, {name:"Bruce", score:403},
{name:"Peter", score:25});
aDg.dataProvider = new DataProvider(aDP_array);
aDg.rowCount = aDg.length;
aDg.width = 200;
```

- 5 Wybierz polecenie Sterowanie > Testuj film.

### Tworzenie instancji składnika DataGrid przy użyciu kodu ActionScript

W tym przykładzie przedstawiono tworzenie składnika `DataGrid` w języku ActionScript oraz wypełnianie go tablicą nazw odtwarzaczy i wartości `score`.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik `DataGrid` z panelu Składniki do panelu Biblioteka bieżącego dokumentu.  
Powoduje to dodanie składnika do biblioteki, ale nie jest on wyświetlany w aplikacji.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.DataGrid;
import fl.data.DataProvider;

var aDg:DataGrid = new DataGrid();
addChild(aDg);
aDg.columns = [ "Name", "Score" ];
aDg.setSize(140, 100);
aDg.move(10, 40);
```

Ten kod tworzy instancję `DataGrid`, a następnie określa wielkość i położenie siatki.

- 4 Utwórz tablicę, dodaj do niej dane i określ tablicę jako dostawcę danych dla składnika `DataGrid`:

**Korzystanie ze składników interfejsu użytkownika**

```
var aDP_array:Array = new Array();
aDP_array.push({Name:"Clark", Score:3135});
aDP_array.push({Name:"Bruce", Score:403});
aDP_array.push({Name:"Peter", Score:25});
aDg.dataProvider = new DataProvider(aDP_array);
aDg.rowCount = aDg.length;
```

- 5 Wybierz polecenie Sterowanie > Testuj film.

**Ładowanie składnika DataGrid za pomocą pliku XML**

W poniższym przykładzie przedstawiono korzystanie z klasy DataGridColumn w celu utworzenia kolumn DataGrid.

W przykładzie składnik DataGrid jest wypełniany poprzez wprowadzanie obiektu XML jako parametru value konstruktora DataProvider().

- 1 Korzystając z edytora tekstu, utwórz plik XML zawierający następujące dane i zapisz go pod nazwą team.xml w tym samym folderze, w którym zapisany zostanie plik FLA.

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

- 2 Utwórz nowy dokument Flash (ActionScript 3.0).
- 3 Na panelu Składniki kliknij dwukrotnie składnik DataGrid, aby dodać go do stołu montażowego.
- 4 W Inspektorze właściwości wprowadź dla instancji nazwę **aDg**.
- 5 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.dataGridClasses.DataGridColumn;
import fl.data.DataProvider;
import flash.net.*;
import flash.events.*;

var request:URLRequest = new URLRequest("team.xml");
var loader:URLLoader = new URLLoader;

loader.load(request);
loader.addEventListener(Event.COMPLETE, loaderCompleteHandler);

function loaderCompleteHandler(event:Event):void {

    var teamXML:XML = new XML(loader.data);

    var nameCol:DataGridColumn = new DataGridColumn("name");
    nameCol.headerText = "Name";
    nameCol.width = 120;
    var avgCol:DataGridColumn = new DataGridColumn("avg");
    avgCol.headerText = "Average";
    avgCol.width = 60;

    var myDP:DataProvider = new DataProvider(teamXML);

    aDg.columns = [nameCol, avgCol];
    aDg.width = 200;
    aDg.dataProvider = myDP;
    aDg.rowCount = aDg.length;
}
```

6 Wybierz polecenie Sterowanie > Testuj film.

## Korzystanie ze składnika Label

Składnik Label wyświetla pojedynczy wiersz tekstu — zwykle w celu identyfikowania innych elementów lub operacji na stronie sieci Web. Składnik Label (etykieta) może zostać sformatowany za pomocą znaczników formatujących HTML. Istnieje również możliwość kontrolowania wielkości i układu etykiety. Składniki Label nie mają krawędzi, nie mogą być aktywowane i nie generują żadnych zdarzeń.

Podgląd aktywny każdej instancji Label odzwierciedla zmiany dokonane w parametrach w Inspektorze właściwości oraz Inspektorze składników podczas tworzenia. Etykieta nie ma krawędzi, dlatego jedynym sposobem wyświetlenia jej podglądu aktywnego jest ustawienie jej parametrów tekstu.

### Posługiwanie się składnikiem Label

Składnik Label służy do tworzenia etykiety tekstowej dla innego składnika, np. etykieta „Nazwa:” po lewej stronie pola TextInput, do którego można wprowadzić nazwę użytkownika. Składnik Label może być używany zamiast zwykłego pola tekstowego, ponieważ możliwe jest używanie stylów w celu zapewnienia spójnego działania i wyglądu.

Jeśli wymagane jest obrócenie składnika Label, wówczas należy osadzić czcionki; w przeciwnym wypadku nie będą widoczne podczas testowania filmu.

## Parametry składnika Label

Dla każdej instancji składnika Label można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry tworzenia treści: `autoSize`, `condenseWhite`, `selectable`, `text` i `wordWrap`. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy Label w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

## Tworzenie aplikacji ze składnikiem Label

Poniższa procedura wyjaśnia jak podczas tworzenia dodać składnik Label do aplikacji. Przykład przedstawia tworzenie etykiety z tekstem „Data ważności”.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik Label z panelu Składniki na stół montażowy i określ dla niego następujące wartości w Inspektorze właściwości:
  - Dla nazwy instancji wprowadź **aLabel**.
  - Wprowadź **80** dla wartości Sz.
  - Wprowadź **100** dla wartości X.
  - Wprowadź **100** dla wartości Y.
  - Wprowadź **Data ważności** dla parametru `text`.
- 3 Przeciągnij składnik TextArea z panelu Składniki na stół montażowy i określ dla niego następujące wartości w Inspektorze właściwości:
  - Dla nazwy instancji wprowadź **aTa**.
  - Wprowadź **22** dla wartości W.
  - Wprowadź **200** dla wartości X.
  - Wprowadź **100** dla wartości Y.
- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
var today:Date = new Date();
var expDate:Date = addDays(today, 14);
aTa.text = expDate.toString();

function addDays(date:Date, days:Number):Date {
    return addHours(date, days*24);
}

function addHours(date:Date, hrs:Number):Date {
    return addMinutes(date, hrs*60);
}

function addMinutes(date:Date, mins:Number):Date {
    return addSeconds(date, mins*60);
}

function addSeconds(date:Date, secs:Number):Date {
    var mSecs:Number = secs * 1000;
    var sum:Number = mSecs + date.getTime();
    return new Date(sum);
}
```

- 5 Wybierz polecenie Sterowanie > Testuj film.

### Tworzenie instancji składnika Label przy użyciu kodu ActionScript

W poniższym przykładzie przedstawiono tworzenie parametru Label za pomocą języka ActionScript. Składnik Label służy do identyfikacji funkcji składnika ColorPicker i korzysta z właściwości `htmlText` w celu zastosowania formatowania do tekstu składnika Label.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik Label z panelu Składniki do panelu Biblioteka bieżącego dokumentu.
- 3 Przeciągnij składnik ColorPicker z panelu Składniki do panelu Biblioteka bieżącego dokumentu.
- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

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

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

addChild(aLabel);
addChild(aCp);

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

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

- 5 Wybierz polecenie Sterowanie > Testuj film.

## Korzystanie ze składnika List

Składnik List to przewijalne pole listy, na którym można wybrać jedną lub większą ilość opcji. Lista (składnik List) może również zawierać elementy graficzne, również inne składniki. W celu dodawania elementów widocznych na liście należy użyć okna dialogowego Wartości, jakie pojawia się po kliknięciu etykiet lub pól parametrów danych. W celu dodawania elementów do listy można również korzystać z metod `List.addItem()` i `List.addItemAt()`.

Składnik List korzysta z indeksu, w którym element z indeksem 0 jest najwyższym elementem wyświetlanym. Podczas dodawania, usuwania lub zastępowania elementów listy za pomocą metod i właściwości klasy List konieczne może być określenie indeksu dla elementu listy.

### Posługiwanie się składnikiem List

Istnieje możliwość skonfigurowania listy, na której użytkownicy będą mogli wybierać jedną wartość lub większą liczbę wartości. Na przykład: użytkownik odwiedzający stronę WWW sklepu elektronicznego musi wybrać elementy do kupienia. Lista zawiera 30 elementów; użytkownik przewija listę elementów i wybiera element, klikając go.

**Korzystanie ze składników interfejsu użytkownika**

Możliwe jest również zaprojektowanie listy (składnik List), która będzie korzystała z niestandardowych klipów jako wierszy, dzięki czemu użytkownik będzie miał dostęp do większej liczby informacji. Na przykład: w aplikacji e-mail każda skrzynka pocztowa może być składnikiem List, a każdy wiersz może zawierać ikony, które będą określały priorytet i status.

Składnik List staje się aktywny, gdy zostanie kliknięty lub wybrany za pomocą klawisza Tab — do kontrolowania tego składnika można używać następujących klawiszy:

Klucz	Opis
Klawisze alfanumeryczne	Przeskok do kolejnego elementu, którego pierwszy znak etykiety to <code>Key.getAscii()</code> .
Control	Klawisz przełączania, który umożliwia nieciągłe zaznaczanie i usuwanie zaznaczenia wielu opcji, nawet jeśli nie następują po sobie.
Strzałka w dół	Przesuwa zaznaczenie w dół o jeden element.
Home	Przesuwa zaznaczenie na górę listy.
Page Down	Przesuwa zaznaczenie w dół o jedną stronę.
Page Up	Przesuwa zaznaczenie w górę o jedną stronę.
Shift	Umożliwia zaznaczanie ciągle.
Strzałka w górę	Przesuwa zaznaczenie w górę o jeden element.

**Uwaga:** Wielkości przewijania są określone w pikselach, a nie w wierszach.

**Uwaga:** Wielkość strony stosowana po naciśnięciu klawisza Page Up i Page Down jest o jeden mniejsza niż liczba elementów, które mieszczą się na obszarze wyświetlania. Przykład: przewijanie stron w liście rozwijanej, z której wyświetlanych jest dziesięć wierszy, powoduje wyświetlanie elementów 0-9, 9-18, 18-27, itd. — jeden element jest wyświetlany na obydwu sąsiednich stronach.

Więcej informacji na temat sterowania aktywnością zawiera opis interfejsu IFocusManager oraz opis klasy FocusManager w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#) oraz w sekcji „Praca z klasą FocusManager” na stronie 29.

Podgląd aktywny każdej instancji List na stole montażowym odzwierciedla zmiany dokonane w parametrach w Inspektorze właściwości oraz Inspektorze składników podczas tworzenia.

Po dodaniu składnika List do aplikacji można go udostępnić dla czytnika ekranu, dodając do kodu ActionScript następujące wiersze:

```
import fl.accessibility.ListAccImpl;
```

```
ListAccImpl.enableAccessibility();
```

Ułatwienia dostępu włącza się dla danego składnika tylko raz, niezależnie od liczby instancji składnika. Więcej informacji zawiera rozdział 18, “Tworzenie treści dostępnych dla użytkowników niepełnosprawnych”, w podręczniku *Korzystanie z programu Flash*.

## Parametry składnika List

Dla każdej instancji składnika List można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry tworzenia treści: `allowMultipleSelection`, `dataProvider`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `multipleSelection`, `verticalLineScrollSize`, `verticalPageScrollSize` i `verticalScrollPolicy`. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy List w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*. Informacje o używaniu parametru `dataProvider` zawiera sekcja „Korzystanie z parametru `dataProvider`” na stronie 31.

## Tworzenie aplikacji ze składnikiem List

Poniższe przykłady przedstawiają sposób dodania składnika List do aplikacji podczas tworzenia.

### Dodawanie prostego składnika List do aplikacji

W tym przykładzie składnik List zawiera etykiety, które identyfikują modele samochodów oraz pola danych, które zawierają ceny.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik List z panelu Składniki na stół montażowy.
- 3 W Inspektorze właściwości wykonaj następujące kroki:
  - Wprowadź dla instancji nazwę `aList`.
  - Do `Sz` (szerokość) przypisz wartość `200`.
- 4 Użyj narzędzia Tekst, aby utworzyć pole tekstowe poniżej `aList` i nadać mu nazwę `aTf`.
- 5 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.List;
import flash.text.TextField;

aTf.type = TextFieldType.DYNAMIC;
aTf.border = false;

// Create these items in the Property inspector when data and label
// parameters are available.
aList.addItem({label:"1956 Chevy (Cherry Red)", data:35000});
aList.addItem({label:"1966 Mustang (Classic)", data:27000});
aList.addItem({label:"1976 Volvo (Xcllnt Cond)", data:17000});
aList.allowMultipleSelection = true;

aList.addEventListener(Event.CHANGE, showData);

function showData(event:Event) {
    aTf.text = "This car is priced at: $" + event.target.selectedItem.data;
}
```

W tym kodzie metoda `addItem()` służy do wypełnienia składnika `aList` trzema elementami — do każdego z tych elementów przypisywana jest pojedyncza wartość `label`, która jest widoczna na liście, oraz wartość `data`. Po wybraniu elementu w składniku List detektor zdarzeń wywołuje funkcję `showData()`, która wyświetla wartość `data` dla wybranego elementu.

- 6 Wybierz polecenia Sterowanie > Testuj film, aby skompilować i uruchomić aplikację.



### Wypełnianie instancji List za pomocą dostawcy danych

W tym przykładzie przedstawiono tworzenie składnika List modeli samochodów i ich cen. W przykładzie składnik List jest wypełniany przez dostawcę danych, a nie przez metodę `addItem()`.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik List z panelu Składniki na stół montażowy.
- 3 W Inspektorze właściwości wykonaj następujące kroki:
  - Wprowadź dla instancji nazwę **aList**.
  - Do Sz (szerokość) przypisz wartość **200**.
- 4 Użyj narzędzia Tekst, aby utworzyć pole tekstowe poniżej aList i nadać mu nazwę **aTf**.
- 5 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.List;
import fl.data.DataProvider;
import flash.text.TextField;

aTf.type = TextFieldType.DYNAMIC;
aTf.border = false;

var cars:Array = [
    {label:"1956 Chevy (Cherry Red)", data:35000},
    {label:"1966 Mustang (Classic)", data:27000},
    {label:"1976 Volvo (Xcllnt Cond)", data:17000},
];
aList.dataProvider = new DataProvider(cars);
aList.allowMultipleSelection = true;

aList.addEventListener(Event.CHANGE, showData);

function showData(event:Event) {
    aTf.text = "This car is priced at: $" + event.target.selectedItem.data;
}
```

- 6 Wybierz polecenia Sterowanie > Testuj film, wyświetlić składnik List z jego elementami.

### Korzystanie ze składnika List w celu kontrolowania instancji MovieClip

W poniższym przykładzie przedstawiono tworzenie składnika List nazw kolorów — po wybraniu nazwy kolor zostaje zastosowany do obiektu MovieClip.

- 1 Utwórz dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik List z panelu Składniki na stół montażowy i określ dla niego następujące wartości w Inspektorze właściwości:
  - Dla nazwy instancji wprowadź **aList**.
  - Wprowadź **60** dla wartości W.
  - Wprowadź **100** dla wartości X.
  - Wprowadź **150** dla wartości Y.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

**Korzystanie ze składników interfejsu użytkownika**

```
aList.addItem({label:"Blue", data:0x0000CC});
aList.addItem({label:"Green", data:0x00CC00});
aList.addItem({label:"Yellow", data:0xFFFF00});
aList.addItem({label:"Orange", data:0xFF6600});
aList.addItem({label:"Black", data:0x000000});

var aBox:MovieClip = new MovieClip();
addChild(aBox);

aList.addEventListener(Event.CHANGE, changeHandler);
function changeHandler(event:Event) {
    drawBox(aBox, event.target.selectedItem.data);
};

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1.0);
    box.graphics.drawRect(225, 150, 100, 100);
    box.graphics.endFill();
}
```

- 4 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.
- 5 Klikaj kolory w składniku List, aby wyświetlić je w obiekcie MovieClip.

**Tworzenie instancji składnika List przy użyciu kodu ActionScript:**

W tym przykładzie przy użyciu kodu ActionScript tworzona jest prosta lista, która jest następnie zapełniana przy użyciu metody addItem().

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik List z panelu Składniki do panelu Biblioteka.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.List;

var aList:List = new List();
aList.addItem({label:"One", data:1});
aList.addItem({label:"Two", data:2});
aList.addItem({label:"Three", data:3});
aList.addItem({label:"Four", data:4});
aList.addItem({label:"Five", data:5});
aList.setSize(60, 40);
aList.move(200,200);
addChild(aList);
aList.addEventListener(Event.CHANGE, changeHandler);
function changeHandler(event:Event):void {
    trace(event.target.selectedItem.data);
}
```

- 4 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

## Korzystanie ze składnika NumericStepper

Składnik NumericStepper umożliwia użytkownikowi przechodzenie przez uporządkowany zestaw liczb. Składnik zawiera liczbę w polu tekstowym, która jest widoczna obok małych przycisków strzałek w górę i w dół. Po naciśnięciu przycisku liczba jest krokowo zwiększana lub zmniejszana — w zależności od jednostki określonej w parametrze `stepSize` — dopóki użytkownik nie zwolni przycisku lub do osiągnięcia wartości maksymalnej albo minimalnej. Tekst znajdujący się w polu tekstowym składnika NumericStepper również jest dostępny do edycji.

Podgląd aktywny każdej instancji NumericStepper odzwierciedla ustawienie parametru wartości w Inspektorze właściwości oraz Inspektorze składników. W podglądzie aktywnym nie istnieje jednak żadna zależność między użyciem myszy lub klawisza na klawiaturze z przyciskami strzałek składnika NumericStepper.

### Posługiwanie się składnikiem NumericStepper

Składnik NumericStepper może być używany wszędzie tam, gdzie użytkownik ma wybrać wartość liczbową. Na przykład: składnik NumericStepper może być używany w formie umożliwiającej wybór miesiąca, dnia i roku daty ważności karty kredytowej. Składnik NumericStepper może również służyć do zwiększania lub zmniejszania wielkości czcionek.

Składnik NumericStepper obsługuje wyłącznie dane liczbowe. Podczas tworzenia wymagane jest powiększenie składnika Stepper, gdy liczba widocznych miejsc liczbowych ma być większa niż dwa (na przykład: liczby 5246 lub 1,34).

W aplikacji można włączyć lub wyłączyć składnik NumericStepper. W stanie disabled składnik NumericStepper nie odbiera danych wprowadzanych za pomocą klawiatury lub myszy. W stanie enabled składnik NumericStepper staje się aktywny po jego kliknięciu lub wybraniu za pomocą klawisza Tab, a w środku składnika aktywne staje się pole tekstowe. Po aktywowaniu instancji NumericStepper można nią sterować za pomocą następujących klawiszy:

Klucz	Opis
Strzałka w dół	Zmiana wartości o jedną jednostkę.
Strzałka w lewo	Przesunięcie punktu wprowadzania w lewo w polu tekstowym.
Strzałka w prawo	Przesunięcie punktu wprowadzania w prawo w polu tekstowym.
Shift+Tab	Przesuwa aktywność do poprzedniego obiektu.
Karta	Przesuwa aktywność do kolejnego obiektu.
Strzałka w górę	Zmiana wartości o jedną jednostkę.

Więcej informacji na temat sterowania aktywnością zawiera opis klasy FocusManager w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#) oraz w sekcji „Praca z klasą FocusManager” na stronie 29.

### Parametry składnika NumericStepper

Dla każdej instancji składnika NumericStepper można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry tworzenia treści: `maximum`, `minimum`, `stepSize` i `value`. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy NumericStepper w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

## Tworzenie aplikacji ze składnikiem NumericStepper

Poniższa procedura wyjaśnia jak podczas tworzenia dodać składnik NumericStepper do aplikacji. Przykład przedstawia umieszczenie składnika NumericStepper i składnika Label na stole montażowym oraz utworzenie detektora dla zdarzenia `Event.CHANGE` w instancji NumericStepper. Gdy wartość składnika NumericStepper zostanie zmieniona, w przykładzie pojawi się nowa wartość we właściwości `text` instancji Label.

- 1 Przeciągnij składnik NumericStepper z panelu Składniki na stół montażowy.
- 2 W Inspektorze właściwości wprowadź dla instancji nazwę `aNs`.
- 3 Przeciągnij składnik Label z panelu Składniki na stół montażowy.
- 4 W Inspektorze właściwości wprowadź dla instancji nazwę `aLabel`.
- 5 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import flash.events.Event;

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

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

W tym przykładzie właściwość `text` etykiety (Label) otrzymuje wartość składnika NumericStepper. Funkcja `changeHandler()` aktualizuje właściwość `text` etykiety przy każdej zmianie wartości instancji NumericStepper.

- 6 Wybierz polecenie Sterowanie > Testuj film.

### Tworzenie składnika NumericStepper przy użyciu języka ActionScript:

W tym przykładzie przedstawiono tworzenie trzech składników NumericStepper za pomocą kodu ActionScript — poszczególne składniki służą do wprowadzania miesiąca, dnia i roku daty urodzenia użytkownika. W przykładzie dodawane są również składniki Label dla monitu oraz dla identyfikatorów poszczególnych składników NumericStepper.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik Label do panelu Biblioteka.
- 3 Przeciągnij składnik NumericStepper do panelu Biblioteka.
- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.Label;
import fl.controls.NumericStepper;

var dobPrompt:Label = new Label();
var moPrompt:Label = new Label();
var dayPrompt:Label = new Label();
var yrPrompt:Label = new Label();

var moNs:NumericStepper = new NumericStepper();
var dayNs:NumericStepper = new NumericStepper();
var yrNs:NumericStepper = new NumericStepper();

addChild(dobPrompt);
addChild(moPrompt);
addChild(dayPrompt);
addChild(yrPrompt);
addChild(moNs);
addChild(dayNs);
addChild(yrNs);

dobPrompt.setSize(65, 22);
dobPrompt.text = "Date of birth:";
dobPrompt.move(80, 150);

moNs.move(150, 150);
moNs.setSize(40, 22);
moNs.minimum = 1;
moNs.maximum = 12;
moNs.stepSize = 1;
moNs.value = 1;

moPrompt.setSize(25, 22);
moPrompt.text = "Mo.";
moPrompt.move(195, 150);

dayNs.move(225, 150);
dayNs.setSize(40, 22);
dayNs.minimum = 1;
dayNs.maximum = 31;
dayNs.stepSize = 1;
dayNs.value = 1;

dayPrompt.setSize(25, 22);
dayPrompt.text = "Day";
dayPrompt.move(270, 150);

yrNs.move(300, 150);
yrNs.setSize(55, 22);
yrNs.minimum = 1900;
yrNs.maximum = 2006;
yrNs.stepSize = 1;
yrNs.value = 1980;

yrPrompt.setSize(30, 22);
yrPrompt.text = "Year";
yrPrompt.move(360, 150);
```

5 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

## Korzystanie ze składnika ProgressBar

Składnik ProgressBar wyświetla postęp ładowania treści, co stanowi istotną informację dla użytkownika, gdy treść ma znaczną objętość, a jej ładowanie może opóźnić wykonanie aplikacji. Składnik ProgressBar jest użyteczny podczas wyświetlania postępu ładowania obrazów i elementów aplikacji. Proces ładowania może być określony lub nieokreślony. Pasek postępu *determinate* stanowi liniową reprezentację postępu zadania w czasie i jest używany, gdy ilość treści do załadowania jest znana. Pasek postępu *indeterminate* jest używany wówczas, gdy ilość treści do załadowania nie jest znana. Istnieje również możliwość dodania składnika Label w celu wyświetlania postępu w postaci wartości procentowej.

Składnik ProgressBar korzysta z 9-plastrowej skali i zawiera karnację paska, karnację śledzenia i karnację *indeterminate*.

### Posługiwanie się składnikiem ProgressBar

Istnieją trzy tryby korzystania ze składnika ProgressBar. Najczęściej używane są tryby: zdarzenie i próbkowanie. Te tryby określają proces ładowania, który generuje zdarzenia `progress` i `complete` (tryby zdarzeń i próbkowania) lub udostępnia właściwości `bytesLoaded` i `bytesTotal` (tryb próbkowania). Składnik ProgressBar może być również używany w trybie ręcznym — w tym celu należy ustawić właściwości `maximum`, `minimum` oraz `value` wraz z wywołaniami metody `ProgressBar.setProgress()`. Właściwość *indeterminate* można ustawić w taki sposób, aby określała, czy składnik ProgressBar zawiera wypełnienie w paski oraz źródło nieznanej wielkości (`true`) lub wypełnienie ciągle i źródło znanej wielkości (`false`).

W celu określenia trybu składnika ProgressBar należy ustawić jego właściwość `mode` — za pośrednictwem parametru `mode` w Inspektorze właściwości lub Inspektorze składników albo za pomocą języka ActionScript.

Gdy składnik ProgressBar służy do prezentacji statusu przetwarzania, np. analizowania 100000 elementów — wówczas, jeśli składnik znajduje się w pojedynczej pętli klatki, aktualizacje składnika ProgressBar nie będą widoczne, ponieważ zawartość ekranu nie będzie odświeżana.

### Parametry składnika ProgressBar

Dla każdej instancji składnika ProgressBar można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry: `direction`, `mode` i `source`. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie.

Istnieje możliwość napisania kodu ActionScript w celu kontrolowania tych i innych opcji dla składnika ProgressBar z wykorzystaniem jego właściwości, metod i zdarzeń. Więcej informacji zawiera opis klasy ProgressBar w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

## Tworzenie aplikacji ze składnikiem ProgressBar

Poniższa procedura wyjaśnia jak podczas tworzenia dodać składnik ProgressBar do aplikacji. W przykładzie składnik ProgressBar korzysta z trybu zdarzenia. W trybie zdarzeń ładowana treść generuje zdarzenia `progress` i `complete`, które rozsyła składnik ProgressBar w celu określenia postępu. Gdy wystąpi zdarzenie `progress`, w przykładzie dochodzi do aktualizacji etykiety w celu przedstawienia procentu załadowanej treści. Gdy wystąpi zdarzenie `complete`, w przykładzie zostaje wyświetlony komunikat "Loading complete" oraz wartość właściwości `bytesTotal`, która jest wielkością pliku.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ProgressBar z panelu Składniki na stół montażowy.
  - W Inspektorze właściwości wprowadź dla instancji nazwę **aPb**.
  - W sekcji Parametry wprowadź **200** dla wartości X.
  - Wprowadź **260** dla wartości Y.
  - Wybierz `event` dla parametru `mode`.
- 3 Przeciągnij składnik Button z panelu Składniki na stół montażowy.
  - W Inspektorze właściwości wprowadź dla instancji nazwę **loadButton**.
  - Wprowadź **220** dla parametru X.
  - Wprowadź **290** dla parametru Y.
  - Dla parametru `label` wprowadź **Load Sound**.
- 4 Przeciągnij składnik Label na stół montażowy i nadaj instancji nazwę **progLabel**.
  - Wprowadź **150** dla wartości Sz.
  - Wprowadź **200** dla parametru X.
  - Wprowadź **230** dla parametru Y.
  - W sekcji Parametry usuń wartość parametru `text`.
- 5 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript ładujący pliki dźwiękowe mp3:

```
import fl.controls.ProgressBar;
import flash.events.ProgressEvent;
import flash.events.IOErrorEvent;

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

aPb.addEventListener(ProgressEvent.PROGRESS, progressHandler);
aPb.addEventListener(Event.COMPLETE, completeHandler);
aSound.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
loadButton.addEventListener(MouseEvent.CLICK, clickHandler);

function progressHandler(event:ProgressEvent):void {
    progLabel.text = ("Sound loading ... " + aPb.percentComplete);
}

function completeHandler(event:Event):void {
    trace("Loading complete");
    trace("Size of file: " + aSound.bytesTotal);
    aSound.close();
    loadButton.enabled = false;
}

function clickHandler(event:MouseEvent) {
    aSound.load(request);
}

function ioErrorHandler(event:IOErrorEvent):void {
    trace("Load failed due to: " + event.text);
}
```

## 6 Wybierz polecenie Sterowanie > Testuj film.

### Tworzenie aplikacji ze składnikiem ProgressBar w trybie próbkowania

Poniższy przykład przedstawia ustawianie trybu próbkowania dla składnika ProgressBar. W trybie próbkowania postęp jest określony na podstawie wykrywania zdarzeń `progress` dla treści oraz za pomocą właściwości `bytesLoaded` i `bytesTotal` w celu obliczenia postępu. Ten przykład ładuje obiekt `Sound`, wykrywa zdarzenia `progress` obiektu i oblicza załadowany procent, wykorzystując w tym celu właściwości `bytesLoaded` i `bytesTotal`. Załadowany procent jest wyświetlany na etykiecie oraz na panelu Wyjście.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ProgressBar z panelu Składniki na stół montażowy i wprowadź dla niego w Inspektorze właściwości następujące wartości:
  - Dla nazwy instancji wprowadź **aPb**.
  - Dla wartości X wprowadź **185**.
  - Dla wartości Y wprowadź **225**.
- 3 Przeciągnij składnik Label na stół montażowy i określ dla niego następujące wartości w Inspektorze właściwości:
  - Dla nazwy instancji wprowadź **progLabel**.
  - Dla wartości X wprowadź **180**.



- Dla wartości Y wprowadź **180**.
- W sekcji Parametry usuń wartość parametru text.

- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript, który utworzy obiekt Sound (aSound) i wywoła metodę loadSound() w celu załadowania dźwięku do obiektu Sound:

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

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

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

aSound.load(request);

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

- 5 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

### Tworzenie aplikacji ze składnikiem ProgressBar w trybie ręcznym

Poniższy przykład przedstawia ustawianie trybu ręcznego dla składnika ProgressBar. W trybie ręcznym postęp należy ustawiać ręcznie, wywołując metodę setProgress() i wprowadzając do niej bieżące i maksymalne wartości w celu określenia zakresu postępu. W trybie ręcznym nie jest ustawiana właściwość source. W przykładzie wykorzystano składnik NumericStepper o wartości maksymalnej równej 250, który służy do zwiększania składnika ProgressBar. Gdy wartość w składniku NumericStepper ulegnie zmianie i wywoła zdarzenie CHANGE, wówczas moduł obsługi zdarzeń (nsChangeHandler) wywoła metodę setProgress() w celu modyfikacji składnika ProgressBar. Nastąpi również wyświetlenie procentu postępu na podstawie wartości maksymalnej.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ProgressBar z panelu Składniki na stół montażowy i określ dla niego następujące wartości w Inspektorze właściwości:
  - Dla nazwy instancji wprowadź **aPb**.
  - Wprowadź **180** dla wartości X.
  - Wprowadź **175** dla wartości Y.
- 3 Przeciągnij składnik NumericStepper na stół montażowy i określ dla niego następujące wartości w Inspektorze właściwości:
  - Dla nazwy instancji wprowadź **aNs**.
  - Wprowadź **220** dla wartości X.
  - Wprowadź **215** dla wartości Y.
  - W sekcji Parametry wprowadź **250** dla parametru maximum, **0** dla wartości minimum, **1** dla parametru stepSize oraz **0** dla parametru value.

4 Przeciągnij składnik Label na stół montażowy i określ dla niego następujące wartości w Inspektorze właściwości:

- Dla nazwy instancji wprowadź **progLabel**.
- Wprowadź **150** dla wartości Sz.
- Wprowadź **180** dla wartości X.
- Wprowadź **120** dla wartości Y.
- Na karcie Parametry usuń wartość Label parametru text.

5 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod:

```
import fl.controls.ProgressBarDirection;
import fl.controls.ProgressBarMode;
import flash.events.Event;

aPb.direction = ProgressBarDirection.RIGHT;
aPb.mode = ProgressBarMode.MANUAL;
aPb.minimum = aNs.minimum;
aPb.maximum = aNs.maximum;
aPb.indeterminate = false;

aNs.addEventListener(Event.CHANGE, nsChangeHandler);

function nsChangeHandler(event:Event):void {
    aPb.value = aNs.value;
    aPb.setProgress(aPb.value, aPb.maximum);
    progLabel.text = "Percent of progress = " + int(aPb.percentComplete) + "%";
}
```

6 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

7 Kliknij przycisk strzałki w górę składnika NumericStepper, aby zwiększyć wartość składnika ProgressBar.

### Tworzenie składnika ProgressBar przy użyciu kodu ActionScript

W poniższym przykładzie przedstawiono tworzenie składnika ProgressBar za pomocą języka ActionScript. Niniejszy przykład — oprócz tego, że tworzy składnik — stanowi kopię poprzedniego przykładu, który tworzy składnik ProgressBar w trybie ręcznym.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ProgressBar do panelu Biblioteka.
- 3 Przeciągnij składnik NumericStepper do panelu Biblioteka.
- 4 Przeciągnij składnik Label do panelu Biblioteka.
- 5 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod:

```
import fl.controls.ProgressBar;
import fl.controls.NumericStepper;
import fl.controls.Label;
import fl.controls.ProgressBarDirection;
import fl.controls.ProgressBarMode;
import flash.events.Event;

var aPb:ProgressBar = new ProgressBar();
var aNs:NumericStepper = new NumericStepper();
var progLabel:Label = new Label();

addChild(aPb);
addChild(aNs);
addChild(progLabel);

aPb.move(180,175);
aPb.direction = ProgressBarDirection.RIGHT;
aPb.mode = ProgressBarMode.MANUAL;

progLabel.setSize(150, 22);
progLabel.move(180, 150);
progLabel.text = "";

aNs.move(220, 215);
aNs.maximum = 250;
aNs.minimum = 0;
aNs.stepSize = 1;
aNs.value = 0;

aNs.addEventListener(Event.CHANGE, nsChangeHandler);

function nsChangeHandler(event:Event):void {
    aPb.setProgress(aNs.value, aNs.maximum);
    progLabel.text = "Percent of progress = " + int(aPb.percentComplete) + "%";
}
```

- Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.
- Kliknij przycisk strzałki w górę składnika NumericStepper, aby zwiększyć wartość składnika ProgressBar.

## Korzystanie ze składnika RadioButton

Składnik RadioButton pozwala wymusić na użytkowniku wybranie jednej opcji z określonego zbioru możliwości. Składnik ten musi być używany w grupie (co najmniej jako jedna z dwu instancji RadioButton). W danej chwili może być wybrany nie więcej niż jeden członek grupy. Wybór jednego przycisku opcji w grupie powoduje usunięcie zaznaczenia innego, obecnie zaznaczonego przycisku opcji w obrębie danej grupy. Parametr `groupName` służy do wskazywania grupy, do której należy dany przycisk opcji.

Przycisk opcji (składnik RadioButton) stanowi podstawową część wielu aplikacji formularzy w sieci WWW. Przyciski opcji mogą być używane zawsze, gdy użytkownik ma wybrać jedną opcję z grupy opcji. Przykład: przyciski opcji mogą być używane w formularzu w celu zapytania o kartę kredytową, której chce użyć użytkownik.

## Posługiwanie się składnikiem RadioButton

Przycisk opcji może być włączony lub wyłączony. W stanie disabled przycisk opcji nie odbiera danych wprowadzanych za pomocą klawiatury lub myszy. Po kliknięciu przez użytkownika lub przejściu klawiszem Tab do dowolnego przycisku z grupy składników RadioButton uaktywniany jest tylko wybrany przycisk opcji. Następnie użytkownik może użyć następujących klawiszy do sterowania:

Klucz	Opis
Strzałka w górę/Strzałka w lewo	Zaznaczenie przesuwa się do poprzedniego przycisku opcji w danej grupie przycisków opcji.
Strzałka w dół/Strzałka w prawo	Zaznaczenie przesuwa się do kolejnego przycisku opcji w danej grupie przycisków opcji.
Karta	Przesuwa obszar aktywny z grupy przycisków opcji do następnego składnika.

Więcej informacji na temat sterowania aktywnością zawiera opis interfejsu IFocusManager oraz opis klasy FocusManager w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#) oraz w sekcji „Praca z klasą FocusManager” na stronie 29.

Podgląd aktywny każdej instancji RadioButton na stole montażowym odzwierciedla zmiany dokonane w parametrach w Inspektorze właściwości oraz Inspektorze składników podczas tworzenia. Jednak wzajemne wykluczenie zaznaczenia nie jest widoczne w podglądzie aktywnym. Jeśli dla wybranego parametru zostanie ustawiona wartość true dla dwóch przycisków opcji w tej samej grupie, obydwa przyciski zostaną zaznaczone, nawet jeśli tylko instancja utworzona jako ostatnia jest zaznaczona w czasie wykonywania. Więcej informacji zawiera sekcja „Parametry składnika RadioButton” na stronie 80.

Po dodaniu składnika RadioButton do aplikacji można go udostępnić dla czytnika ekranu, dodając do kodu ActionScript następujące wiersze:

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

Ułatwienia dostępu włącza się dla danej klasy składników tylko raz, niezależnie od liczby posiadanych instancji. Więcej informacji zawiera rozdział 18, „Tworzenie treści dostępnych dla użytkowników niepełnosprawnych”, w podręczniku Korzystanie z programu Flash.

## Parametry składnika RadioButton

Dla każdej instancji składnika RadioButton można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry tworzenia treści: groupName, label, LabelPlacement, selected i value. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy RadioButton w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

W kodzie ActionScript możliwe jest ustawienie dodatkowych opcji dla instancji RadioButton za pomocą metod, właściwości i zdarzeń klasy RadioButton.

## Tworzenie aplikacji ze składnikiem RadioButton

Poniższa procedura wyjaśnia jak podczas tworzenia dodać składniki RadioButton do aplikacji. W tym przykładzie składniki RadioButton reprezentują pytanie, na które udzielana jest odpowiedź tak lub nie. Dane ze składnika RadioButton są wyświetlane w obszarze TextArea.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij dwa składniki RadioButton z panelu Składniki na stół montażowy.

- 3 Zaznacz pierwszy przycisk opcji. W Inspektorze właściwości nadaj instancji przycisku nazwę **yesRb**, a grupie nazwę **rbGroup**.
- 4 Zaznacz drugi przycisk opcji. W Inspektorze właściwości nadaj instancji przycisku nazwę **noRb**, a grupie nazwę **rbGroup**.
- 5 Przeciągnij składnik TextArea z panelu Składniki na stół montażowy i nadaj nazwę instancji **aTa**.
- 6 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
yesRb.label = "Yes";
yesRb.value = "For";
noRb.label = "No";
noRb.value = "Against";

yesRb.move(50, 100);
noRb.move(100, 100);
aTa.move(50, 30);
noRb.addEventListener(MouseEvent.CLICK, clickHandler);
yesRb.addEventListener(MouseEvent.CLICK, clickHandler);

function clickHandler(event:MouseEvent):void {
    aTa.text = event.target.value;
}
```

- 7 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

### Tworzenie składnika RadioButton przy użyciu kodu ActionScript

W tym przykładzie wykorzystano język ActionScript w celu utworzenia trzech składników RadioButton dla kolorów: czerwonego, niebieskiego i zielonego oraz w celu narysowania szarego pola. Właściwość `value` każdego składnika RadioButton określa szesnastkową wartość dla koloru powiązanego z przyciskiem. Gdy użytkownik kliknie jeden z przycisków RadioButton, funkcja `clickHandler()` wywoła funkcję `drawBox()`, przekazując kolor z właściwości `value` składnika RadioButton w celu nałożenia koloru pola.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik RadioButton do panelu Biblioteka.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.RadioButton;
import fl.controls.RadioButtonGroup;

var redRb:RadioButton = new RadioButton();
var blueRb:RadioButton = new RadioButton();
var greenRb:RadioButton = new RadioButton();
var rbGrp:RadioButtonGroup = new RadioButtonGroup("colorGrp");

var aBox:MovieClip = new MovieClip();
drawBox(aBox, 0xCCCCCC);

addChild(redRb);
addChild(blueRb);
addChild(greenRb);
addChild(aBox);

redRb.label = "Red";
redRb.value = 0xFF0000;
blueRb.label = "Blue";
blueRb.value = 0x0000FF;
greenRb.label = "Green";
greenRb.value = 0x00FF00;
redRb.group = blueRb.group = greenRb.group = rbGrp;
redRb.move(100, 260);
blueRb.move(150, 260);
greenRb.move(200, 260);

rbGrp.addEventListener(MouseEvent.CLICK, clickHandler);

function clickHandler(event:MouseEvent):void {
    drawBox(aBox, event.target.selection.value);
}

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1.0);
    box.graphics.drawRect(125, 150, 100, 100);
    box.graphics.endFill();
}
```

#### 4 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

Więcej informacji zawiera opis klasy `RadioButton` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

## Korzystanie ze składnika ScrollPane

Składnik `ScrollPane` umożliwia wyświetlanie treści zbyt dużej, by zmieściła się w obszarze, do którego jest ładowana. Przykład: jeśli istnieje duży obraz i dostępna jest tylko mała ilość miejsca w aplikacji, można załadować obraz do składnika `ScrollPane`. Składnik `ScrollPane` akceptuje klipy filmowe oraz pliki w formatach JPEG, PNG, GIF i SWF.

**Korzystanie ze składników interfejsu użytkownika**

Składniki, takie jak `ScrollPane` i `UILoader` zawierają zdarzenia `complete`, dzięki którym możliwe jest określenie, kiedy treść została załadowana. Jeśli wymagane jest ustawienie właściwości treści składnika `ScrollPane` lub `UILoader`, należy wykryć zdarzenie `complete` i ustawić właściwość w module obsługi zdarzeń. Przykład: poniższy kod tworzy detektor zdarzeń dla zdarzenia `Event.Zdarzenie COMPLETE` oraz moduł obsługi zdarzeń, który ustawia dla właściwości `alpha` treści składnika `ScrollPane` wartość `.5`:

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

Jeśli położenie zostanie określone podczas ładowania treści do składnika `ScrollPane`, należy określić je (współrzędne X i Y) jako 0, 0. Przykład: poniższy kod ładuje składnik `ScrollPane` poprawnie, ponieważ pole zostaje narysowane w położeniu 0, 0:

```
var box:MovieClip = new MovieClip();
box.graphics.beginFill(0xFF0000, 1);
box.graphics.drawRect(0, 0, 150, 300);
box.graphics.endFill();
aSp.source = box;//load ScrollPane
```

Więcej informacji zawiera opis klasy `ScrollPane` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

**Posługiwanie się składnikiem ScrollPane**

Składnik `ScrollPane` może być włączony lub wyłączony. W stanie `disabled` składnik `ScrollPane` nie odbiera danych wprowadzanych za pomocą klawiatury lub myszy. Użytkownik może korzystać z następujących klawiszy w celu sterowania aktywnym składnikiem `ScrollPane`:

Klucz	Opis
Strzałka w dół	Treść przesuwa się w górę o jedną linię przewijania w pionie.
Strzałka w górę	Treść przesuwa się w dół o jedną linię przewijania w pionie.
End	Treść przesuwa się do dołu składnika <code>ScrollPane</code> .
Strzałka w lewo	Treść przesuwa się w prawo o jedną linię przewijania w poziomie.
Strzałka w prawo	Treść przesuwa się w lewo o jedną linię przewijania w poziomie.
Home	Treść przesuwa się do góry składnika <code>ScrollPane</code> .
End	Treść przesuwa się do dołu składnika <code>ScrollPane</code> .
PageDown	Treść przesuwa się w górę o jedną stronę przewijania w pionie.
PageUp	Treść przesuwa się w dół o jedną stronę przewijania w pionie.

Użytkownik może użyć myszy w celu posługiwania się składnikiem `ScrollPane` w zakresie jego treści oraz w obszarze pasków przewijania w poziomie i w pionie. Użytkownik może przeciągać treść za pomocą myszy, gdy dla właściwości `scrollDrag` ustawiona jest wartość `true`. Jeśli na obszarze treści widoczny jest wskaźnik w postaci dłoni, oznacza to, że użytkownik może przeciągać treść. W tym przypadku — inaczej niż w większości elementów sterowania — operacja rozpoczyna się po naciśnięciu przycisku myszy i jest kontynuowana do jego zwolnienia. Jeśli treść zawiera poprawne punkty tabulacji, dla `scrollDrag` należy ustawić wartość `false`. W przeciwnym wypadku każde naciśnięcie przycisku myszy w obrębie treści będzie powodowało przeciąganie z przewijaniem.

## Parametry składnika ScrollPane

Dla każdej instancji składnika ScrollPane można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry: `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `scrollDrag`, `source`, `verticalLineScrollSize`, `verticalPageScrollSize` i `verticalScrollPolicy`. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy ScrollPane w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

Istnieje możliwość napisania kodu ActionScript w celu kontrolowania tych i innych opcji dla składnika ScrollPane z wykorzystaniem jego właściwości, metod i zdarzeń.

## Tworzenie aplikacji ze składnikiem ScrollPane

Poniższa procedura wyjaśnia jak podczas tworzenia dodać składnik ScrollPane do aplikacji. W tym przykładzie składnik ScrollPane ładuje obraz ze ścieżki określonej przez właściwość `source`.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ScrollPane z panelu Składniki na stół montażowy i nadaj nazwę instancji **aSp**.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.events.ScrollEvent;

aSp.setSize(300, 200);

function scrollListener(event:ScrollEvent):void {
    trace("horizontalScPosition: " + aSp.horizontalScrollPosition +
        ", verticalScrollPosition = " + aSp.verticalScrollPosition);
};
aSp.addEventListener(ScrollEvent.SCROLL, scrollListener);

function completeListener(event:Event):void {
    trace(event.target.source + " has completed loading.");
};
// Add listener.
aSp.addEventListener(Event.COMPLETE, completeListener);

aSp.source = "http://www.helpexamples.com/flash/images/image1.jpg";
```

- 4 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

## Tworzenie instancji składnika ScrollPane przy użyciu kodu ActionScript

Przykład prezentuje tworzenie instancji ScrollPane, określanie jej rozmiaru oraz ładowanie do niej obrazu za pomocą właściwości `source`. W przykładzie tworzone są również dwa detektory. Pierwszy z nich wykrywa zdarzenie `scroll` i wyświetla położenie obrazu, gdy użytkownik przewija w poziomie i w pionie. Drugi wykrywa zdarzenie `complete`, a po wykryciu wyświetla na panelu Wyjście komunikat z informacją o zakończeniu ładowania obrazu.

W tym przykładzie przedstawiono tworzenie instancji ScrollPane za pomocą kodu ActionScript oraz umieszczanie w instancji obiektu MovieClip (czerwone pole) o szerokości 150 pikseli i wysokości 300 pikseli.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ScrollPane z panelu Składniki do panelu Biblioteka.
- 3 Przeciągnij składnik DataGrid z panelu Składniki do panelu Biblioteka.
- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:



```

import fl.containers.ScrollPane;
import fl.controls.ScrollPolicy;
import fl.controls.DataGrid;
import fl.data.DataProvider;

var aSp:ScrollPane = new ScrollPane();
var aBox:MovieClip = new MovieClip();
drawBox(aBox, 0xFF0000); //draw a red box

aSp.source = aBox;
aSp.setSize(150, 200);
aSp.move(100, 100);

addChild(aSp);

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1);
    box.graphics.drawRect(0, 0, 150, 300);
    box.graphics.endFill();
}

```

5 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

## Korzystanie ze składnika Slider

Składnik Slider umożliwia wybór wartości poprzez przesunięcie *uchwyty* graficznego między punktami końcowymi paska, który odpowiada zakresowi wartości. Dzięki suwakowi (składnik Slider) użytkownik może wybrać wartość, np. liczbę lub procent. Istnieje również możliwość takiego napisania kodu ActionScript, aby wartość wybrana na suwaku wpływała na działanie innego obiektu. Przykład: suwak można skojarzyć z obrazem i powiększać lub zmniejszać obraz w zależności od względnego położenia albo wartości uchwytu suwaka.

Bieżąca wartość składnika Slider jest określona przez względne położenie uchwytu między punktami końcowymi paska albo między wartościami minimum i maksimum składnika Slider.

Składnik Slider może zawierać ciągły zakres między wartościami minimum i maksimum, ale możliwe jest również ustawienie parametru `snapInterval` w taki sposób, aby określał przerwy między wartościami minimum i maksimum. Na składniku Slider mogą być widoczne znaczniki wyboru, które są niezależne od wartości przypisanych do suwaka i znajdują się w określonych odstępach na całym pasku.

Suwak ma domyślnie orientację poziomą, ale można ją zmienić na pionową, ustawiając dla parametru `direction` wartość pionową. Pasek suwaka rozciąga się między końcami, a znaczniki wyboru są rozmieszczane od lewej do prawej tuż nad paskiem.

## Posługiwanie się składnikiem Slider

Po aktywowaniu instancji Slider można nią sterować za pomocą następujących klawiszy:

Klucz	Opis
Strzałka w prawo	Zwiększa wartość skojarzoną z suwakiem poziomym.
Strzałka w górę	Zwiększa wartość skojarzoną z suwakiem pionowym.
Strzałka w lewo	Zmniejsza wartość skojarzoną z suwakiem poziomym.

Klucz	Opis
Strzałka w dół	Zmniejsza wartość skojarzoną z suwakiem pionowym.
Shift+Tab	Przesuwa aktywność do poprzedniego obiektu.
Karta	Przesuwa aktywność do kolejnego obiektu.

Więcej informacji na temat sterowania aktywnością zawiera opis interfejsu IFocusManager oraz opis klasy FocusManager w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#) oraz w sekcji „Praca z klasą FocusManager” na stronie 29.

Podgląd aktywny każdej instancji Slider odzwierciedla zmiany dokonane w parametrach w Inspektorze właściwości oraz Inspektorze składników podczas tworzenia.

## Parametry składnika Slider

Dla każdej instancji składnika Slider można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry tworzenia treści: `direction`, `liveDragging`, `maximum`, `minimum`, `snapInterval`, `tickInterval` i `value`. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy Slider w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

## Tworzenie aplikacji ze składnikiem Slider

W poniższym przykładzie przedstawiono tworzenie instancji Slider, która umożliwi użytkownikowi wyrażenie poziomu zadowolenia z hipotetycznego zdarzenia. Użytkownik przesuwając suwak (składnik Slider) w prawo lub w lewo, aby określić wysoki lub niski poziom zadowolenia.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik Label z panelu Składniki na środek stołu montażowego.
  - Nadaj instancji nazwę **valueLabel**.
  - Do parametru `text` przypisz wartość **0percent**.
- 3 Przeciągnij składnik Slider z panelu Składniki i wyśrodkuj go poniżej `value_1b1`.
  - Nadaj instancji nazwę **aSlider**.
  - Przypisz składnikowi szerokość (Sz) równą **200**.
  - Przypisz składnikowi wysokość (W) równą **10**.
  - Do parametru `maximum` przypisz wartość **100**.
  - Do parametrów `snapInterval` i `tickInterval` przypisz wartość **10**.
- 4 Przeciągnij kolejną instancję Label z panelu Biblioteka i wyśrodkuj ją poniżej `aSlider`.
  - Nadaj instancji nazwę **promptLabel**.
  - Przypisz instancji szerokość (Sz) równą 250.
  - Przypisz instancji wysokość (W) równą 22.
  - Dla parametru `text` wprowadź tekst **Określ swój poziom zadowolenia**.
- 5 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.Slider;
import fl.events.SliderEvent;
import fl.controls.Label;

aSlider.addEventListener(SliderEvent.CHANGE, changeHandler);

function changeHandler(event:SliderEvent):void {
    valueLabel.text = event.value + "percent";
}
```

#### 6 Wybierz polecenie Sterowanie > Testuj film.

W tym przykładzie podczas przesuwania uchwytu suwaka między odstępami detektor zdarzenia `SliderEvent.CHANGE` aktualizuje właściwość `text` instancji `valueLabel` w celu wyświetlenia wartości procentowej, która odpowiada położeniu wskaźnika.

## Tworzenie aplikacji ze składnikiem Slider przy użyciu kodu ActionScript

W poniższym przykładzie przedstawiono tworzenie składnika Slider za pomocą kodu ActionScript. W przykładzie pobierany jest obraz kwiatka, a składnik Slider umożliwia użytkownikowi ukrywanie i rozjaśnianie obrazu poprzez zmianę jego właściwości `alpha` w taki sposób, aby odpowiadała wartości składnika Slider.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składniki Label i Slider z panelu Składniki do panelu Biblioteka bieżącego dokumentu.  
Dzięki temu składniki zostaną dodane do biblioteki, ale nie będą widoczne w aplikacji.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod w celu utworzenia i ustawienia instancji składnika:

```
import fl.controls.Slider;
import fl.events.SliderEvent;
import fl.controls.Label;
import fl.containers.UILoader;

var sliderLabel:Label = new Label();
sliderLabel.width = 120;
sliderLabel.text = "< Fade - Brighten >";
sliderLabel.move(170, 350);

var aSlider:Slider = new Slider();
aSlider.width = 200;
aSlider.snapInterval = 10;
aSlider.tickInterval = 10;
aSlider.maximum = 100;
aSlider.value = 100;
aSlider.move(120, 330);

var aLoader:UILoader = new UILoader();
aLoader.source = "http://www.flash-mx.com/images/image1.jpg";
aLoader.scaleContent = false;

addChild(sliderLabel);
addChild(aSlider);
addChild(aLoader);

aLoader.addEventListener(Event.COMPLETE, completeHandler);

function completeHandler(event:Event) {
    trace("Number of bytes loaded: " + aLoader.bytesLoaded);
}

aSlider.addEventListener(SliderEvent.CHANGE, changeHandler);

function changeHandler(event:SliderEvent):void {
    aLoader.alpha = event.value * .01;
}
```

- 4 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.
- 5 Przesuń wskaźnik składnika Slider w lewo, aby ukryć obraz, albo w prawo, aby go rozjaśnić.

## Korzystanie ze składnika TextArea

Składnik TextArea jest opakowaniem rodzimego obiektu TextField w języku ActionScript. Składnik TextArea może być używany w celu wyświetlania tekstu oraz w celu edytowania i odbierania danych tekstowych, jeśli dla właściwości `editable` ustawiona jest wartość `true`. Składnik może wyświetlać lub odbierać wiele wierszy tekstu i zawijać długie wiersze tekstu, pod warunkiem że dla właściwości `wordWrap` ustawiona jest wartość `true`. Właściwość `restrict` umożliwia ograniczenie znaków, jakie użytkownik może wprowadzać, a właściwość `maxChars` umożliwia określenie maksymalnej liczby znaków, jaką może wprowadzić użytkownik. Jeśli tekst przekracza poziome i pionowe krawędzie obszaru tekstowego, automatycznie pojawiają się poziome i pionowe paski przewijania, chyba że dla skojarzonych z nimi właściwości `horizontalScrollPolicy` i `verticalScrollPolicy` ustawiono wartość `off`.

Składnik TextArea może być używany wszędzie tam, gdzie wymagane jest wielowierszowe pole tekstowe. Na przykład: składnik TextArea może być używany jako pole komentarza w formularzu. Istnieje możliwość skonfigurowania detektora, który będzie sprawdzał, czy pole jest puste, gdy użytkownik wybierze inne pole za pomocą klawisza Tab. Ten detektor może wyświetlać komunikat o błędzie, który będzie informował o konieczności wprowadzenia komentarza do pola.

Jeśli potrzebne jest jednowierszowe pole tekstowe, należy skorzystać ze składnika TextInput.

Za pomocą metody `setStyle()` można ustawić styl `textFormat` w celu zmiany stylu tekstu, jaki pojawi się w instancji TextArea. Składnik TextArea można również sformatować w kodzie HTML, korzystając z właściwości `htmlText` języka ActionScript. Ustawienie dla właściwości `displayAsPassword` wartości `true` sprawia, że tekst jest wyświetlany w postaci gwiazdek. Jeśli dla właściwości `condenseWhite` zostanie ustawiona wartość `true`, program Flash usunie z nowego tekstu białe znaki tj. spacje, znaki nowych linii itp. Właściwość nie ma wpływu na tekst, który znajduje się już w elemencie sterowania.

## Posługiwanie się składnikiem TextArea

Składnik TextArea może być włączony lub wyłączony. W stanie `disabled` składnik TextArea nie odbiera danych wprowadzanych za pomocą klawiatury lub myszy. Po włączeniu (stan `enabled`) dla tego składnika obowiązują te same reguły aktywacji, zaznaczania i nawigacji, które obowiązują dla obiektu TextField języka ActionScript. Po aktywowaniu instancji TextArea można nią sterować za pomocą następujących klawiszy:

Klucz	Opis
Klawisze strzałek	Przesuwanie punktu wstawiania w górę, w dół, w prawo lub w lewo w obszarze tekstu, pod warunkiem że tekst jest dostępny do edycji.
Page Down	Przesuwa punkt wstawiania do końca tekstu, pod warunkiem że tekst jest dostępny do edycji.
Page Up	Przesuwa punkt wstawiania na początek tekstu, pod warunkiem że tekst jest dostępny do edycji.
Shift+Tab	Przesuwa aktywność do poprzedniego obiektu w pętli tabulacji.
Karta	Przesuwa aktywność do następnego obiektu w pętli tabulacji.

Więcej informacji na temat sterowania aktywnością zawiera opis klasy FocusManager w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#) oraz w sekcji „Praca z klasą FocusManager” na stronie 29.

## Parametry składnika TextArea

Dla każdej instancji składnika TextArea można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry tworzenia treści: `condenseWhite`, `editable`, `horizontalScrollPolicy`, `maxChars`, `restrict`, `text`, `verticalScrollPolicy` i `wordwrap`. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy TextArea w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

Podgląd aktywny każdej instancji TextArea odzwierciedla zmiany dokonane w parametrach w Inspektorze właściwości oraz Inspektorze składników podczas tworzenia. Jeśli wymagany jest pasek przewijania, jest widoczny w podglądzie aktywnym, ale nie działa. W podglądzie aktywnym tekst nie może być zaznaczany i nie można wprowadzać tekstu do instancji składnika na stole montażowym.

Istnieje możliwość napisania kodu ActionScript w celu kontrolowania tych i innych opcji dla składnika TextArea z wykorzystaniem jego właściwości, metod i zdarzeń. Więcej informacji zawiera opis klasy TextArea w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

## Tworzenie aplikacji ze składnikiem TextArea

Poniższa procedura wyjaśnia jak podczas tworzenia dodać składnik TextArea do aplikacji. Przykład przedstawia konfigurację modułu obsługi zdarzeń `focusOut` w instancji TextArea — ten moduł sprawdza, czy przed aktywacją innej części interfejsu użytkownik wpisał tekst do pola tekstowego.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik TextArea z panelu Składniki na stół montażowy i nadaj nazwę instancji **aTa**. Pozostaw dla parametrów ustawienia domyślne.
- 3 Przeciągnij drugi składnik TextArea z panelu Składniki do stołu montażowego, umieść go poniżej pierwszego składnika i nadaj jego instancji nazwę **bTa**. Pozostaw dla parametrów ustawienia domyślne.
- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import flash.events.FocusEvent;

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

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

W tym przykładzie znaki, jakie można wprowadzać do pola tekstowego `aTa`, zostały ograniczone do małych liter, przecinków, apostrofów i spacji. Przykład konfiguruje również moduły obsługi zdarzeń `change`, `KEY_FOCUS_CHANGE` oraz `MOUSE_FOCUS_CHANGE` w obszarze tekstowym `aTa`. Funkcja `changeHandler()` sprawia, że tekst wprowadzony do obszaru tekstowego `aTa` automatycznie pojawia się w obszarze `bTa` — dzieje się tak, dlatego że przy każdym zdarzeniu `change` wartość `aTa.text` zostaje przypisana do wartości `bTa.text`. Funkcja `k_m_fHandler()` dla zdarzeń `KEY_FOCUS_CHANGE` i `MOUSE_FOCUS_CHANGE` uniemożliwia naciśnięcie klawisza `Tab` w celu przejścia do kolejnego pola przed wprowadzeniem tekstu. Funkcja blokuje działanie domyślne.

- 5 Wybierz polecenie Sterowanie > Testuj film.

Jeśli klawisz `Tab` zostanie naciśnięty w celu przejścia do kolejnego obszaru tekstowego przed wprowadzeniem tekstu, powinien pojawić się komunikat o błędzie, a obszar aktywny powinien powrócić do pierwszego obszaru tekstowego. Podczas wprowadzania tekstu do pierwszego obszaru tekstowego ten sam tekst będzie duplikowany w drugim obszarze tekstowym.

## Tworzenie instancji składnika TextArea przy użyciu kodu ActionScript

W poniższym przykładzie przedstawiono tworzenie składnika TextArea za pomocą języka ActionScript. W przykładzie dla właściwości `condenseWhite` zostaje ustawiona wartość `true`, co sprawia, że znaki białe będą zagęszczane, a ponadto do właściwości `htmlText` zostaje przypisany tekst, co umożliwi korzystanie z atrybutów HTML do formatowania tekstu.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik TextArea do panelu Biblioteka.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.TextArea;

var aTa:TextArea = new TextArea();

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

W tym przykładzie użyto właściwości `htmlText` w celu zastosowania atrybutów pogrubienia i podkreślenia HTML do bloku tekstu i wyświetlenia tego tekstu w obszarze tekstowym `a_ta`. W tym przykładzie dla właściwości `condenseWhite` ustawiono wartość `true` w celu zagęszczenia znaków białych w bloku tekstu. Metoda `setSize()` ustawia wysokość i szerokość obszaru tekstowego, a metoda `move()` ustawia położenie tego obszaru. Metoda `addChild()` dodaje instancję `TextArea` do stołu montażowego.

#### 4 Wybierz polecenie Sterowanie > Testuj film.

## Korzystanie ze składnika TextInput

Składnik `TextInput` jest składnikiem pojedynczego wiersza tekstu, który stanowi opakowanie dla rodzimego obiektu `TextField` języka `ActionScript`. Jeśli wymagane jest wielowierszowe pole tekstowe, należy użyć składnika `TextArea`. Na przykład: składnik `TextInput` może być używany jako pole hasła w formularzu. Istnieje możliwość skonfigurowania detektora, który będzie sprawdzał, czy pole zawiera dostateczną liczbę znaków, gdy użytkownik wybierze inne pole za pomocą klawisza `Tab`. Ten detektor może wyświetlać komunikat o błędzie, który będzie informował o konieczności wprowadzenia odpowiedniej liczby znaków.

Za pomocą metody `setStyle()` można ustawić właściwość `textFormat` w celu zmiany stylu tekstu, jaki pojawi się w instancji `TextInput`. Składnik `TextInput` może zostać również sformatowany za pomocą kodu HTML lub jako pole hasła, w którym tekst będzie ukrywany.

### Posługiwanie się składnikiem TextInput

Składnik `TextInput` może być włączony lub wyłączony w aplikacji. W stanie `disabled` składnik `TextArea` nie odbiera danych wprowadzanych za pomocą klawiatury lub myszy. Po włączeniu (stan `enabled`) dla tego składnika obowiązują te same reguły aktywacji, zaznaczania i nawigacji, które obowiązują dla obiektu `TextField` języka `ActionScript`. Po aktywowaniu instancji `TextInput` można nią sterować za pomocą następujących klawiszy:

Klucz	Opis
Klawisze strzałek	Przesuwanie punktu wprowadzania o jeden znak w lewo lub w prawo.
Shift+Tab	Przesuwa aktywność do poprzedniego obiektu.
Karta	Przesuwa aktywność do kolejnego obiektu.

Więcej informacji na temat sterowania aktywnością zawiera opis klasy `FocusManager` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#) oraz w sekcji „Praca z klasą `FocusManager`” na stronie 29.

Podgląd aktywny każdej instancji TextInput odzwierciedla zmiany dokonane w parametrach w Inspektorze właściwości oraz Inspektorze składników podczas tworzenia. W podglądzie aktywnym tekst nie może być zaznaczany i nie można wprowadzać tekstu do instancji składnika na stole montażowym.

Po dodaniu składnika TextInput do aplikacji można go udostępnić dla czytnika ekranu, korzystając z panelu Ułatwienia dostępu.

## Parametry składnika TextInput

Dla każdej instancji składnika TextInput można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry tworzenia treści: `editable`, `displayAsPassword`, `maxChars`, `restrict` i `text`. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy TextInput w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

Istnieje możliwość napisania kodu ActionScript w celu kontrolowania tych i innych opcji dla składnika TextInput z wykorzystaniem jego właściwości, metod i zdarzeń. Więcej informacji zawiera opis klasy TextInput w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

## Tworzenie aplikacji ze składnikiem TextInput

Poniższa procedura wyjaśnia jak podczas tworzenia dodać składnik TextInput do aplikacji. W przykładzie wykorzystano dwa pola TextInput w celu wpisania i potwierdzenia hasła. Przykład przedstawia użycie detektora zdarzeń, który wykrywa, czy do każdego z pól zostało wprowadzonych co najmniej osiem znaków oraz czy teksty w każdym z pól są takie same.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik Label z panelu Składniki na stół montażowy i określ dla niego następujące wartości w Inspektorze właściwości:
  - Wprowadź dla instancji nazwę **pwdLabel**.
  - Wprowadź wartość dla Sz: **100**.
  - Wprowadź wartość dla X: **50**.
  - Wprowadź wartość dla Y: **150**.
  - W sekcji Parametry wprowadź wartość **Hasło**: dla parametru tekstu.
- 3 Przeciągnij drugi składnik Label z panelu Składniki na stół montażowy i określ dla niego następujące wartości:
  - Wprowadź dla instancji nazwę **confirmLabel**.
  - Wprowadź wartość dla Sz: **100**.
  - Wprowadź wartość dla X: **50**.
  - Wprowadź wartość dla Y: **200**.
  - W sekcji Parametry wprowadź wartość **Potwierdźhasło**: dla parametru tekstu.
- 4 Przeciągnij składnik TextInput z panelu Składniki na stół montażowy i określ dla niego następujące wartości:
  - Wprowadź dla instancji nazwę **pwdTi**.
  - Wprowadź wartość dla Sz: **150**.
  - Wprowadź wartość dla X: **190**.
  - Wprowadź wartość dla Y: **150**.



- W sekcji Parametry kliknij dwukrotnie wartość dla parametru `displayAsPassword` i wybierz opcję **true**. To spowoduje, że wartość wprowadzana do pola tekstowego będzie maskowana gwiazdkami.
- 5 Przeciągnij drugi składnik `TextInput` z panelu Składniki na stół montażowy i określ dla niego następujące wartości:
- Wprowadź dla instancji nazwę **confirmTi**.
  - Wprowadź wartość dla Sz: **150**.
  - Wprowadź wartość dla X: **190**.
  - Wprowadź wartość dla Y: **200**.
  - W sekcji Parametry kliknij dwukrotnie wartość dla parametru `displayAsPassword` i wybierz opcję **true**. To spowoduje, że wartość wprowadzana do pola tekstowego będzie maskowana gwiazdkami.
- 6 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
function tiListener(evt_obj:Event){
if(confirmTi.text != pwdTi.text || confirmTi.length < 8)
{
    trace("Password is incorrect. Please reenter it.");
}
else {
    trace("Your password is: " + confirmTi.text);
}
}
confirmTi.addEventListener("enter", tiListener);
```

Ten kod konfiguruje moduł obsługi zdarzeń `enter` w instancji `TextInput` o nazwie `confirmTi`. Jeśli obydwa hasła nie będą takie same lub użytkownik wprowadzi mniej niż osiem znaków, w pojawi się komunikat: „Password is incorrect. Please reenter it”. Jeśli hasła zawierają osiem znaków lub więcej i są zgodne, wprowadzona wartość zostanie wyświetlona w panelu Wyjście.

- 7 Wybierz polecenie Sterowanie > Testuj film.

## Tworzenie instancji składnika `TextInput` przy użyciu kodu ActionScript

W poniższym przykładzie przedstawiono tworzenie składnika `TextInput` za pomocą języka ActionScript. W przykładzie tworzony jest również składnik `Label` wykorzystywany w celu wyświetlenia monitu, który informuje o konieczności wprowadzenia nazwy użytkownika. W przykładzie ustawiana jest właściwość `restrict`, która umożliwia wprowadzanie tylko małych i wielkich liter, kropek i spacji. W przykładzie tworzony jest również obiekt `TextFormat`, który jest wykorzystywany w celu formatowania tekstu w składnikach `Label` i `TextInput`.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik `TextInput` z panelu Składniki do panelu Biblioteka.
- 3 Przeciągnij składnik `Label` z panelu Składniki do panelu Biblioteka.
- 4 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.controls.Label;
import fl.controls.TextInput;

var nameLabel:Label = new Label();
var nameTi:TextInput = new TextInput();
var tf:TextFormat = new TextFormat();

addChild(nameLabel);
addChild(nameTi);

nameTi.restrict = "A-Z .a-z";

tf.font = "Georgia";
tf.color = 0x0000CC;
tf.size = 16;

nameLabel.text = "Name: " ;
nameLabel.setSize(50, 25);
nameLabel.move(100,100);
nameLabel.setStyle("textFormat", tf);
nameTi.move(160, 100);
nameTi.setSize(200, 25);
nameTi.setStyle("textFormat", tf);
```

5 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

## Korzystanie ze składnika TileList

Składnik `TileList` składa się z listy utworzonej z wierszy i kolumn dostarczanych z danymi przez dostawcę danych. *Element* określa jednostkę danych, jaka jest przechowywana w komórce składnika `TileList`. *Element*, którego źródłem jest dostawca danych, zwykle zawiera właściwość `label` i `source`. Właściwość `label` identyfikuje treść wyświetlaną w komórce, a właściwość `source` udostępnia jej wartość.

Istnieje możliwość utworzenia instancji `Array` lub pobrania jej z serwera. Składnik `TileList` zawiera metody, które są odpowiednikami metod dostawcy danych składnika, np. metody `addItem()` i `removeItem()`. Jeśli lista nie zawiera żadnego zewnętrznego dostawcy danych, te metody tworzą automatycznie instancję dostawcy danych, który jest dostępny za pośrednictwem `List.dataProvider`.

### Posługiwanie się składnikiem TileList

Składnik `TileList` renderuje każdą komórkę za pomocą ikonki, która implementuje interfejs `ICellRenderer`. Moduł renderowania można określić za pomocą właściwości `cellRenderer` składnika `TileList`. Domyślny moduł `CellRenderer` składnika `TileList` to obiekt `ImageCell`, który wyświetla obraz (klasę, mapę bitową, instancję lub adres URL), a także opcjonalną etykietę. Etykieta zawiera pojedynczy wiersz, który jest zawsze przypisywany do dołu komórki. Składnik `TileList` można przewijać tylko w jednym kierunku.

Po aktywowaniu instancji `TileList` dostęp do elementów można uzyskiwać za pomocą następujących klawiszy:

Klucz	Opis
Strzałka w górę i strzałka w dół	Umożliwia przechodzenie w górę i w dół w kolumnie. Jeśli właściwość <code>allowMultipleSelection</code> ma wartość <code>true</code> , te klawisze mogą być używane wraz z klawiszem Shift w celu zaznaczania wielu komórek.
Strzałka w lewo i strzałka w prawo	Umożliwia przechodzenie w lewo i w prawo w wierszu. Jeśli właściwość <code>allowMultipleSelection</code> ma wartość <code>true</code> , te klawisze mogą być używane wraz z klawiszem Shift w celu zaznaczania wielu komórek.
Home	Zaznacza pierwszą komórkę składnika <code>TileList</code> . Jeśli właściwość <code>allowMultipleSelection</code> ma wartość <code>true</code> , przytrzymanie klawisza Shift i naciśnięcie klawisza Home spowoduje zaznaczenie wszystkich komórek od aktualnie zaznaczonej do pierwszej.
End	Zaznacza ostatnią komórkę składnika <code>TileList</code> . Jeśli właściwość <code>allowMultipleSelection</code> ma wartość <code>true</code> , przytrzymanie klawisza Shift i naciśnięcie klawisza End spowoduje zaznaczenie wszystkich komórek od aktualnie zaznaczonej do ostatniej.
Ctrl	Jeśli właściwość <code>allowMultipleSelection</code> ma wartość <code>true</code> , możliwe jest zaznaczanie wielu komórek bez określonej kolejności.

Po dodaniu składnika `TileList` do aplikacji można go udostępnić dla czytnika ekranu, dodając do kodu ActionScript następujące wiersze:

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

Ułatwienia dostępu włącza się dla danego składnika tylko raz, niezależnie od liczby instancji składnika. Więcej informacji zawiera rozdział 18, "Tworzenie treści dostępnych dla użytkowników niepełnosprawnych", w podręczniku *Korzystanie z programu Flash*.

## Parametry składnika `TileList`

Dla każdej instancji składnika `TileList` można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry tworzenia treści: `allowMultipleSelection`, `columnCount`, `columnWidth`, `dataProvider`, `direction`, `horizontalScrollLineSize`, `horizontalScrollPageSize`, `labels`, `rowCount`, `rowHeight`, `scrollPolicy`, `verticalScrollLineSize` i `verticalScrollPageSize`. Każdy z parametrów posiada odpowiadającą mu właściwość ActionScript o tej samej nazwie. Informacje o używaniu parametru `dataProvider` zawiera sekcja „[Korzystanie z parametru `dataProvider`](#)” na stronie 31.

W kodzie ActionScript możliwe jest ustawienie dodatkowych opcji dla składnika `TileList` z wykorzystaniem jego metod, właściwości i zdarzeń. Więcej informacji zawiera opis klasy `TileList` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

## Tworzenie aplikacji ze składnikiem `TileList`

W tym przykładzie wykorzystano obiekty `MovieClip` w celu wypełnienia składnika `TileList` tablicą kolorów farb.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik `TileList` na stół montażowy i nadaj instancji nazwę **aTL**.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

```
import fl.data.DataProvider;
import flash.display.DisplayObject;

var aBoxes:Array = new Array();
var i:uint = 0;
var colors:Array = new Array(0x000000, 0xFF0000, 0x0000CC, 0x00CC00, 0xFFFF00);
var colorNames:Array = new Array("Midnight", "Cranberry", "Sky", "Forest", "July");
var dp:DataProvider = new DataProvider();
for(i=0; i < colors.length; i++) {
    aBoxes[i] = new MovieClip();
    drawBox(aBoxes[i], colors[i]); // draw box w next color in array
    dp.addItem( {label:colorNames[i], source:aBoxes[i]} );
}
aTl.dataProvider = dp;
aTl.columnWidth = 110;
aTl.rowHeight = 130;
aTl.setSize(280,150);
aTl.move(150, 150);
aTl.setStyle("contentPadding", 5);

function drawBox(box:MovieClip,color:uint):void {
    box.graphics.beginFill(color, 1.0);
    box.graphics.drawRect(0, 0, 100, 100);
    box.graphics.endFill();
}
```

- 4 Wybierz polecenia Sterowanie > Testuj film, aby przetestować aplikację.

### Tworzenie instancji składnika TileList przy użyciu kodu ActionScript

Ten przykład dynamicznie tworzy instancję klasy TileList i dodaje do niej instancje składników ColorPicker, ComboBox, NumericStepper i CheckBox. W przykładzie utworzono tablicę, która zawiera etykiety i wyświetlane nazwy składnika, a ponadto przypisano tablicę (dp) do właściwości dataProvider składnika TileList. W przykładzie wykorzystano właściwości columnWidth i rowHeight oraz metodę setSize() w celu określenia układu składnika TileList; metodę move() w celu ustawienia składnika na stole montażowym; styl contentPadding w celu umieszczania spacji między krawędziami a treścią instancji TileList oraz metodę sortItemsOn() w celu sortowania treści według etykiet.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij następujące składniki z panelu Składniki do panelu Biblioteka: ColorPicker, ComboBox, NumericStepper, CheckBox i TileList.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:

**Korzystanie ze składników interfejsu użytkownika**

```
import fl.controls.CheckBox;
import fl.controls.ColorPicker;
import fl.controls.ComboBox;
import fl.controls.NumericStepper;
import fl.controls.TileList;
import fl.data.DataProvider;

var aCp:ColorPicker = new ColorPicker();
var aCb:ComboBox = new ComboBox();
var aNs:NumericStepper = new NumericStepper();
var aCh:CheckBox = new CheckBox();
var aTl:TileList = new TileList();

var dp:Array = [
    {label:"ColorPicker", source:aCp},
    {label:"ComboBox", source:aCb},
    {label:"NumericStepper", source:aNs},
    {label:"CheckBox", source:aCh},
];
aTl.dataProvider = new DataProvider(dp);
aTl.columnWidth = 110;
aTl.rowHeight = 100;
aTl.setSize(280,130);
aTl.move(150, 150);
aTl.setStyle("contentPadding", 5);
aTl.sortItemsOn("label");
addChild(aTl);
```

- 4 Wybierz polecenia Sterowanie > Testuj film, aby przetestować aplikację.

## Korzystanie ze składnika UI Loader

Składnik UI Loader jest kontenerem, w którym mogą być wyświetlane pliki SWF, JPEG, progresywne JPEG, PNG i GIF. Składnik UI Loader może być używany zawsze, gdy wymagane jest pobranie treści z lokalizacji zdalnej i umieszczenie jej w aplikacji Flash. Przykład: składnik UI Loader może służyć w celu dodania logo przedsiębiorstwa (plik JPEG) do formularza. Składnik UI Loader może być również używany w aplikacji, która wyświetla zdjęcia. Użyj metody `load()` w celu załadowania treści, właściwości `percentLoaded` w celu określenia ilości treści załadowanej, a zdarzenia `complete` w celu ustalenia, czy ładowanie zostało zakończone.

Możliwe jest skalowanie treści składnika UI Loader lub zmiana wielkości samego składnika UI Loader w celu dostosowania jego wielkości do treści. Domyślnie skala treści umożliwi dopasowanie jej do składnika UI Loader. Treść można również ładować w czasie wykonywania i monitorować postęp ładowania (jednak po załadowaniu treści do pamięci podręcznej postęp szybko wzrasta do 100%). Jeśli położenie zostanie określone podczas ładowania treści do składnika UI Loader, należy określić położenie (współrzędne X i Y) jako 0, 0.

### Posługiwanie się składnikiem UI Loader

Składnik UI Loader nie może być aktywowany. Jednak możliwe jest aktywowanie treści składnika UI Loader i możliwa jest obsługa aktywnej treści. Więcej informacji na temat sterowania aktywnością zawiera opis klasy `FocusManager` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#) oraz w sekcji „Praca z klasą `FocusManager`” na stronie 29.

## Parametry składnika UILoader

Dla każdej instancji składnika UILoader można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry tworzenia treści: `autoLoad`, `maintainAspectRatio`, `source` i `scaleContent`. Każdy z parametrów posiada odpowiadającą mu właściwość `ActionScript` o tej samej nazwie.

Podgląd aktywny każdej instancji UILoader odzwierciedla zmiany dokonane w parametrach w Inspektorze właściwości oraz Inspektorze składników podczas tworzenia.

W kodzie `ActionScript` możliwe jest ustawienie dodatkowych opcji dla składnika UILoader z wykorzystaniem jego metod, właściwości i zdarzeń. Więcej informacji zawiera opis klasy UILoader w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

## Tworzenie aplikacji ze składnikiem UILoader

Poniższa procedura wyjaśnia jak podczas tworzenia dodać składnik UILoader do aplikacji. W tym przykładzie moduł ładowania ładuje obraz GIF logo.

- 1 Utwórz nowy dokument Flash (`ActionScript 3.0`).
- 2 Przeciągnij składnik UILoader z panelu Składniki na stół montażowy.
- 3 W Inspektorze właściwości wprowadź dla instancji nazwę **aUI**.
- 4 Wybierz moduł ładowania na stole montażowym oraz w Inspektorze składników, a następnie wprowadź wartość <http://www.helpexamples.com/images/logo.gif> dla parametru `source`.

## Tworzenie instancji składnika UILoader przy użyciu kodu ActionScript

W tym przykładzie tworzony jest składnik UILoader za pomocą języka `ActionScript` i ładowany jest obraz JPEG kwiatka. Gdy wystąpi zdarzenie `complete`, następuje wyświetlenie liczby bajtów załadowanych na panelu Wyjście.

- 1 Utwórz nowy dokument Flash (`ActionScript 3.0`).
- 2 Przeciągnij składnik UILoader z panelu Składniki do panelu Biblioteka.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod `ActionScript`:

```
import fl.containers.UILoader;

var aLoader:UILoader = new UILoader();
aLoader.source = "http://www.flash-mx.com/images/image1.jpg";
aLoader.scaleContent = false;
addChild(aLoader);

aLoader.addEventListener(Event.COMPLETE, completeHandler);
function completeHandler(event:Event) {
    trace("Number of bytes loaded: " + aLoader.bytesLoaded);
}
```

- 4 Wybierz polecenie Sterowanie > Testuj film.

## Korzystanie ze składnika UIScrollBar

Składnik UIScrollBar umożliwia dodawanie paska przewijania do pola tekstowego. Pasek przewijania można dodać do pola tekstowego podczas tworzenia lub w czasie wykonywania w ActionScript. Aby użyć składnika UIScrollBar, utwórz pole tekstowe na stole montażowym, a następnie przeciągnij składnik UIScrollBar z panelu Składniki do dowolnego kwadrantu w obwiedni pola tekstowego.

Jeśli długość paska przewijania jest mniejsza niż łączna długość strzałek przewijania, pasek nie będzie wyświetlany poprawnie. Jeden z przycisków strzałki zostanie ukryty za drugim. W programie Flash nie jest dostępna funkcja kontroli takiego błędu. W takim przypadku dobrym sposobem jest ukrycie paska przewijania za pomocą kodu ActionScript. Jeśli pasek przewijania ma taką wielkość, że nie wystarcza miejsca na pole przewijania, program Flash ukryje pole przewijania.

Składnik UIScrollBar działa jak każdy inny pasek przewijania. Zawiera przyciski strzałek na każdym końcu, a także ścieżkę i pole przewijania między strzałkami. Składnik można dołączyć do dowolnej krawędzi pola tekstowego i używany w poziomie lub w pionie.

Informacje na temat klasy TextField zawiera opis klasy TextField w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

### Posługiwanie się składnikiem UIScrollBar

Składnik UIScrollBar — w przeciwieństwie do wielu innych składników — może odbierać dane wejściowe z myszy wprowadzane w sposób ciągły, np. gdy użytkownik przytrzymuje przycisk myszy — wówczas nie występują powtarzalne kliknięcia.

Składnika UIScrollBar nie można obsługiwać za pomocą klawiatury.

### Parametry składnika UIScrollBar

Dla każdej instancji składnika UIScrollBar można w Inspektorze właściwości lub w Inspektorze składników ustawić następujące parametry tworzenia treści: `direction` i `scrollTargetName`. Każdy z parametrów posiada odpowiadającą mu właściwość ActionScript o tej samej nazwie.

W kodzie ActionScript możliwe jest ustawienie dodatkowych opcji dla składnika UIScrollBar z wykorzystaniem jego klas, metod, właściwości i zdarzeń. Więcej informacji zawiera opis klasy UIScrollBar w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

### Tworzenie aplikacji ze składnikiem UIScrollBar

Poniższa procedura opisuje jak podczas tworzenia dodać składnik UIScrollBar do aplikacji.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Utwórz dynamiczne pole tekstowe o wysokości wystarczającej dla jednego lub dwu wierszy tekstu, a następnie nadaj instancji nazwę `myText` w Inspektorze właściwości.
- 3 W Inspektorze właściwości ustaw dla opcji Typ wiersza pola wprowadzania tekstu wartość Wielowierszowy lub Wielowierszowy bez zawijania, jeśli planowane jest użycie paska przewijania w poziomie.
- 4 Otwórz panel Operacje, wybierz Klatkę nr 1 na głównej osi czasu, a następnie wprowadź następujący kod ActionScript w celu wprowadzenia wartości do właściwości `text` w taki sposób, aby użytkownik musiał przewinąć w celu wyświetlenia całości:

```
myText.text="When the moon is in the seventh house and Jupiter aligns with Mars, then peace will guide the planet and love will rule the stars."
```

***Uwaga:** Upewnij się, że pole tekstowe na stole montażowym jest dostatecznie małe, aby konieczne było jego przewijanie w celu wyświetlenia całego tekstu. Jeśli pole nie jest tak małe, pasek przewijania może być niewidoczny lub może się pojawić jako dwa wiersze bez wskaźnika (część przeciągana w celu przewijania).*

- 5 Upewnij się, że przyciąganie do obiektów jest włączone (Widok > Przyciąganie > Przyciągaj do obiektów).
- 6 Przeciągnij instancję UIScrollBar z panelu Składniki do pola wprowadzania tekstu obok strony, do której chcesz dołączyć instancję. Podczas zwalniania przycisku myszy składnik musi nachodzić na pole tekstowe — tylko wówczas składnik będzie poprawnie powiązany z polem. Nadaj instancji nazwę **mySb**.

Właściwość `scrollTargetName` składnika automatycznie zostanie wypełniona nazwą instancji pola tekstowego z Inspektora właściwości i Inspektora składników. Jeśli nazwa właściwości nie będzie widoczna na karcie Parametry, możliwe, że instancja składnika UIScrollBar nie nachodziła dostatecznie na pole tekstowe.

- 7 Wybierz polecenie Sterowanie > Testuj film.

### **Tworzenie instancji składnika UIScrollBar przy użyciu kodu ActionScript**

Za pomocą języka ActionScript można utworzyć instancję UIScrollBar i skojarzyć ją z polem tekstowym w czasie wykonywania. Poniższy przykład przedstawia tworzenie instancji poziomego składnika UIScrollBar oraz dołączanie go do dołu instancji pola tekstowego o nazwie **myTxt** — to pole jest ładowane z adresu URL. Przykład ustawia również wielkość paska przewijania zgodną z wielkością pola tekstowego:

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ScrollBar do panelu Biblioteka.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod ActionScript:



```
import flash.net.URLLoader;
import fl.controls.UIScrollBar;
import flash.events.Event;

var myTxt:TextField = new TextField();
myTxt.border = true;
myTxt.width = 200;
myTxt.height = 16;
myTxt.x = 200;
myTxt.y = 150;

var mySb:UIScrollBar = new UIScrollBar();
mySb.direction = "horizontal";
// Size it to match the text field.
mySb.setSize(myTxt.width, myTxt.height);

// Move it immediately below the text field.
mySb.move(myTxt.x, myTxt.height + myTxt.y);

// put them on the Stage
addChild(myTxt);
addChild(mySb);
// load text
var loader:URLLoader = new URLLoader();
var request:URLRequest = new URLRequest("http://www.helpexamples.com/flash/lorem.txt");
loader.load(request);
loader.addEventListener(Event.COMPLETE, loadcomplete);

function loadcomplete(event:Event) {
    // move loaded text to text field
    myTxt.text = loader.data;
    // Set myTxt as target for scroll bar.
    mySb.scrollTarget = myTxt;
}
```

4 Wybierz polecenie Sterowanie > Testuj film.

# Rozdział 5: Dostosowywanie składników interfejsu użytkownika

## Informacje o dostosowywaniu składnika interfejsu użytkownika

Wygląd składników aplikacji można dostosować przez modyfikację jednego lub obu poniższych z elementów:

**Style** Każdy składnik ma zestaw stylów, które można ustawić w celu określenia, z jakich wartości program Flash będzie korzystał podczas renderowania wyglądu składnika. Style określają karnacje i ikony do użycia dla różnych stanów składnika, a także wartości używane do formatowania i ustawiania szerokości wypełnienia tekstu.

**Karnacje** *Karnacja* zawiera kolekcję symboli składających się na postać graficzną składnika w danym stanie. Styl określa karnację do użycia, natomiast karnacja jest elementem graficznym używanym przez program Flash do narysowania składnika. *Nakładanie karnacji* jest procesem zmiany wyglądu składnika przez modyfikację lub zastąpienie jego grafiki.

**Uwaga:** Domyślnym wyglądem składników języka ActionScript 3.0 może być temat (*Aeon Halo*), są to jednak karnacje wybudowane w składniki. W składnikach języka ActionScript 3.0 nie są obsługiwane zewnętrzne pliki tematów, tak jak w składnikach języka ActionScript 2.0.

## Ustawianie stylów

Style składnika określają wartości jego karnacji, ikon, formatowania tekstu i szerokości wypełnienia podczas rysowania przez program Flash składnika w różnych stanach. Na przykład, w programie Flash przycisk rysowany jest z inną karnacją, aby pokazać, że znajduje się on w stanie down; stan ten występuje w momencie kliknięcia myszą na przycisku, po czym przycisk powraca on do stanu up lub stanu normalnego. W programie inna karnacja używana jest także dla stanu „wyłączony” przez ustawienie właściwości `enabled` na wartość `false`.

Style składników można ustawiać na poziomach dokumentu, klasy i instancji. Ponadto niektóre właściwości stylu mogą być dziedziczone ze składnika nadrzędnego. Na przykład składnik `List` dziedziczy style `ScrollBar`, dziedzicząc po `BaseScrollPane`.

W celu dostosowania składnika można ustawić style w jeden z następujących sposobów:

- Należy ustawić style dla instancji składnika. Właściwości koloru i tekstu można zmieniać dla pojedynczej instancji składnika. W niektórych sytuacjach jest to wydajna metoda, jednak może być ona czasochłonna, jeśli konieczne jest oddzielne ustawienie właściwości dla wszystkich składników dokumentu.
- Należy ustawić w dokumencie style dla wszystkich składników danego typu. Jeśli dla wszystkich składników danego typu zastosowany ma zostać zgodny wygląd, np. dla wszystkich składników `CheckBox` lub `Button` w dokumencie, należy ustawić style na poziomie składnika.

Wartości dla właściwości stylu ustawione w kontenerach dziedziczone są przez zawarte w nim składniki.

Zmiany dokonane we właściwościach stylu nie są prezentowane w programie Flash podczas wyświetlania składników na stole montażowym za pomocą funkcji Podgląd aktywny.

## Informacje o ustawieniach stylu

Poniżej przedstawiono kilka najważniejszych zasad dotyczących używania stylów:

**Dziedziczenie** Składnik podrzędny należy ustawić tak, aby dziedziczył styl od składnika nadrzędnego. W języku ActionScript nie można ustawić dziedziczenia dla stylów.

**Pierwszeństwo** Jeśli styl składnika został ustawiony na więcej niż jeden sposób, w programie Flash użyty zostanie pierwszy napotkany styl według pierwszeństwa. W programie Flash style wyszukiwane są, aż do momentu znalezienia wartości w następującej kolejności:

- 1 Właściwość stylu wyszukiwana jest przez program Flash w instancji składnika.
- 2 Jeśli styl jest jednym z dziedziczących stylów, dziedziczona wartość wyszukiwana będzie w programie Flash w hierarchii składnika nadrzędnego.
- 3 Styl wyszukiwany jest przez program Flash w składniku.
- 4 Ustawienie globalne wyszukiwane jest przez program Flash w klasie StyleManager.
- 5 Jeśli właściwość pozostaje wciąż niezdefiniowana, wartością właściwości jest `undefined`.

## Uzyskiwanie dostępu do domyślnych stylów składnika

Dostęp do stylów domyślnych składnika można uzyskać za pomocą metody statycznej `getStyleDefinition()` klasy składnika. Na przykład, w poniższym kodzie dla składnika `ComboBox` pobierane są style domyślne i wyświetlane są wartości domyślne dla właściwości `buttonWidth` i `downArrowDownSkin`:

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

## Ustawianie i pobieranie stylów instancji składnika

Dowolna instancja składnika interfejsu użytkownika może bezpośrednio wywoływać metody `setStyle()` i `getStyle()` w celu ustawienia lub pobrania stylu. Następująca składnia ustawia styl i wartość dla instancji składnika:

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

Składnia służąca do pobierania stylu dla instancji składnika:

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

Należy zauważyć, że metoda `getStyle()` zwraca typ `Object`, ponieważ może zwracać wiele stylów o różnych typach danych. Na przykład, poniższy kod ustawia styl czcionki dla instancji `TextArea` (`aTa`), a następnie pobiera go za pomocą metody `getStyle()`. W przykładzie zwrócona wartość rzutowana jest na obiekt `TextFormat` w celu przypisania jej do zmiennej `TextFormat`. Brak rzutowania spowodowałby, że kompilator zgłosi błąd powodowany próbą sprowadzenia zmiennej `Object` do zmiennej `TextFormat`.

```
import flash.text.TextFormat;

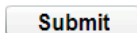
var tf:TextFormat = new TextFormat();
tf.font = "Georgia";
aTa.setStyle("textFormat", tf);
aTa.text = "Hello World!";
var aStyle:TextFormat = aTa.getStyle("textFormat") as TextFormat;
trace(aStyle.font);
```

## Korzystanie z obiektu `TextFormat` w celu ustawiania właściwości tekstu

Obiekt `TextFormat` służy do formatowania tekstu instancji składnika. Obiekt `TextFormat` ma właściwości, które umożliwiają określenie własności tekstu takich, jak `bold`, `bullet`, `color`, `font`, `italic`, `size` i kilka innych. Właściwości te można ustawić w obiekcie `TextFormat`, a następnie wywołać metodę `setStyle()` w celu zastosowania ich w instancji składnika. Na przykład, poniższy kod ustawia właściwości `font`, `size` i `bold` obiektu `TextFormat` i stosuje je w instancji `Button`:

```
/* Create a new TextFormat object to set text formatting properties. */
var tf:TextFormat = new TextFormat();
tf.font = "Arial";
tf.size = 16;
tf.bold = true;
a_button.setStyle("textFormat", tf);
```

Na poniższej ilustracji przedstawiono efekt tych ustawień dla przycisku z etykietą `Submit`:



Właściwości stylu ustawione w instancji składnika za pomocą metody `setStyle()` mają najwyższy priorytet i przesłaniają wszystkie pozostałe ustawienia stylu. Jednak im więcej właściwości zostanie ustawionych za pomocą metody `setStyle()` dla pojedynczej instancji składnika, tym wolniej składnik ten będzie renderowany w czasie wykonywania.

## Ustawianie stylów dla wszystkich instancji składnika

Styl dla wszystkich instancji klasy składnika można ustawić za pomocą metody statycznej `setComponentStyle()` klasy `StyleManager`. Na przykład, dla wszystkich składników `Button` można ustawić czerwony kolor tekstu przez przeciągnięcie składnika `Button` na stół montażowy, a następnie dodanie poniższego kodu ActionScript do panelu Operacje klatki nr 1 na osi czasu:

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

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

Wszystkie składniki `Button` kolejno dodawane na stole montażowym będą miały czerwone etykiety.

## Ustawianie stylu dla wszystkich składników

Styl dla wszystkich składników można ustawić za pomocą metody statycznej `setStyle()` klasy `StyleManager`.

- 1 Przeciągnij składnik `List` na stół montażowy i nadaj instancji nazwę **aList**.
- 2 Przeciągnij składnik `Button` na stół montażowy i nadaj instancji nazwę **aButton**.
- 3 Naciśnij klawisz **F9** lub wybierz opcję Operacje z menu Okno, aby otworzyć panel Operacje, jeśli nie jest on jeszcze otwarty, i wprowadź poniższy kod do klatki nr.1 na osi czasu, aby ustawić czerwony kolor tekstu dla wszystkich składników:

```
import fl.managers.StyleManager;

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

- 4 Dodaj poniższy kod w panelu Operacje, aby wypełnić składnik List tekstem.

```
aList.addItem({label:"1956 Chevy (Cherry Red)", data:35000});  
aList.addItem({label:"1966 Mustang (Classic)", data:27000});  
aList.addItem({label:"1976 Volvo (Xcllnt Cond)", data:17000});  
aList.allowMultipleSelection = true;
```

- 5 Wybierz opcje Sterowanie > Testuj film lub naciśnij klawisze Ctrl+Enter, aby skompilować kod i przetestować jego zawartość. Tekst w etykiecie przycisku i na liście powinien być czerwony.

## Informacje o karnacjach

Na wygląd składnika składają się elementy graficzne takie, jak obrys, wypełnienie kolorem, ikony, a także inne składniki. ComboBox na przykład zawiera składnik List, a składnik List zawiera ScrollBar. Razem elementy graficzne składają się na wygląd ComboBox. Jednak wygląd składnika zmienia się w oparciu o jego bieżący stan. Na przykład, CheckBox bez etykiety wygląda w aplikacji użytkownika mniej więcej tak:



*CheckBox w normalnym stanie up*

Jeśli użytkownik kliknie i przytrzyma przycisk myszy na składniku CheckBox, jego wygląd zmieni się na następujący:



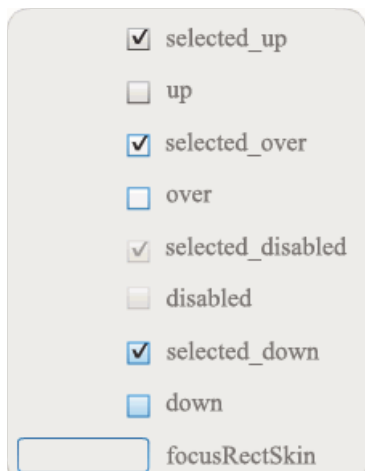
*CheckBox w stanie down*

Po zwolnieniu przycisku myszy CheckBox powróci do pierwotnego wyglądu; będzie miał on jednak symbol zaznaczenia wskazujący, że został wybrany.



*CheckBox w stanie selected*

Ogólnie ikony reprezentujące składnik i jego różne stany nazywane są jego *karnacjami*. Wygląd składnika można zmienić w dowolnym lub we wszystkich jego stanach, edytując jego karnacje w programie Flash tak, jak wszystkie pozostałe symbole Flash. Dostęp do karnacji składnika można uzyskać na dwa sposoby. Najprostszym sposobem jest przeciągnięcie składnika na stół montażowy i dwukrotne jego kliknięcie. Powoduje to otwarcie palety karnacji składnika, która dla składnika CheckBox wygląda następująco.



Karnacje składnika CheckBox

Oddzielny dostęp do karnacji składnika można uzyskać także w panelu Biblioteka. Podczas przeciągania składnika na stół montażowy jest on również kopiowany do biblioteki razem z folderem jego zasobów oraz wszystkimi pozostałymi zawartymi w nim składnikami. Na przykład, jeśli ComboBox przeciągnięte zostanie na stół montażowy, panel Biblioteka będzie zawierał także składniki List, ScrollBar i TextInput, które wbudowane są w ComboBox, oraz folder karnacji dla każdego z tych składników i folder Shared Assets, który zawiera elementy współdzielone przez składniki. Karnację dla dowolnego z tych składników można edytować, otwierając folder karnacji danego składnika (ComboBoxSkins, ListSkins, ScrollBarSkins lub TextInputSkins) i dwukrotnie klikając ikonę karnacji, która ma zostać poddana edycji. Dwukrotne kliknięcie np. ComboBox\_downSkin powoduje otwarcie karnacji w trybie edycji symbolu tak, jak przedstawione to zostało na poniższej ilustracji:



ComboBox\_downSkin

## Tworzenie nowej karnacji

Jeśli dla składnika dokumentu utworzony ma zostać nowy wygląd, należy dokonać edycji karnacji składnika w celu zmiany ich sposobu wyświetlania. Aby uzyskać dostęp do karnacji składnika, należy dwukrotnie kliknąć składnik na stole montażowym w celu otwarcia palety jego karnacji. Następnie należy dwukrotnie kliknąć karnację do edycji w celu otwarcia jej w trybie edycji symbolu. Na przykład, dwukrotne kliknięcie na stole montażowym składnika TextArea powoduje otwarcie jego zasobów w trybie edycji symbolu. Aby zmienić wygląd symbolu, należy ustawić element sterujący powiększenia na wartość 400% lub większą i dokonać edycji symbolu. Po zakończeniu zmiana zostanie wprowadzona we wszystkich instancjach składnika w dokumencie. Alternatywnym sposobem jest dwukrotne kliknięcie konkretnej karnacji w panelu Biblioteka w celu otwarcia jej na stole montażowym w trybie edycji symbolu.

Modyfikacji karnacji składnika można dokonać w następujący sposób:

- Tworzenie nowej karnacji dla wszystkich instancji
- Tworzenie nowych karnacji dla wybranych instancji

## Tworzenie karnacji dla wszystkich instancji

Podczas edycji karnacji składnika domyślnie zmieniany jest wygląd składnika dla wszystkich jej instancji w dokumencie. Jeśli dla tego samego składnika utworzone mają zostać różne wyglądy, należy powielić karnacje do zmiany, nadać im różne nazwy, dokonać ich edycji, a następnie ustawić odpowiednie style w celu ich zastosowania. Więcej informacji zawiera sekcja „[Tworzenie karnacji dla wybranych instancji](#)” na stronie 107.

W tym rozdziale opisane zostały sposoby zmiany co najmniej jednej karnacji dla każdego ze składników interfejsu użytkownika. Jeśli w celu zmiany co najmniej jednej z karnacji składnika interfejsu użytkownika wykonana zostanie jedna z tych z procedur, karnacja zostanie zmieniona dla wszystkich instancji w dokumencie.

## Tworzenie karnacji dla wybranych instancji

Karnację dla wybranych instancji składnika można utworzyć za pomocą następującej procedury ogólnej:

- Wybierz karnację w folderze Zasoby składnika w panelu Biblioteka.
- Powiel karnację i przypisz do niej unikalną nazwę klasy.
- Dokonaj edycji karnacji w celu nadania jej pożądanego wyglądu.
- Wywołaj dla instancji składnika metodę `setStyle()`, aby przypisać nową karnację do stylu karnacji.

Poniższa procedura tworzy nową karnację `selectedDownSkin` dla jednej z dwóch instancji `Button`.

- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij dwa składniki `Button` z panelu Składniki na stół montażowy i nadaj im nazwy instancji **aButton** i **bButton**.
- 3 Otwórz panel Biblioteka, a następnie wewnątrz niego foldery Zasoby składników i `ButtonSkins`.
- 4 Kliknij karnację `selectedDownSkin`, aby ją wybrać.
- 5 Kliknij prawym klawiszem myszy, aby otworzyć menu kontekstowe i wybierz opcję Powiel.
- 6 W oknie dialogowym Powiel symbol nadaj nowej karnacji unikalną nazwę, na przykład **Button\_mySelectedDownSkin**. Kliknij OK.
- 7 W folderze Biblioteka > Zasoby składników > `ButtonSkins` wybierz karnację `Button_mySelectedDownSkin` i kliknij prawym przyciskiem myszy, aby otworzyć menu kontekstowe. Wybierz opcję Połączenie, aby otworzyć okno dialogowe Właściwości połączenia.
- 8 Kliknij pole wyboru Eksportuj do ActionScript. Pozostaw zaznaczone pole wyboru Eksport w pierwszej klatce i upewnij się, że nazwa instancji jest unikalna. Kliknij przycisk OK, a następnie ponownie OK w odpowiedzi na ostrzeżenie, informujące o tym, że definicja klasy nie została odnaleziona i zostanie ona utworzona.
- 9 W panelu Biblioteka dwukrotnie kliknij karnację `Button_mySelectedDownSkin`, aby otworzyć ją w trybie edycji symbolu.
- 10 Klikaj niebieskie wypełnienie na środku karnacji do momentu pojawienia się koloru w wybieraku Koloru wypełnienia w Inspektorze właściwości. Kliknij wybierak koloru i wybierz dla wypełnienia karnacji kolor `#00CC00`.
- 11 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 12 W Inspektorze właściwości dla każdego przycisku kliknij kartę Parametry i ustaw parametr `toggle` na wartość `true`.
- 13 Dodaj poniższy kod do panelu Operacje w klatce nr 1 na osi czasu.

```
bButton.setStyle("selectedDownSkin", Button_mySelectedDownSkin);
bButton.setStyle("downSkin", Button_mySelectedDownSkin);
```
- 14 Wybierz polecenie Sterowanie > Film tekstowy.

15 Kliknij każdy przycisk. Należy zauważyć, że karnacja dla stanu down (zaznaczony i niezaznaczony) obiektu `bButton` używa nowego symbolu karnacji.

## Dostosowywanie składnika Button

Składnik `Button` można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub dowolnej z odpowiednich właściwości klasy `Button`, takich jak `height`, `width`, `scaleX` czy `scaleY`.

Zmiana rozmiaru przycisku nie powoduje zmiany rozmiaru ikony, ani etykiety. Obwódka składnika `Button` odpowiada jego granicy i wskazuje obszar klikania dla instancji. Jeśli zwiększony zostanie rozmiar instancji, zwiększony zostanie również rozmiar obszaru klikania. Jeśli obwódka jest zbyt mała, aby dopasować etykietę, etykieta zostanie przycięta.

Jeśli składnik `Button` zawiera ikonę i jest ona większa niż składnik, ikona będzie rozciągać się poza jego granicami.

## Używanie stylów ze składnikiem Button

Style składnika `Button` określają wartości jego karnacji, ikon, formatowania tekstu i szerokości wypełnienia podczas rysowania składnika w różnych stanach.

Poniższa procedura umieszcza dwa składniki `Button` na stole montażowym i ustawia właściwość `emphasized` na wartość `true` dla obu składników po kliknięciu przez użytkownika jednego z nich. Ustawia też styl `emphasizedSkin` dla drugiego składnika `Button` na styl `selectedOverSkin` po kliknięciu go przez użytkownika, dlatego dwa składniki `Button` pokazują różne karnacje dla tego samego stanu.

- 1 Utwórz plik Flash (ActionScript 3.0).
- 2 Przeciągnij za jednym razem na stół montażowy dwa składniki `Button` i nadaj im nazwy instancji `aBtn` i `bBtn`. Na karcie Parametry w Inspektorze właściwości nadaj im etykiety `Button A` i `Button B`.
- 3 Dodaj poniższy kod do panelu Operacje w klatce nr 1 na osi czasu.

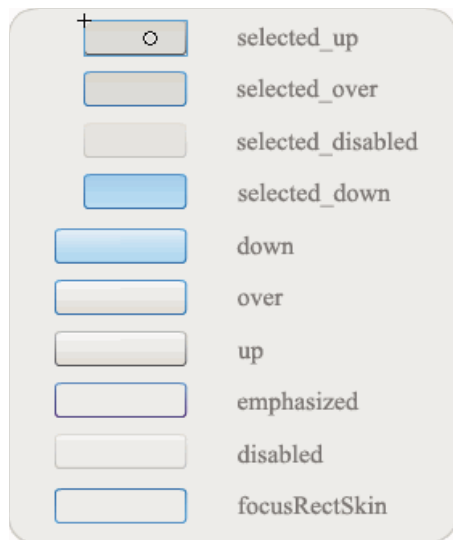
```
bBtn.emphasized = true;
aBtn.emphasized = true;
bBtn.addEventListener(MouseEvent.CLICK, Btn_handler);
function Btn_handler(evt:MouseEvent):void {
    bBtn.setStyle("emphasizedSkin", "Button_selectedOverSkin");
}
```

- 4 Wybierz polecenie Sterowanie > Testuj film.
- 5 Kliknij jeden z przycisków, aby wyświetlić efekt zastosowania stylu `emphasizedSkin` dla każdego z nich.

## Używanie karnacji ze składnikiem Button

Składnik `Button` używa poniższych karnacji, które odpowiadają jego różnym stanom. Aby dokonać edycji co najmniej jednej karnacji w celu zmiany wyglądu składnika `Button`, należy dwukrotnie kliknąć instancję `Button` na stole montażowym, aby otworzyć paletę karnacji tak, jak przedstawione to zostało na poniższej ilustracji:



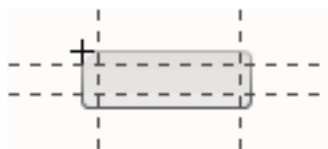


Karnacje składnika Button

Jeśli przycisk jest włączony, po przesunięciu na niego wskaźnika wyświetlany jest stan over. W momencie naciśnięcia przycisku, przycisk aktywowany jest do wprowadzania i wyświetlany jest stan down. Przycisk powraca do stanu over po zwolnieniu przycisku myszy. Jeśli wskaźnik z naciśniętym przyciskiem myszy zostanie przesunięty poza przycisk, powróci on do pierwotnego stanu. Jeśli parametr toggle ma wartość `true`, stan naciśnięcia wyświetlany jest z karnacją `selectedDownSkin`, stan up z karnacją `selectedUpSkin`, a stan over z karnacją `selectedOverSkin`.

Jeśli składnik Button jest wyłączony, stan disabled jest wyświetlany niezależnie od działania użytkownika.

Aby dokonać edycji jednej z karnacji, należy kliknąć ją dwukrotnie w celu otwarcia w trybie edycji symbolu w sposób przedstawiony na poniższej ilustracji:



Składnik Button w trybie edycji symbolu

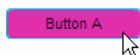
W tym miejscu do edycji karnacji wg upodobań użytkownika można użyć narzędzi do tworzenia treści programu Flash.

Poniższa procedura zmienia kolor karnacji `selected_over` dla składnika Button.

- 1 Utwórz nowy plik Flash (ActionScript 3.0).
- 2 Przeciągnij składnik Button z panelu Składniki na stół montażowy. Na karcie Parametry ustaw parametr toggle na wartość `true`.
- 3 Kliknij dwukrotnie składnik Button, aby otworzyć paletę karnacji.
- 4 Kliknij dwukrotnie karnację `selected_over`, aby otworzyć ją w trybie edycji symbolu.
- 5 Ustaw element sterujący powiększenia na wartość 400%, aby powiększyć ikonę do edycji.
- 6 Klikaj dwukrotnie tło do momentu pojawienia się jego koloru w wybieraku Koloru wypełnienia w Inspektorze właściwości.
- 7 Wybierz kolor `#CC0099` w wybieraku Koloru wypełnienia w celu zastosowania go do tła karnacji `selected_over`.

- 8 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 9 Wybierz polecenie Sterowanie > Testuj film.
- 10 Kliknij przycisk, aby umieścić go w wybranym stanie.

Po przesunięciu wskaźnika myszy nad składnik Button, stan `selected_over` powinien zostać wyświetlony w sposób przedstawiony na poniższej ilustracji.



*Przycisk przedstawiający karnację `selected_over` ze zmodyfikowanym kolorem*

## Dostosowywanie składnika CheckBox

Składnik CheckBox można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub odpowiednich właściwości klasy CheckBox. Na przykład, można zmienić rozmiar składnika CheckBox, ustawiając właściwości `height`, `width`, `scaleX` czy `scaleY`. Zmiana rozmiaru składnika CheckBox nie powoduje zmiany rozmiaru etykiety ani ikony pola wyboru; zmiana ulega tylko rozmiar obwiedni.

Obwiednia instancji CheckBox jest niewidoczna i wskazuje obszar klikania dla instancji. Jeśli zwiększony zostanie rozmiar instancji, zwiększony zostanie również rozmiar obszaru klikania. Jeśli obwiednia jest zbyt mała, aby dopasować etykietę, etykieta zostanie przycięta.

### Używanie stylów ze składnikiem CheckBox

Właściwości stylu można ustawiać w celu zmiany wyglądu instancji CheckBox. Na przykład, poniższa procedura powoduje zmianę rozmiaru i koloru etykiety składnika CheckBox.

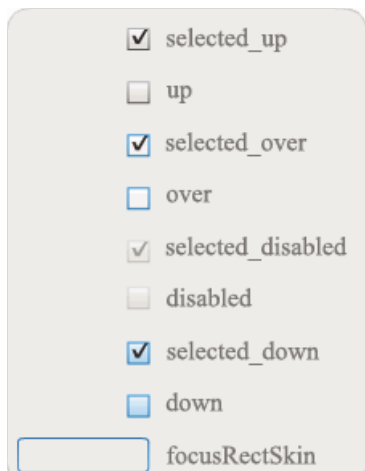
- 1 Przeciągnij składnik CheckBox z panelu Składniki na stół montażowy i nadaj instancji nazwę **myCb**.
- 2 Kliknij kartę Parametry w Inspektorze właściwości i wprowadź następującą wartość dla parametru etykiety: **Less than \$500?**
- 3 W panelu Operacje do klatki nr 1 na głównej osi czasu wprowadź następujący kod:

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

Więcej informacji zawiera sekcja „[Ustawianie stylów](#)” na stronie 102. Informacje na temat ustawiania właściwości stylu w celu zmiany ikon i karnacji składnika zawiera sekcja „[Tworzenie nowej karnacji](#)” na stronie 106 i „[Używanie karnacji ze składnikiem CheckBox](#)” na stronie 110.

### Używanie karnacji ze składnikiem CheckBox

Składnik CheckBox zawiera poniższe karnacje, które można edytować w celu zmiany jego wyglądu.



Karnacje składnika CheckBox

W tym przykładzie zmieniany jest kolor konturu oraz kolor tła składnika dla jego stanów `up` i `selectedUp`. Podobne kroki należy wykonać w celu zmiany karnacji pozostałych stanów.

- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij składnik CheckBox na stół montażowy; powoduje to również umieszczenie składnika w bibliotece razem z folderem jego zasobów.
- 3 Kliknij dwukrotnie na stole montażowym składnik CheckBox, aby otworzyć jego panel ikon karnacji.
- 4 Kliknij dwukrotnie ikonę `selected_up`, aby otworzyć ją w trybie edycji symbolu.
- 5 Ustaw element sterujący powiększenia na wartość 800%, aby powiększyć ikonę do edycji.
- 6 Kliknij krawędź składnika CheckBox, aby ją zaznaczyć. Użyj wybieraka Koloru wypełnienia w Inspektorze właściwości, aby wybrać kolor `#0033FF` i zastosować go dla krawędzi.
- 7 Kliknij dwukrotnie tło składnika CheckBox, aby je zaznaczyć i ponownie użyj wybieraka Koloru wypełnienia w celu ustawienia koloru tła na wartość `#00CCFF`.
- 8 Powtórz kroki od 4 do 8 dla karnacji `up` składnika CheckBox.
- 9 Wybierz polecenie Sterowanie > Testuj film.

## Dostosowywanie składnika ColorPicker

Jedyna zmiana rozmiaru, której można użyć dla składnika ColorPicker, dokonywana jest za pomocą jego stylów: `swatchWidth`, `swatchHeight`, `backgroundPadding`, `textFieldWidth` i `textFieldHeight`. Przy próbie zmiany rozmiaru składnika ColorPicker za pomocą narzędzia Przekształć lub języka ActionScript z wykorzystaniem metody `setSize()` lub właściwości `width`, `height`, `scaleX` bądź `scaleY`, wartości te zostaną zignorowane podczas tworzenia pliku SWF, a składnik ColorPicker zostanie wyświetlony w rozmiarze domyślnym. Rozmiar tła palety zostanie zmieniony, aby odpowiadał liczbie kolumn ustawionych za pomocą metody `setStyle()` dla stylu `columnCount`. Domyślna liczba kolumn to 18. Liczbę kolorów niestandardowych można ustawić na 1024, wówczas rozmiar palety w pionie zostanie zmieniony w celu dopasowania do liczby próbek.

## Używanie stylów ze składnikiem ColorPicker

W celu zmiany wyglądu składnika ColorPicker można ustawić kilka stylów. Na przykład, poniższa procedura powoduje zmianę liczby kolumn (`columnCount`) w ColorPicker na 12, zmianę wysokości (`swatchHeight`) i szerokości (`swatchWidth`) próbek koloru oraz zmianę wypełnienia dla pola tekstowego (`textPadding`) oraz tła (`backgroundPadding`).

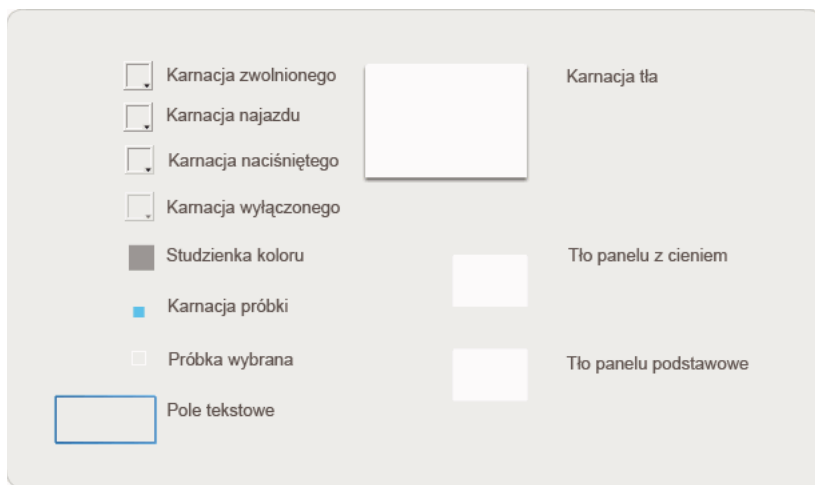
- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ColorPicker na stół montażowy i nadaj instancji nazwę **aCp**.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod:

```
aCp.setStyle("columnCount", 12);  
aCp.setStyle("swatchWidth", 8);  
aCp.setStyle("swatchHeight", 12);  
aCp.setStyle("swatchPadding", 2);  
aCp.setStyle("backgroundPadding", 3);  
aCp.setStyle("textPadding", 7);
```

- 4 Wybierz polecenie Sterowanie > Testuj film.
- 5 Kliknij składnik ColorPicker, aby go otworzyć i zobaczyć jak zastosowane ustawienia zmieniły jego wygląd.

## Używanie karnacji ze składnikiem ColorPicker

Składnik ColorPicker używa poniższych karnacji w celu zaprezentowania swoich graficznych stanów.



*Karnacje składnika ColorPicker*

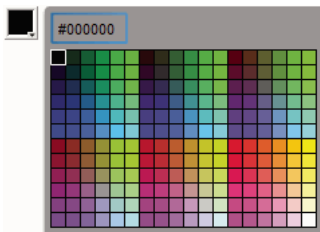
Aby zmienić kolor tła palety, można zmienić kolor karnacji Background.

- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ColorPicker na stół montażowy.
- 3 Kliknij go dwukrotnie, aby otworzyć jego paletę karnacji.
- 4 Klikaj dwukrotnie karnację Background do momentu jej zaznaczenia i pojawienia się wybieraka Koloru wypełnienia w Inspektorze właściwości.
- 5 Wybierz kolor #999999 za pomocą wybieraka Koloru wypełnienia w celu zastosowania go w karnacji Background.

**Dostosowywanie składników interfejsu użytkownika**

- 6 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 7 Wybierz polecenie Sterowanie > Testuj film.

Po kliknięciu składnika ColorPicker tło palety powinno uzyskać kolor szary tak, jak przedstawione to zostało na poniższej ilustracji.



Składnik ColorPicker z ciemnoszarą karnacją Background

## Dostosowywanie składnika ComboBox

Składnik ComboBox można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub odpowiednich właściwości klasy ComboBox takich, jak `height`, `width`, `scaleX` czy `scaleY`.

Rozmiar składnika ComboBox zostanie zmieniony w celu dopasowania do określonej szerokości i wysokości. Rozmiar listy zostanie zmieniony w celu dopasowania szerokości składnika, chyba że ustawiona została właściwość `dropdownWidth`.

Jeśli tekst jest zbyt długi, aby zostać dopasowanym w składniku ComboBox, tekst zostanie przycięty. Należy wówczas zmienić rozmiar składnika ComboBox i ustawić właściwość `dropdownWidth`, aby dopasować tekst.

## Używanie stylów ze składnikiem ComboBox

Właściwości stylu można ustawiać w celu zmiany wyglądu składnika ComboBox. Style określają wartości dla karnacji składnika, mechanizmu renderującego komórki, wypełnienia i szerokości przycisku. W poniższym przykładzie ustawiane są style `buttonWidth` i `textPadding`. Styl `buttonWidth` ustawia szerokość obszaru uderzenia przycisku i jego działanie odnosi skutek, jeśli pole ComboBox jest edytowalne — wówczas w celu otwarcia listy rozwijanej wystarczy nacisnąć przycisk. Styl `textPadding` określa wielkość odstępu między zewnętrzną krawędzią pola tekstowego a tekstem. Jest to użyteczne do pionowego wyśrodkowania tekstu w polu tekstowym, jeśli składnik ComboBox jest wydłużany. W przeciwnym wypadku tekst mógłby zostać wyświetlony u góry pola tekstowego.

- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ComboBox na stół montażowy i nadaj instancji nazwę `aCb`.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod:

**Dostosowywanie składników interfejsu użytkownika**

```
import fl.data.DataProvider;

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

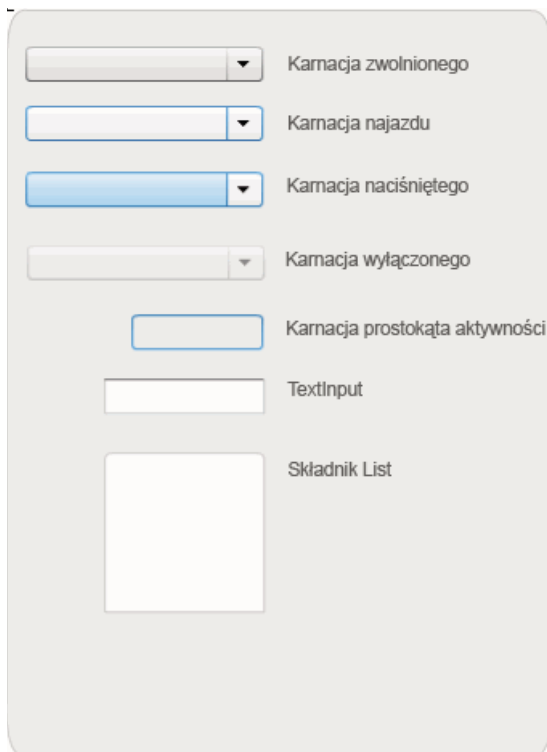
var items:Array = [
    {label:"San Francisco", data:"601 Townsend St."},
    {label:"San Jose", data:"345 Park Ave."},
    {label:"San Diego", data:"10590 West Ocean Air Drive, Suite 100"},
    {label:"Santa Rosa", data:"2235 Mercury Way, Suite 105"},
    {label:"San Luis Obispo", data:"3220 South Higuera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

**4** Wybierz polecenie Sterowanie > Testuj film.

Należy zauważyć, że obszar przycisku, który można kliknąć w celu otwarcia listy rozwijanej, stanowi jedynie wąski obszar po prawej stronie. Należy zwrócić również uwagę na to, że tekst w polu jest wyśrodkowany w pionie. Przykład można uruchomić z wyłączeniem dwóch instrukcji `setStyle()`, aby zobaczyć ich efekt.

**Używanie karnacji ze składnikiem ComboBox**

Składnik ComboBox używa poniższych karnacji w celu zaprezentowania swoich graficznych stanów:



*Karnacje składnika ComboBox*

Aby zmienić na stole montażowym kolor składnika w stanie nieaktywnym, można zmienić kolor karnacji Up.

**1** Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).

- 2 Przeciągnij składnik ComboBox na stół montażowy.
- 3 Kliknij go dwukrotnie, aby otworzyć jego paletę karnacji.
- 4 Klikaj dwukrotnie karnację Up do momentu jej zaznaczenia i otwarcia do edycji.
- 5 Ustaw element sterujący powiększenia na wartość 400%.
- 6 Klikaj na środku obszaru karnacji do momentu pojawienia się jej koloru w wybieraku Koloru wypełnienia w Inspektorze właściwości.
- 7 Wybierz kolor #33FF99 za pomocą wybieraka Koloru wypełnienia w celu zastosowania go w karnacji Up.
- 8 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 9 Wybierz polecenie Sterowanie > Testuj film.

Składnik ComboBox powinien zostać wyświetlony na stole montażowym tak, jak zostało to przedstawione na poniższej ilustracji.



Składnik ComboBox z niestandardowym kolorem dla karnacji Background

## Dostosowywanie składnika DataGrid

Składnik DataGrid można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub odpowiednich właściwości, takich jak `width`, `height`, `scaleX` czy `scaleY`. Jeśli brakuje poziomego paska przewijania, szerokości kolumn zostaną ustawione proporcjonalnie. Jeśli nastąpi korekcja rozmiaru kolumny (a zatem i komórki), tekst w komórkach może zostać przycięty.

### Używanie stylów ze składnikiem DataGrid

Właściwości stylu można ustawiać w celu zmiany wyglądu składnika DataGrid. Składnik DataGrid dziedziczy style po składniku List. (Patrz „[Używanie stylów ze składnikiem List](#)” na stronie 121).

### Ustawianie stylów dla pojedynczej kolumny

Obiekt DataGrid może zawierać wiele kolumn i możliwe jest określenie różnych mechanizmów renderujących komórki dla poszczególnych kolumn. Każda kolumna obiektu DataGrid ma odzwierciedlenie w postaci obiektu DataGridColumn, a klasa DataGridColumn zawiera właściwość `cellRenderer` umożliwiającą zdefiniowanie mechanizmu renderującego komórki danej kolumny.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik DataGrid do panelu Biblioteka.
- 3 Dodaj poniższy kod do panelu Operacje w klatce 1 na osi czasu. Kod ten tworzy składnik DataGrid z długim ciągiem znaków w trzeciej kolumnie. Na końcu we właściwości `cellRenderer` kolumny ustawiana jest nazwa mechanizmu renderującego komórki, który renderuje komórkę wielowierszową.

```
/* This is a simple cell renderer example. It invokes
the MultiLineCell cell renderer to display a multiple
line text field in one of a DataGrid's columns. */

import fl.controls.DataGrid;
import fl.controls.dataGridClasses.DataGridColumn;
import fl.data.DataProvider;
import fl.controls.ScrollPolicy;

// Create a new DataGrid component instance.
var aDg:DataGrid = new DataGrid();

var aLongString:String = "An example of a cell renderer class that displays a multiple line
TextField"
var myDP:Array = new Array();
myDP = [{firstName:"Winston", lastName:"Elstad", note:aLongString, item:100},
        {firstName:"Ric", lastName:"Dietrich", note:aLongString, item:101},
        {firstName:"Ewing", lastName:"Canepa", note:aLongString, item:102},
        {firstName:"Kevin", lastName:"Wade", note:aLongString, item:103},
        {firstName:"Kimberly", lastName:"Dietrich", note:aLongString, item:104},
        {firstName:"AJ", lastName:"Bilow", note:aLongString, item:105},
        {firstName:"Chuck", lastName:"Yushan", note:aLongString, item:106},
        {firstName:"John", lastName:"Roo", note:aLongString, item:107},
];

// Assign the data provider to the DataGrid to populate it.
// Note: This has to be done before applying the cellRenderers.
aDg.dataProvider = new DataProvider(myDP);

/* Set some basic grid properties.
Note: The data grid's row height should reflect
the number of lines you expect to show in the multiline cell.
The cell renderer will size to the row height.
About 40 for 2 lines or 60 for 3 lines.*/

aDg.columns = ["firstName", "lastName", "note", "item"];
aDg.setSize(430,190);
aDg.move(40,40);
aDg.rowHeight = 40;// Allows for 2 lines of text at default text size.
aDg.columns[0].width = 70;
aDg.columns[1].width = 70;
aDg.columns[2].width = 230;
aDg.columns[3].width = 60;
aDg.resizableColumns = true;
aDg.verticalScrollPolicy = ScrollPolicy.AUTO;
addChild(aDg);
// Assign cellRenderers.
var col3:DataGridColumn = new DataGridColumn();
col3 = aDg.getColumnAt(2);
col3.cellRenderer = MultiLineCell;
```

- 4 Zapisz plik FLA jako MultiLineGrid fla.
- 5 Utwórz nowy plik ActionScript.
- 6 Skopiuj następujący kod ActionScript do okna Skrypt:



```
package {  
  
    import fl.controls.listClasses.CellRenderer;  
  
    public class MultiLineCell extends CellRenderer  
    {  
  
        public function MultiLineCell()  
        {  
            textField.wordWrap = true;  
            textField.autoSize = "left";  
        }  
        override protected function drawLayout():void {  
            textField.width = this.width;  
            super.drawLayout();  
        }  
    }  
}
```

- 7 Zapisz plik ActionScript jako MultiLineCell.as w tym samym folderze, w którym zapisany został plik MultiLineGrid fla.
- 8 Powróć do aplikacji MultiLineGrid fla i wybierz polecenia Sterowanie > Testuj film.

Składnik DataGrid powinien wyglądać tak:

firstName	lastName	note	item
Winston	Elstad	An example of a cell renderer class that displays a multiple line TextField	100
Ric	Dietrich	An example of a cell renderer class that displays a multiple line TextField	101
Ewing	Canepa	An example of a cell renderer class that displays a multiple line TextField	102
Kevin	Wade	An example of a cell renderer class that displays a multiple line TextField	103

Składnik DataGrid dla aplikacji MultiLineGrid fla

## Ustawianie stylów nagłówka

Styl tekstu dla wiersza nagłówka można ustawić za pomocą stylu headerTextFormat. W poniższym przykładzie obiekt TextFormat użyty został w celu ustawienia stylu headerTextFormat do użytku z pochyloną czcionką Arial w kolorze czerwonym o rozmiarze 14.

- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij składnik DataGrid na stół montażowy i nadaj instancji nazwę aDg.
- 3 Otwórz panel Operacje, zaznacz klatkę nr 1 na głównej osi czasu i wprowadź następujący kod:

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

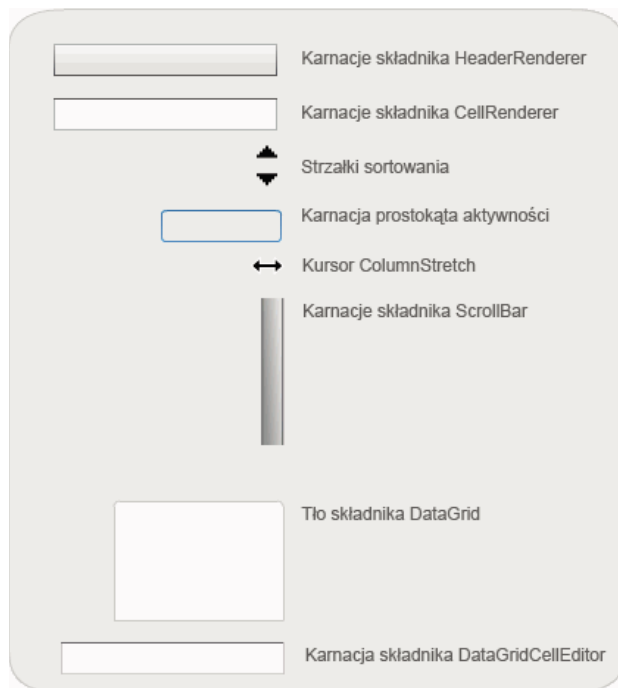
var myDP:Array = new Array();
myDP = [{FirstName:"Winston", LastName:"Elstad"},
        {FirstName:"Ric", LastName:"Dietrich"},
        {FirstName:"Ewing", LastName:"Canepa"},
        {FirstName:"Kevin", LastName:"Wade"},
        {FirstName:"Kimberly", LastName:"Dietrich"},
        {FirstName:"AJ", LastName:"Bilow"},
        {FirstName:"Chuck", LastName:"Yushan"},
        {FirstName:"John", LastName:"Roo"},
];

// Assign the data provider to the DataGrid to populate it.
// Note: This has to be done before applying the cellRenderers.
aDg.dataProvider = new DataProvider(myDP);
aDg.setSize(160,190);
aDg.move(40,40);
aDg.columns[0].width = 80;
aDg.columns[1].width = 80;
var tf:TextFormat = new TextFormat();
tf.size = 14;
tf.color = 0xff0000;
tf.italic = true;
tf.font = "Arial"
aDg.setStyle("headerTextFormat", tf);
```

- 4 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić aplikację.

## Używanie karnacji ze składnikiem DataGrid

Składnik DataGrid używa poniższych karnacji w celu zaprezentowania swoich graficznych stanów:



*Karnacje składnika DataGrid*

Karnacja CellRenderer jest karnacją używaną dla komórek treści składnika DataGrid, natomiast karnacja HeaderRenderer używana jest dla wiersza nagłówka. Poniższa procedura powoduje zmianę koloru tła wiersza nagłówka, ale ten sam proces może posłużyć do zmiany koloru tła komórek treści składnika DataGrid poprzez edycję karnacji CellRenderer.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik DataGrid na stół montażowy i nadaj instancji nazwę **aDg**.
- 3 Kliknij składnik dwukrotnie, aby otworzyć jego paletę karnacji.
- 4 Ustaw element sterujący powiększenia na wartość 400%, aby powiększyć ikony do edycji.
- 5 Kliknij dwukrotnie karnację HeaderRenderer, aby otworzyć paletę karnacji HeaderRenderer.
- 6 Kliknij dwukrotnie karnację Up\_Skin, aby otworzyć ją w trybie edycji symbolu i klikaj tło karnacji do momentu jej zaznaczenia i pojawienia się wybieraka Koloru wypełnienia w Inspektorze właściwości.
- 7 Wybierz kolor #00CC00 za pomocą wybieraka Koloru wypełnienia w celu zastosowania go do tła karnacji Up\_Skin HeaderRenderer.
- 8 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 9 Dodaj poniższy kod do panelu Operacje w klatce nr 1 na osi czasu, aby dodać dane do składnika DataGrid:

**Dostosowywanie składników interfejsu użytkownika**

```
import fl.data.DataProvider;

bldRosterGrid(aDg);
var aRoster:Array = new Array();
aRoster = [
    {Name:"Wilma Carter",Home: "Redlands, CA"},
    {Name:"Sue Pennypacker",Home: "Athens, GA"},
    {Name:"Jill Smithfield",Home: "Spokane, WA"},
    {Name:"Shirley Goth", Home: "Carson, NV"},
    {Name:"Jennifer Dunbar",Home: "Seaside, CA"}
];
aDg.dataProvider = new DataProvider(aRoster);
function bldRosterGrid(dg:DataGrid) {
    dg.setSize(400, 130);
    dg.columns = ["Name", "Home"];
    dg.move(50,50);
    dg.columns[0].width = 120;
    dg.columns[1].width = 120;
};
```

10 Wybierz polecenia Sterowanie > Testuj film, aby przetestować aplikację.

Składnik DataGrid powinien zostać wyświetlony w sposób przedstawiony na poniższej ilustracji z tłem dla wiersza nagłówka w kolorze zielonym.

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

Składnik DataGrid z niestandardowym tłem w wierszu nagłówka

## Dostosowywanie składnika Label

Składnik Label można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. Można również ustawić parametr `autoSize` tworzenia; ustawienie tego parametru nie powoduje zmiany obwiedni dla aktywnego podglądu, ale rozmiar składnika Label jest zmieniany. Zmiana rozmiaru składnika Label zależy od parametru `wordwrap`. Jeśli parametr ma wartość `true`, rozmiar w pionie składnika Label zmieniany jest w celu dopasowania do tekstu. Jeśli parametr ma wartość `false`, dla składnika Label zmieniony zostanie rozmiar w poziomie. W czasie wykonywania należy użyć metody `setSize()`. Więcej informacji zawierają opisy metody `Label.setSize()` i właściwości `Label.autoSize` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*. Zobacz także „[Tworzenie aplikacji ze składnikiem Label](#)” na stronie 65.

## Używanie stylów ze składnikiem Label

Właściwości stylu można ustawić w celu zmiany wyglądu instancji etykiety. Cały tekst instancji składnika Label musi współdzielić ten sam styl. Składnik Label zawiera styl `textFormat`, który ma te same atrybuty co obiekt `TextFormat` i umożliwia użytkownikowi ustawianie tych samych właściwości dla zawartości `Label.text` tak, jak dla normalnego składnika `TextField` programu Flash. W poniższym przykładzie kolor tekstu w etykiecie ustawiany jest na kolor czerwony.

- 1 Przeciągnij składnik Label z panelu Składniki na stół montażowy i nadaj instancji nazwę `a_label`.
- 2 Kliknij kartę Parametry i zastąp wartość właściwości tekstowej tekstem:

### **Pokoloruj mnie na czerwono**

- 3 Zaznacz klatkę nr 1 na głównej osi czasu, otwórz panel Operacje i wprowadź następujący kod:

```
/* Create a new TextFormat object, which allows you to set multiple text properties at a time. */  
  
var tf:TextFormat = new TextFormat();  
tf.color = 0xFF0000;  
/* Apply this specific text format (red text) to the Label instance. */  
a_label.setStyle("textFormat", tf);
```

- 4 Wybierz polecenie Sterowanie > Testuj film.

Więcej informacji na temat stylów składnika Label zawiera klasa Label opisana w [dokumentacji języka ActionScript 3.0 dla programu Flash Professional](#).

## Karnacje i składnik Label

Składnik Label nie zawiera żadnych elementów graficznych, do których można zastosować karnację.

## Dostosowywanie składnika List

Składnik List można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy używać metody `setSize()` i odpowiednich właściwości klasy List takich, jak `height`, `width`, `scaleX` czy `scaleY`.

Podczas zmiany rozmiaru listy w poziomie wiersze listy są zmniejszane, przycinając znajdujący się w nich tekst. W pionie w razie konieczności do listy dodawane do niej lub usuwane są z niej wiersze. W razie konieczności automatycznie umieszczane są także paski przewijania.

## Używanie stylów ze składnikiem List

Właściwości stylu można ustawiać w celu zmiany wyglądu składnika List. Style określają wartości dla karnacji składnika i wypełnienia podczas rysowania składnika.

Różne style karnacji umożliwiają określenie innych klas do użytku dla karnacji. Więcej informacji na temat używania stylów karnacji zawiera sekcja „[Informacje o karnacjach](#)” na stronie 105.

**Dostosowywanie składników interfejsu użytkownika**

Poniższa procedura ustawia wartość stylu `contentPadding` dla składnika `List`. Należy zauważyć, że wartość tego ustawienia pobierana jest z rozmiaru składnika `List` w celu otrzymania wypełnienia wokół zawartości, dlatego konieczne może być zwiększenie rozmiaru składnika `List`, aby zapobiec przycinaniu w nim tekstu.

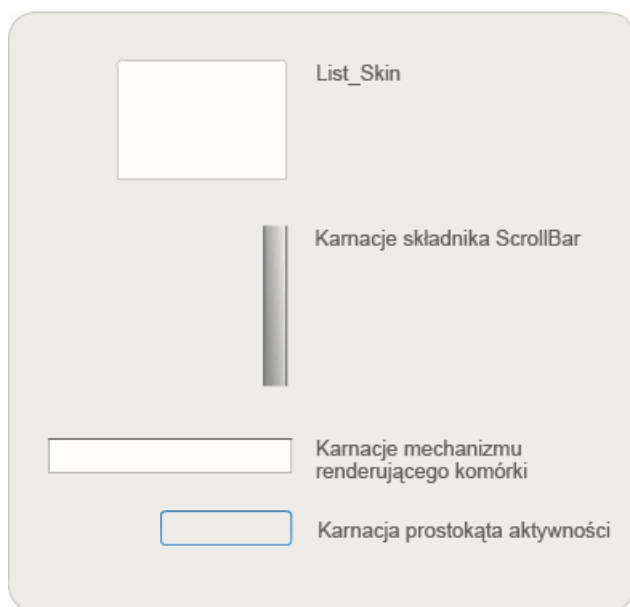
- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij składnik `List` z panelu Składniki na stół montażowy i nadaj nazwę instancji **aList**.
- 3 Zaznacz klatkę nr 1 na głównej osi czasu, otwórz panel Operacje i wprowadź następujący kod, który ustawia styl `contentPadding` i dodaje dane do składnika `List`:

```
aList.setStyle("contentPadding", 5);
aList.setSize(145, 200);
aList.addItem({label:"1956 Chevy (Cherry Red)", data:35000});
aList.addItem({label:"1966 Mustang (Classic)", data:27000});
aList.addItem({label:"1976 Volvo (Xcllnt Cond)", data:17000});
aList.rowCount = aList.length;
```

- 4 Wybierz polecenie Sterowanie > Testuj film.

**Używanie karnacji ze składnikiem List**

Składnik `List` używa poniższych karnacji w celu zaprezentowania swoich graficznych stanów:

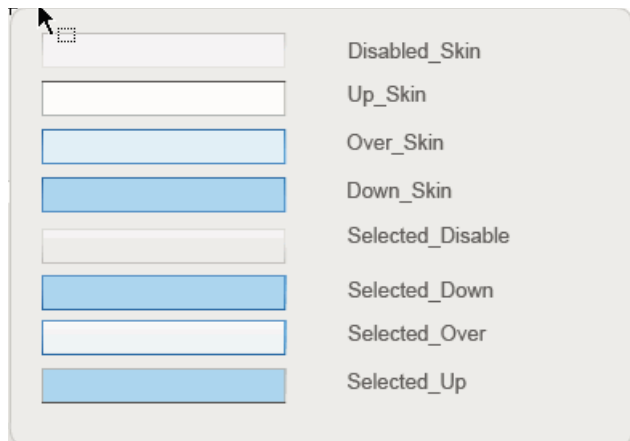


*Karnacje składnika List*

Więcej informacji na temat nakładania karnacji dla `ScrollBar` zawiera sekcja „[Dostosowywanie składnika UIScrollBar](#)” na stronie 136. Więcej informacji na temat nakładania karnacji dla karnacji `Focus Rect` zawiera sekcja „[Dostosowywanie składnika TextArea](#)” na stronie 131.

**Uwaga:** Zmiana karnacji `ScrollBar` w jednym składniku powoduje zmianę we wszystkich pozostałych składnikach, które korzystają ze `ScrollBar`.

Kliknij dwukrotnie karnację `Cell Renderer`, aby otworzyć drugą paletę karnacji dla innych stanów komórki składnika `List`.



Karnacje Cell Renderer składnika List

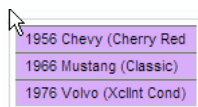
Wygląd komórek składnika List można zmienić, edytując karnacje. Poniższa procedura powoduje zmianę koloru karnacji Up w celu zmiany wyglądu składnika List w jego normalnym nieaktywnym stanie.

- 1 Utwórz nowy plik z dokumentem Flash (ActionScript 3.0).
- 2 Przeciągnij składnik List z panelu Składniki na stoł montażowy i nadaj nazwę instancji **aList**.
- 3 Kliknij składnik List dwukrotnie, aby otworzyć jego paletę karnacji.
- 4 Kliknij dwukrotnie karnację Cell Renderer, aby otworzyć paletę karnacji Cell Renderer.
- 5 Kliknij dwukrotnie karnację Up\_Skin, aby otworzyć ją do edycji.
- 6 Kliknij obszar wypełnienia karnacji, aby go zaznaczyć. W Inspektorze właściwości wyświetlony powinien zostać wybierak Koloru wypełnienia z bieżącym kolorem wypełnienia karnacji.
- 7 Wybierz kolor #CC66FF za pomocą wybieraka Koloru wypełnienia w celu zastosowania go do wypełnienia karnacji Up\_Skin.
- 8 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 9 Dodaj poniższy kod do panelu Operacje w klatce nr 1 na osi czasu, aby dodać dane do składnika List:

```
aList.setStyle("contentPadding", 5);
aList.setSize(145, 200);
aList.addItem({label:"1956 Chevy (Cherry Red)", data:35000});
aList.addItem({label:"1966 Mustang (Classic)", data:27000});
aList.addItem({label:"1976 Volvo (Xcllnt Cond)", data:17000});
aList.rowCount = aList.length;
```

- 10 Wybierz polecenie Sterowanie > Testuj film.

Składnik List powinien zostać wyświetlony tak, jak zostało to przedstawione na poniższej ilustracji:



Komórki składnika List z niestandardowym kolorem Up\_Skin

Obramowanie wyniku z ustawienia `contentPadding` stylu.

## Dostosowywanie składnika NumericStepper

Składnik NumericStepper można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub dowolnej z odpowiednich właściwości i metod klasy NumericStepper, takich jak `width`, `height`, `scaleX` czy `scaleY`.

Zmiana rozmiaru składnika NumericStepper nie powoduje zmiany szerokości przycisków strzałek „w górę” i „w dół”. Jeśli rozmiar obiektu stepper zostanie zmieniony na większy niż wysokość domyślna, zachowanie domyślne spowoduje, że przyciski strzałek zostaną przypięte na górze i na dole składnika. W przeciwnym wypadku sposób rysowania przycisków wskaże 9-plastrowe skalowanie. Przyciski strzałek zawsze wyświetlane są po prawej stronie pola tekstowego.

### Style i składnik NumericStepper

Właściwości stylu składnika NumericStepper można ustawiać w celu zmiany jego wyglądu. Style określają wartości dla karnacji składnika, wypełnienia oraz formatowania tekstu podczas rysowania składnika. Styl `textFormat` umożliwia zmianę rozmiaru i wyglądu wartości składnika NumericStepper. Różne style karnacji umożliwiają określenie innych klas do użytku dla karnacji. Więcej informacji na temat używania stylów karnacji zawiera sekcja „[Informacje o karnacjach](#)” na stronie 105.

W procedurze tej do zmiany wyglądu wartości wyświetlanej przez składnik NumericStepper używany jest styl `textFormat`.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik NumericStepper z panelu Składniki na stół montażowy i nadaj nazwę instancji **myNs**.
- 3 Dodaj poniższy kod do panelu Operacje w klatce nr 1 na głównej osi czasu:

```
var tf:TextFormat = new TextFormat();
myNs.setSize(100, 50);
tf.color = 0x0000CC;
tf.size = 24;
tf.font = "Arial";
tf.align = "center";
myNs.setStyle("textFormat", tf);
```

- 4 Wybierz polecenie Sterowanie > Testuj film.

### Karnacje i składnik NumericStepper

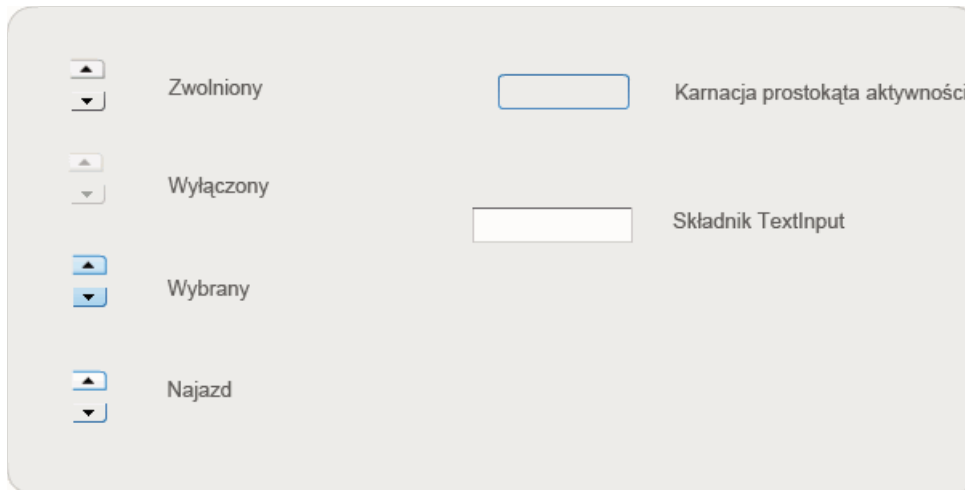
Składnik NumericStepper zawiera karnacje do reprezentowania stanów up, down, disabled i selected swoich przycisków.

Jeśli obiekt stepper jest włączony, przyciski w górę i w dół wyświetlają swoje stany over po przesunięciu na nie wskaźnika. Po naciśnięciu przycisków wyświetlany jest stan down. Przyciski powracają do stanu over po zwolnieniu przycisku myszy. Jeśli wskaźnik z naciśniętym przyciskiem myszy zostanie przesunięty poza przyciski, powrócą one do pierwotnego stanu.

Jeśli obiekt stepper jest wyłączony, stan disabled jest wyświetlany niezależnie od działania użytkownika.

Składnik NumericStepper zawiera następujące karnacje:





*Karnacje składnika NumericStepper*

- 1 Utwórz nowy plik FLA.
- 2 Przeciągnij składnik NumericStepper na stół montażowy.
- 3 Ustaw element sterujący powiększenia na wartość 400%, aby powiększyć obraz do edycji.
- 4 Klikaj dwukrotnie tło karnacji składnika TextInput na panelu karnacji do momentu przejścia do poziomu grupy i pojawienia się koloru tła w wybieraku Koloru wypełnienia w Inspektorze właściwości.
- 5 Za pomocą wybieraka Koloru wypełnienia w Inspektorze właściwości wybierz kolor #9999FF, aby zastosować go do tła karnacji składnika TextInput.
- 6 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 7 Kliknij dwukrotnie składnik NumericStepper, aby ponownie otworzyć panel karnacji.
- 8 Klikaj dwukrotnie tło przycisku strzałki do góry w grupie Up do momentu zaznaczenia tła i wyświetlenia jego koloru w selektorze Kolor wypełnienia w Inspektorze właściwości.
- 9 Wybierz kolor #9966FF w celu zastosowania go do tła przycisku strzałki do góry.
- 10 Powtórz kroki 8 i 9 dla strzałki w dół w grupie Up.
- 11 Wybierz polecenie Sterowanie > Testuj film.

Instancja NumericStepper powinna zostać wyświetlona tak, jak zostało to przedstawione na poniższej ilustracji:



## Dostosowywanie składnika ProgressBar

Składnik ProgressBar można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub odpowiednich właściwości klasy ProgressBar takich, jak `height`, `width`, `scaleX` czy `scaleY`.

Składnik ProgressBar zawiera trzy karnacje: karnacja track, karnacja bar i karnacja indeterminate. Do skalowania zasobów używa skalowania 9-plastrowego.

## Style i składnik ProgressBar

Właściwości stylu można ustawić w celu zmiany wyglądu instancji ProgressBar. Style składnika ProgressBar określają wartości dla jego karnacji i wypełnienia podczas rysowania składnika. W poniższym przykładzie powiększany jest rozmiar instancji ProgressBar oraz ustawiany jest jego styl barPadding.

- 1 Utwórz nowy plik FLA.
- 2 Przeciągnij składnik ProgressBar z panelu Składniki na stół montażowy i nadaj nazwę instancji **myPb**.
- 3 W panelu Operacje do klatki nr 1 na głównej osi czasu wprowadź następujący kod:

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

- 4 Wybierz polecenie Sterowanie > Testuj film.

Informacje na temat ustawiania stylów karnacji zawiera sekcja „[Informacje o karnacjach](#)” na stronie 105.

## Karnacje i składnik ProgressBar

Składnik ProgressBar używa karnacji do reprezentowania ścieżki paska postępu, paska zakończonego oraz paska nieokreślonego w sposób przedstawiony na poniższej ilustracji.



Karnacje składnika ProgressBar

Pasek jest umieszczany nad karnacją ścieżki z użyciem barPadding, aby określić ustawienie. Zasoby są skalowane za pomocą 9-plastrowego skalowania.

Pasek nieokreślony jest używany, gdy właściwość indeterminate instancji ProgressBar ma wartość true. Rozmiar karnacji jest zmieniany w pionie i poziomie w celu dopasowania do rozmiaru składnika ProgressBar.

W celu zmiany wyglądu składnika ProgressBar można dokonać edycji karnacji. Na przykład w poniższym przykładzie zmieniany jest kolor paska nieokreślonego.

- 1 Utwórz nowy plik FLA.
- 2 Przeciągnij składnik ProgressBar na stół montażowy i kliknij go dwukrotnie, aby otworzyć panel ikon karnacji.
- 3 Kliknij dwukrotnie karnację paska nieokreślonego.
- 4 Ustaw element sterujący powiększenia na wartość 400%, aby powiększyć ikonę do edycji.
- 5 Kliknij dwukrotnie jeden z diagonalnych pasków, a następnie przytrzymując klawisz Shift kliknij pozostałe. Biejący kolor pojawi się w wybieraku Koloru wypełnienia w Inspektorze właściwości.

- 6 Kliknij wybierak Koloru wypełnienia w Inspektorze właściwości, aby go otworzyć i wybrać kolor #00CC00 w celu zastosowania do zaznaczonych, diagonalnych pasków.
- 7 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 8 Wybierz polecenie Sterowanie > Testuj film.  
Składnik ProgressBar powinien zostać wyświetlony tak, jak zostało to przedstawione na poniższej ilustracji.



## Dostosowywanie składnika RadioButton

Składnik RadioButton można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()`.

Obwiednia składnika RadioButton jest niewidoczna i wskazuje także obszar klikania składnika. Jeśli zwiększony zostanie rozmiar składnika, zwiększony zostanie również rozmiar obszaru klikania.

Jeśli obwiednia składnika jest zbyt mała, aby dopasować etykietę składnika, zostanie ona przycięta.

## Używanie stylów ze składnikiem RadioButton

Właściwości stylu można ustawić w celu zmiany wyglądu składnika RadioButton. Właściwości stylu składnika RadioButton określają wartości dla jego karnacji, ikon, formatowania tekstu i wypełnienia podczas rysowania składnika. Style składnika RadioButton określają wartości dla jego karnacji i wypełnienia dla warstwy podczas rysowania składnika.

W poniższym przykładzie ze składnika CheckBox pobierany jest styl `textFormat`, który stosowany jest następnie do składnika RadioButton tak, aby styl ich etykiet był identyczny.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik CheckBox na stół montażowy i nadaj instancji nazwę **myCh** w Inspektorze właściwości.
- 3 Przeciągnij składnik RadioButton na stół montażowy i nadaj instancji nazwę **myRb** w Inspektorze właściwości.
- 4 Dodaj poniższy kod do panelu Operacje w klatce 1 na osi czasu.

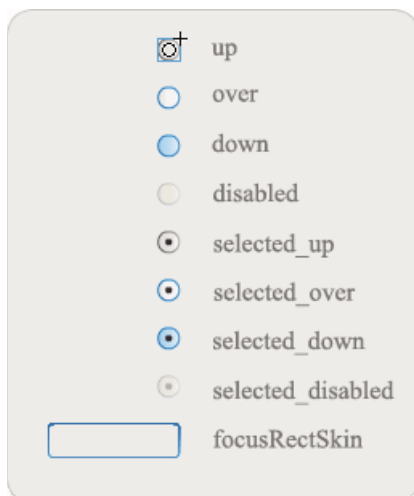
```
var tf:TextFormat = new TextFormat();
tf.color = 0x00FF00;
tf.font = "Georgia";
tf.size = 18;
myCh.setStyle("textFormat", tf);
myRb.setStyle("textFormat", myCh.getStyle("textFormat"));
```

Kod ten ustawia styl `textFormat` dla składnika CheckBox, a następnie stosuje go dla składnika RadioButton przez wywołanie w CheckBox metody `getStyle()`.

- 5 Wybierz polecenie Sterowanie > Testuj film.

## Karnacje i składnik RadioButton

Składnik RadioButton zawiera poniższe karnacje, które można edytować w celu zmiany jego wyglądu:



Karnacje składnika RadioButton

Jeśli składnik RadioButton jest włączony, ale nie jest zaznaczony, po przesunięciu na niego wskaźnika wyświetlana jest karnacja over. W momencie kliknięcia składnika RadioButton składnik jest aktywowany do wprowadzania i wyświetlana jest jego karnacja selected\_down. W momencie zwolnienia przycisku myszy dla składnika RadioButton zostanie wyświetlona karnacja selected\_up. Jeśli wskaźnik z wciśniętym przyciskiem myszy zostanie przesunięty poza obszar klikania składnika RadioButton, wyświetlona zostanie karnacja up.

Jeśli składnik RadioButton jest wyłączony, stan disabled jest wyświetlany niezależnie od działania użytkownika.

W poniższym przykładzie karnacja selected\_up, która wskazuje wybrany stan, jest zastępowana.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik RadioButton na stół montażowy i kliknij go dwukrotnie, aby otworzyć paletę karnacji.
- 3 Ustaw element sterujący powiększenia na wartość 800%, aby powiększyć ikonę do edycji.
- 4 Kliknij dwukrotnie karnację selected\_up, aby ją zaznaczyć i naciśnij klawisz Delete celem jej usunięcia.
- 5 Wybierz narzędzie Prostokąt w panelu Narzędzia.
- 6 W Inspektorze właściwości ustaw kolor linii na czerwony (#FF0000) i Kolor wypełnienia na czarny (#000000).
- 7 Kliknij i przeciągnij wskaźnik, aby narysować prostokąt, zaczynając od krzyżyka, który wskazuje punkt odniesienia symbolu (nazywany także *punktem początkowym* lub *punktem zerowym*).
- 8 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 9 Wybierz polecenie Sterowanie > Testuj film.
- 10 Kliknij składnik RadioButton, aby go zaznaczyć.

Składnik RadioButton w stanie selected powinien zostać wyświetlony podobnie, jak składnik na poniższej ilustracji.



## Dostosowywanie składnika ScrollPane

Składnik ScrollPane można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub dowolnej z odpowiednich właściwości lub metod klasy ScrollPane, takich jak `height`, `width`, `scaleX` czy `scaleY`.

Składnik ScrollPane zawiera następujące właściwości graficzne:

- Punkt odniesienia (nazywany także *punktem początkowym* lub *punktem zerowym*) swojej zawartości znajdujący się w lewym górnym rogu panelu.
- Gdy poziomy pasek przewijania jest wyłączony, od góry do dołu wzdłuż prawej strony panelu przewijania wyświetlany jest pionowy pasek przewijania. Gdy pionowy pasek przewijania jest wyłączony, od strony lewej do prawej u dołu panelu przewijania wyświetlany jest poziomy pasek przewijania. Możliwe jest także wyłączenie obu pasków.
- Jeśli panel przewijania jest zbyt mały, treść może nie zostać wyświetlona poprawnie.
- Gdy zmieniany jest rozmiar panelu przewijania, rozszerzane lub zwężane są ścieżka i uchwyt przewijania oraz ich obszar klikania. Rozmiar przycisków pozostaje ten sam.

## Używanie stylów ze składnikiem ScrollPane

Właściwości stylu składnika ScrollPane określają wartości dla jego karnacji i wypełnienia dla warstwy podczas rysowania składnika. Różne style karnacji umożliwiają określenie innych klas do użytku dla karnacji składnika. Więcej informacji na temat używania stylów karnacji zawiera sekcja „[Informacje o karnacjach](#)” na stronie 105.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik ScrollPane na stół montażowy i nadaj instancji nazwę `mySp`.
- 3 Kliknij kartę Parametry w Inspektorze właściwości i wprowadź dla parametru `source` następującą wartość:  
`http://www.helpexamples.com/flash/images/image1.jpg`

- 4 W panelu Operacje do klatki nr 1 na głównej osi czasu dodaj następujący kod:

```
mySp.setStyle("contentPadding", 5);
```

Należy zauważyć, że wypełnienie stosowane jest między krawędzią składnika i jego treścią, na zewnątrz pasków przewijania.

- 5 Wybierz polecenie Sterowanie > Testuj film.

## Karnacje i składnik ScrollPane

Składnik ScrollPane używa krawędzi i pasków przewijania do zasobów przewijania. Informacje na temat używania karnacji w paskach przewijania zawiera sekcja „[Używanie karnacji ze składnikiem UIScrollBar](#)” na stronie 137.

## Dostosowywanie składnika Slider

Składnik Slider można przekształcać w poziomie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub dowolnej z odpowiednich właściwości klasy Slider, takich jak `width` i `scaleX`.

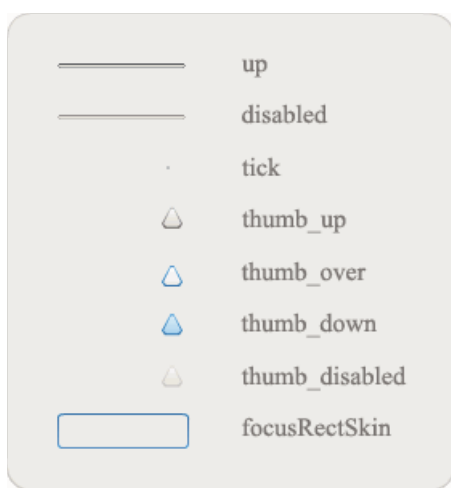
Suwak można tylko wydłużyć. Nie można go rozszerzyć. Program Flash zignoruje właściwość `height` oraz parametr `height` metody `setSize()`. Można jednak utworzyć suwak pionowy i wydłużyć go w tej orientacji.

## Style i składnik Slider

Style składnika Slider określają jedynie klasy dla karnacji oraz wartość dla parametru `FocusRectPadding`, który określa liczbę pikseli używanych dla wypełnienia między obwiednią składnika, a jego zewnętrznymi krawędziami. Więcej informacji o używaniu stylów karnacji zawiera sekcja „[Informacje o karnacjach](#)” na stronie 105.

## Karnacje i składnik Slider

Składnik Slider używa poniższych karnacji, które można edytować w celu zmiany jego wyglądu.



Karnacje składnika Slider

W poniższym przykładzie w celu zmiany koloru ścieżki up na niebieski dokonywana jest jej edycja.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik Slider z panelu Składniki na stół montażowy.
- 3 Kliknij dwukrotnie składnik Slider, aby otworzyć panel karnacji.
- 4 Kliknij dwukrotnie znak odniesienia ścieżki up, aby otworzyć ją w trybie edycji symbolu.
- 5 Ustaw element sterujący powiększenia na wartość 800%, aby powiększyć ikonę do edycji. Należy zauważyć, że ścieżka składnika Slider składa się z trzech pasków.
- 6 Kliknij górny pasek, aby go zaznaczyć. Po zaznaczeniu jego kolor wyświetlony zostanie w wybieraku Koloru wypełnienia w Inspektorze właściwości.
- 7 Za pomocą wybieraka Koloru wypełnienia w Inspektorze właściwości wybierz kolor #000066, aby zastosować go do górnego paska ścieżki składnika Slider.
- 8 Kliknij środkowy pasek ścieżki składnika Slider, aby go zaznaczyć. Po zaznaczeniu jego kolor wyświetlany jest w wybieraku Koloru wypełnienia w Inspektorze właściwości.
- 9 Za pomocą wybieraka Koloru wypełnienia w Inspektorze właściwości wybierz kolor #0066FF, aby zastosować go do środkowego paska ścieżki składnika Slider.
- 10 Kliknij dolny pasek ścieżki składnika Slider, aby go zaznaczyć. Po zaznaczeniu jego kolor wyświetlany jest w wybieraku Koloru wypełnienia w Inspektorze właściwości.

- 11 Za pomocą wybieraka Koloru wypełnienia w Inspektorze właściwości wybierz kolor #00CCFF, aby zastosować go do dolnego paska ścieżki składnika Slider.
- 12 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 13 Wybierz polecenie Sterowanie > Testuj film.

Składnik Slider powinien zostać wyświetlony tak, jak zostało to przedstawione na poniższej ilustracji.



## Dostosowywanie składnika TextArea

Składnik TextArea można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub odpowiednich właściwości klasy TextArea takich, jak `height`, `width`, `scaleX` czy `scaleY`.

Podczas zmiany rozmiaru składnika TextArea rozmiar krawędzi zmieniany jest na wielkość nowej obwiedni. W razie potrzeby na dolnej i prawej krawędzi umieszczane są paski przewijania. Rozmiar obszaru tekstowego zmieniony zostanie wówczas w istniejącym obszarze; w składniku TextArea nie ma elementów o stałym rozmiarze. Jeśli szerokość składnika TextArea jest zbyt wąska do wyświetlenia tekstu, zostanie on przycięty.

### Style i składnik TextArea

Style składnika TextArea określają wartości dla jego karnacji, wypełnienia oraz formatowania tekstu podczas rysowania składnika. Styl tekstu wyświetlanego w składniku TextArea kontrolują style `textFormat` i `disabledTextFormat`. Więcej informacji o właściwościach stylu karnacji zawiera sekcja „[Używanie karnacji ze składnikiem TextArea](#)” na stronie 131.

W poniższym przykładzie styl `disabledTextFormat` ustawiany jest w celu zmiany wyglądu tekstu, gdy składnik TextArea jest wyłączony, a dla włączonego składnika TextArea w tym samym procesie stosowany jest styl `textFormat`.

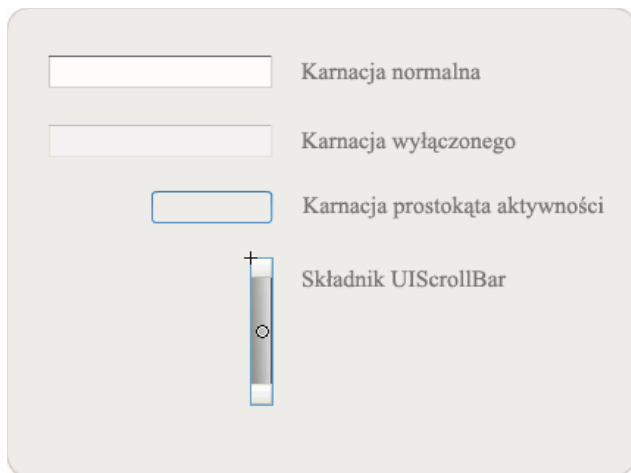
- 1 Utwórz nowy plik Flash.
- 2 Przeciągnij składnik TextArea na stół montażowy i nadaj instancji nazwę **myTa**.
- 3 Dodaj poniższy kod do panelu Operacje w klatce nr 1 na głównej osi czasu.

```
var tf:TextFormat = new TextFormat();
tf.color = 0xCC99FF;
tf.font = "Arial Narrow";
tf.size = 24;
myTa.setStyle("disabledTextFormat", tf);
myTa.text = "Hello World";
myTa.setSize(120, 50);
myTa.move(200, 50);
myTa.enabled = false;
```

- 4 Wybierz polecenie Sterowanie > Testuj film.

### Używanie karnacji ze składnikiem TextArea

Składnik TextArea używa poniższych karnacji, które można edytować w celu zmiany jego wyglądu.



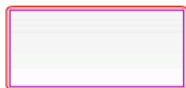
Karnacje składnika TextArea

**Uwaga:** Zmiana karnacji ScrollBar w jednym składniku powoduje zmianę we wszystkich pozostałych składnikach, które korzystają ze ScrollBar.

Poniższa procedura powoduje zmianę kolorów krawędzi karnacji Focus Rect, która jest wyświetlana w momencie aktywowania składnika TextArea, oraz karnacji Normal.

- 1 Utwórz nowy plik Flash.
- 2 Przeciągnij składnik TextArea na stół montażowy i kliknij go dwukrotnie, aby otworzyć panel ikon karnacji.
- 3 Kliknij dwukrotnie karnację Focus Rect.
- 4 Kliknij krawędź karnacji Focus Rect, aby ją zaznaczyć. Po jej zaznaczeniu bieżący kolor wyświetlany jest w wybieraku Koloru wypełnienia w Inspektorze właściwości.
- 5 Kliknij wybierak Koloru wypełnienia w Inspektorze właściwości, aby go otworzyć i wybrać kolor #CC0000 w celu zastosowania do krawędzi.
- 6 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 7 Kliknij dwukrotnie składnik TextArea, aby otworzyć panel ikon karnacji.
- 8 Kliknij dwukrotnie karnację Normal.
- 9 Zaznacz osobno każdą z krawędzi obramowania karnacji Normal i ustaw jej kolor na #990099.
- 10 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 11 Wybierz polecenie Sterowanie > Testuj film.

Po zaznaczeniu składnika TextArea w celu rozpoczęcia wprowadzania tekstu jego obramowanie powinno zostać wyświetlone tak, jak zostało to przedstawione na poniższej ilustracji:



Krawędzią zewnętrzną jest karnacja Focus Rect, a krawędzią wewnętrzną karnacja Normal.



Informacje o edytowaniu składnika UIScrollBar zawiera sekcja „[Dostosowywanie składnika UIScrollBar](#)” na stronie 136.

## Dostosowywanie składnika TextInput

Rozmiar instancji TextInput można zmieniać podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształć swobodnie lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy używać metody `setSize()` lub odpowiednich właściwości klasy TextInput takich, jak `height`, `width`, `scaleX` czy `scaleY`.

Podczas zmiany rozmiaru składnika TextInput rozmiar krawędzi zmieniany jest na wielkość nowej obwiedni. W składniku TextInput nie są używane paski przewijania, ale punkt wstawiania zostanie automatycznie przewinięty, gdy użytkownik rozpocznie pracę z tekstem. Rozmiar pola tekstowego zmieniony zostanie wówczas w istniejącym obszarze; w składniku TextInput nie ma elementów o stałym rozmiarze. Jeśli składnik TextInput jest zbyt mały do wyświetlenia tekstu, zostanie on przycięty.

### Style i składnik TextInput

Style składnika TextInput określają wartości dla jego karnacji, wypełnienia oraz formatowania tekstu podczas rysowania składnika. Styl tekstu wyświetlanego w składniku kontrolują style `textFormat` i `disabledTextFormat`. Więcej informacji o właściwościach stylu karnacji zawiera sekcja „[Karnacje i składnik TextInput](#)” na stronie 133.

W poniższym przykładzie ustawiany jest styl `textFormat` w celu ustawienia czcionki, rozmiaru i koloru tekstu dla składnika TextInput. Ten sam proces stosowany jest w celu ustawienia stylu `disabledTextFormat`, który znajduje zastosowanie dla składnika, gdy jest on wyłączony.

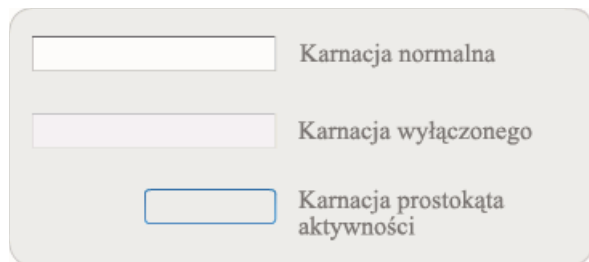
- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik TextInput na stół montażowy i nadaj instancji nazwę **myTi**.
- 3 Dodaj poniższy kod do panelu Operacje w klatce nr 1 na głównej osi czasu.

```
var tf:TextFormat = new TextFormat();
tf.color = 0x0000FF;
tf.font = "Verdana";
tf.size = 30;
tf.align = "center";
tf.italic = true;
myTi.setStyle("textFormat", tf);
myTi.text = "Enter your text here";
myTi.setSize(350, 50);
myTi.move(100, 50);
```

- 4 Wybierz polecenie Sterowanie > Testuj film.

### Karnacje i składnik TextInput

Składnik TextInput używa poniższych karnacji, które można edytować w celu zmiany jego wyglądu:



Napis składnika TextInput

Poniższa procedura powoduje zmianę koloru krawędzi i tła składnika TextInput:

- 1 Utwórz nowy plik Flash.
- 2 Przeciągnij składnik TextInput na stół montażowy i kliknij go dwukrotnie, aby otworzyć panel karnacji.
- 3 Kliknij dwukrotnie karnację Normal.
- 4 Ustaw element sterujący powiększenia na wartość 800%, aby powiększyć ikonę do edycji.
- 5 Zaznacz osobno każdą z krawędzi obramowania karnacji Normal i ustaw jej kolor na #993399 w celu zastosowania.
- 6 Klikaj dwukrotnie tło do momentu pojawienia się jego koloru w wybieraku Koloru wypełnienia w Inspektorze właściwości. Wybierz kolor #99CCCC w celu zastosowania go do tła.
- 7 Kliknij przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym, aby powrócić do trybu edycji dokumentu.
- 8 Wybierz polecenie Sterowanie > Testuj film.

Składnik TextInput powinien zostać wyświetlony tak, jak zostało to przedstawione na poniższej ilustracji:



## Dostosowywanie składnika TileList

Składnik TileList można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształć swobodnie lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub odpowiednich właściwości, takich jak `width`, `height`, `columnCount`, `rowCount`, `scaleX` czy `scaleY`. Składnik ScrollBar zawarty w składniku TileList, jest skalowany razem z polem listy.

### Style i składnik TileList

Style składnika TileList określają wartości dla jego karnacji, wypełnienia oraz formatowania tekstu podczas rysowania składnika. Styl tekstu wyświetlanego w składniku kontrolują style `textFormat` i `disabledTextFormat`. Więcej informacji o stylach karnacji zawiera sekcja „[Używanie karnacji ze składnikiem TileList](#)” na stronie 135.

W poniższym przykładzie wywoływana jest metoda `setRendererStyle()`, która używa stylu `textFormat` w celu ustawienia czcionki, rozmiaru, koloru i atrybutów tekstu dla etykiet wyświetlanych w instancji TileList. Ten sam proces stosowany jest w celu ustawienia stylu `disabledTextFormat`, który znajduje zastosowanie, gdy właściwość `enabled` ma wartość `false`.

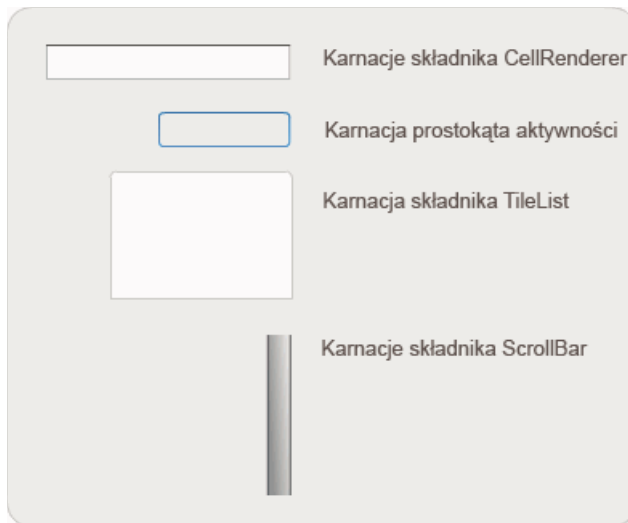
- 1 Utwórz nowy dokument Flash (ActionScript 3.0).

- 2 Przeciągnij składnik TileList na stół montażowy i nadaj instancji nazwę **myTl**.
- 3 Dodaj poniższy kod do panelu Operacje w klatce 1 na osi czasu.

```
myTl.setSize(100, 100);  
myTl.addItem({label:"#1"});  
myTl.addItem({label:"#2"});  
myTl.addItem({label:"#3"});  
myTl.addItem({label:"#4"});  
var tf:TextFormat = new TextFormat();  
tf.font = "Arial";  
tf.color = 0x00FF00;  
tf.size = 16;  
tf.italic = true;  
tf.bold = true;  
tf.underline = true;  
tf.align = "center";  
myTl.setRendererStyle("textFormat", tf);
```

## Używanie karnacji ze składnikiem TileList

Składnik TileList zawiera karnacje TileList, CellRenderer oraz ScrollBar. W celu zmiany wyglądu składnika TileList można dokonać edycji karnacji:



Karnacje składnika TileList

**Uwaga:** Zmiana karnacji ScrollBar w jednym składniku powoduje zmianę we wszystkich pozostałych składnikach, które korzystają ze ScrollBar.

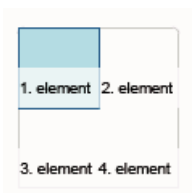
Poniższa procedura zmienia kolor karnacji CellRenderer Selected\_Up składnika TileList.

- 1 Utwórz dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik TileList na stół montażowy i kliknij go dwukrotnie, aby otworzyć panel karnacji.
- 3 Kliknij dwukrotnie karnację CellRenderer; kliknij dwukrotnie karnację Selected\_Up, a następnie kliknij prostokątne tło.
- 4 Wybierz kolor #99FFFF za pomocą wybieraka Koloru wypełnienia w Inspektorze właściwości, aby zastosować go w karnacji Selected\_Up.

**Dostosowywanie składników interfejsu użytkownika**

- 5 Klikaj przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym do momentu powrotu do trybu edycji dokumentu.
- 6 Na karcie Parametry Inspektora właściwości kliknij dwukrotnie drugą kolumnę wiersza dataProvider, aby otworzyć okno dialogowe Wartości. Dodaj elementy z następującymi etykietami: 1st item, 2nd item, 3rd item, 4th item.
- 7 Wybierz polecenie Sterowanie > Testuj film.
- 8 Kliknij jedną z komórek w składniku TileList, aby ją zaznaczyć, a następnie przesuń mysz poza zaznaczoną komórkę.

Zaznaczona komórka powinna zostać wyświetlona tak, jak zostało to przedstawione na poniższej ilustracji:



Składnik TileList ze zmodyfikowanym kolorem karnacji Selected\_Up

## Dostosowywanie składnika UILoader

Składnik UILoader można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub odpowiednich właściwości takich, jak `width`, `height`, `scaleX` czy `scaleY`.

Dopasowanie rozmiaru składnika UILoader sterowane jest za pomocą właściwości `scaleContent`. Jeśli właściwość `scaleContent` ma wartość `true`, treść jest skalowana w celu dopasowania do obwiedni modułu loader (i jest ponownie skalowana w momencie wywołania metody `setSize()`). Jeśli właściwość `scaleContent` ma wartość `false`, rozmiar składnika wiązany jest z rozmiarem treści, wówczas użycie metody `setSize()` i właściwości służących do dopasowania rozmiaru nie przyniesie żadnego efektu.

Składnik UILoader nie zawiera elementów interfejsu użytkownika, do których można stosować style lub karnacje.

## Dostosowywanie składnika UIScrollBar

Składnik UIScrollBar można przekształcać w poziomie i w pionie podczas tworzenia i w czasie wykonywania. Jednak pionowy składnik UIScrollBar nie zezwala na modyfikację szerokości, a poziomy UIScrollBar na modyfikację wysokości. W tym celu podczas tworzenia należy wybrać na stole montażowym składnik i użyć narzędzia Przekształcenie swobodne lub dowolnego z poleceń Modyfikuj > Przekształć. W czasie wykonywania należy użyć metody `setSize()` lub dowolnej z odpowiednich właściwości klasy UIScrollBar takich, jak `width`, `height`, `scaleX` czy `scaleY`.

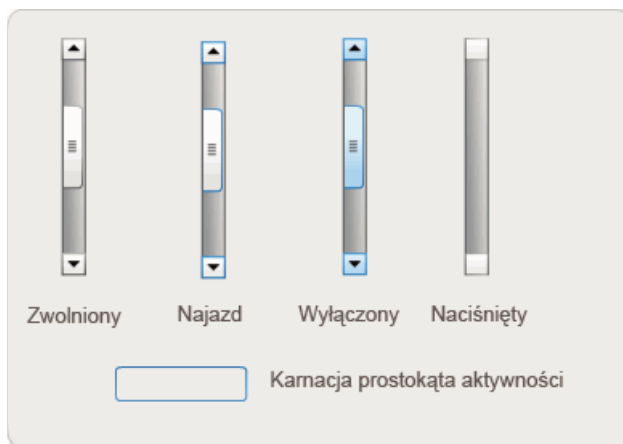
**Uwaga:** Użycie metody `setSize()` umożliwia zmianę tylko szerokości poziomego paska przewijania lub zmianę wysokości paska pionowego. W czasie tworzenia możliwe jest ustawienie wysokości poziomego paska przewijania lub szerokości paska pionowego, jednak w momencie publikowania filmu wartości te zostaną wyzerowane. Zmienić można jedynie wymiar odnoszący się do długości paska przewijania.

## Używanie stylów ze składnikiem UIScrollBar

Style składnika UIScrollBar określają jedynie klasy dla karnacji oraz wartość dla parametru FocusRectPadding, który określa liczbę pikseli używanych dla wypełnienia między obwiednią składnika, a jego zewnętrznymi krawędziami. Więcej informacji o używaniu stylów karnacji zawiera sekcja „[Informacje o karnacjach](#)” na stronie 105.

## Używanie karnacji ze składnikiem UIScrollBar

Składnik UIScrollBar używa następujących karnacji.



Karnacje składnika UIScrollBar

Oba paski przewijania (poziomy i pionowy) używają tych samych karnacji; podczas wyświetlania poziomego paska przewijania karnacje są odpowiednio obracane przez składnik UIScrollBar.

**Uwaga:** Zmiana karnacji ScrollBar w jednym składniku powoduje zmianę we wszystkich pozostałych składnikach, które korzystają ze ScrollBar.

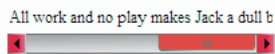
W poniższym przykładzie zademonstrowany został sposób, w jaki zmieniany jest kolor uchwytu składnika UIScrollBar oraz przyciski strzałek.

- 1 Utwórz nowy dokument Flash (ActionScript 3.0).
- 2 Przeciągnij składnik UIScrollBar na stół montażowy i nadaj instancji nazwę **mySb**. Na karcie Parametry ustaw kierunek na poziomy.
- 3 Kliknij dwukrotnie pasek przewijania, aby otworzyć panel karnacji.
- 4 Kliknij karnację Up, aby ją zaznaczyć.
- 5 Ustaw element sterujący powiększenia na wartość 400%, aby powiększyć ikonę do edycji.
- 6 Klikaj dwukrotnie tło przycisku strzałki w prawo (lub do góry dla pionowego paska przewijania) do momentu zaznaczenia tła i wyświetlenia jego koloru w wybieraku Koloru wypełnienia w Inspektorze właściwości.
- 7 Wybierz kolor #CC0033 w celu zastosowania go do tła przycisku.
- 8 Klikaj przycisk Wstecz po lewej stronie paska edycji nad stołem montażowym do momentu powrotu do trybu edycji dokumentu.
- 9 Powtórz kroki 6, 7 i 8 dla elementu uchwytu i strzałki w lewo (lub w dół dla pionowego paska przewijania).
- 10 Dodaj poniższy kod do panelu Operacje w klatce nr 1 na osi czasu, aby dołączyć pasek przewijania do składnika TextField.

```
var tf:TextField = new TextField();
addChild(tf);
tf.x = 150;
tf.y = 100;
mySb.width = tf.width = 200;
tf.height = 22;
tf.text = "All work and no play makes Jack a dull boy. All work and no play makes Jack a
dull boy. All . . .";
mySb.y = tf.y + tf.height;
mySb.x = tf.x + tf.width;x
mySb.scrollTarget = tf;
```

#### 11 Wybierz polecenie Sterowanie > Testuj film.

Składnik UIScrollBar powinien zostać wyświetlony tak, jak zostało to przedstawione na poniższej ilustracji.



*Poziomy składnik ScrollBar z uchwytem i strzałkami w lewo i w prawo w kolorze czerwonym*

# Rozdział 6: Korzystanie ze składnika FLVPlayback

Składnik FLVPlayback umożliwia łatwe dołączenie odtwarzacza wideo do aplikacji Adobe Flash CS5 Professional w celu odtwarzania plików wideo pobieranych progresywnie przez HTTP lub w celu strumieniowego przesyłania plików wideo z serwera Adobe Macromedia Flash Media Server albo z usługi FVSS (Flash Video Streaming Service).

W oprogramowaniu Adobe Flash Player 9, aktualizacji 3 (wersja 9.0.115.0 lub późniejsze), wprowadzono istotne udoskonalenia funkcji odtwarzania wideo w programie Flash Player. Ta aktualizacja obejmuje zmiany składnika FLVPlayback, który korzysta z urządzeń wideo w systemie użytkownika końcowego w celu zwiększenia wydajności odtwarzania sygnału wideo. Zmiany składnika FLVPlayback zwiększają również wierność plików wideo wyświetlanych w trybie pełnoekranowym.

Ponadto w programie Flash Player 9, w aktualizacji 3, udoskonalono funkcje składnika FLVPlayback poprzez dodanie funkcji obsługi formatów MPEG-4 HD, które korzystają z kodowania w branżowym standardzie H.264. Do tych formatów należą: MP4, M4A, MOV, MP4V, 3GP i 3G2.

**Uwaga:** *Zabezpieczone pliki MP4 — np. pobrane z serwisu Apple® iTunes® lub zaszyfrowane cyfrowo przez oprogramowanie FairPlay® — nie są obsługiwane.*

Łatwy w użyciu składnik FLVPlayback ma następujące cechy i zalety:

- Może być szybko przeciągany na stół montażowy, a następnie szybko implementowany
- Obsługuje format pełnoekranowy
- Udostępnia kolekcję wstępnie zaprojektowanych *karnacji*, które umożliwiają dostosowanie wyglądu elementów sterujących odtwarzaniem w składniku
- Umożliwia wybieranie koloru i wartości alfa dla wstępnie zaprojektowanych karnacji
- Umożliwia zaawansowanym użytkownikom tworzenie własnych karnacji
- Umożliwia podgląd aktywny podczas tworzenia
- Udostępnia właściwości układu, dzięki którym możliwe jest centrowanie pliku wideo w przypadku zmian wielkości
- Umożliwia rozpoczęcie odtwarzania po pobraniu dostatecznej ilości pliku wideo pobieranego progresywnie
- Udostępnia punkty sygnalizacji, które umożliwiają synchronizację wideo z tekstem, grafiką i animacją
- Zachowuje plik SWF w odpowiednich rozmiarach

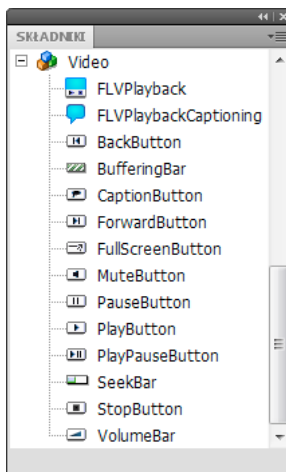
## Użycie składnika FLVPlayback

Korzystanie ze składnika FLVPlayback obejmuje umieszczenie go na stole montażowym i określenie pliku wideo do odtworzenia. Ponadto można ustawiać różne parametry przeznaczone do sterowania działaniem składnika oraz można opisywać pliki wideo.

**Korzystanie ze składnika FLVPlayback**

Składnik FLVPlayback udostępnia również aplikacyjny interfejs programistyczny ActionScript (API). Interfejs API zawiera następujące klasy, które zostały szczegółowo opisane w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*: CuePointType, FLVPlayback, FLVPlaybackCaptioning, NCManager, NCManagerNative, VideoAlign, VideoError, VideoPlayer, VideoState, a także kilka klas zdarzeń — AutoLayoutEvent, LayoutEvent, MetadataEvent, SkinErrorEvent, SoundEvent, VideoEvent i VideoProgressEvent.

Składnik FLVPlayback zawiera niestandardowe składniki UI FLV Playback. Składnik FLVPlayback stanowi połączenie obszaru wyświetlania oraz odtwarzacza wideo, w którym można wyświetlać plik wideo i sterować nim. Niestandardowe składniki UI FLV Playback udostępniają przyciski sterowania i mechanizmy, które mogą być używane do odtwarzania, zatrzymywania, wstrzymywania oraz sterowania plikiem wideo w inny sposób. Do tych elementów sterowania należą: BackButton, BufferingBar, CaptionButton (dla FLVPlaybackCaptioning), ForwardButton, FullScreenButton, MuteButton, PauseButton, PlayButton, PlayPauseButton, SeekBar, StopButton oraz VolumeBar. Składnik FLVPlayback oraz elementy sterujące niestandardowych składników UI FLV Playback pojawiają się na panelu Składniki, co zostało przedstawione na poniższym rysunku:



Składniki FLVPlayback na panelu Składniki

Proces dodawania elementów sterowania odtwarzaniem do składnika FLVPlayback jest nazywany procesem *tworzenia karnacji*. Składnik FLVPlayback ma wstępnie przypisaną domyślną karnację — SkinOverAll.swf, która udostępnia elementy sterujące przeznaczone do odtwarzania, zatrzymywania, cofania, przewijania, wyszukiwania, wyciszania, regulacji głośności, wyświetlania na pełnym ekranie oraz do zapisu. W celu zmiany karnacji można wykonać jedną z poniższych czynności:

- Wybierz karnację z kolekcji wstępnie zaprojektowanych karnacji
- Utwórz karnację niestandardową i dodaj ją do kolekcji wstępnie zaprojektowanych karnacji
- Wybierz poszczególne elementy sterowania spośród niestandardowych składników UI FLV Playback i dostosuj je

Po wybraniu wstępnie zaprojektowanej karnacji możliwe jest wybranie osobno jej koloru i wartości alfa podczas tworzenia lub w środowisku wykonawczym. Więcej informacji zawiera sekcja „[Wybór wstępnie zaprojektowanej karnacji](#)” na stronie 158.

Po wybraniu innej karnacji wybrana karnacja staje się nową, domyślną karnacją.

Więcej informacji na temat wybierania lub tworzenia karnacji dla składnika FLVPlayback zawiera sekcja „[Dostosowywanie składnika FLVPlayback](#)” na stronie 158.



## Tworzenie aplikacji za pomocą składnika FLVPlayback

Istnieją następujące sposoby dołączania składnika FLVPlayback do aplikacji:

- Przeciągnij składnik FLVPlayback z panelu Składniki na stół montażowy i określ wartość parametru `source`.
- Użyj kreatora Import wideo, aby utworzyć składnik na stole montażowym, a następnie dostosuj go, wybierając karnację.
- Użyj konstruktora `FLVPlayback()` w celu dynamicznego utworzenia instancji FLVPlayback na stole montażowym, przy założeniu, że składnik znajduje się w bibliotece.

**Uwaga:** Jeśli instancja FLVPlayback jest tworzona za pomocą ActionScript, należy również przypisać do niej karnację, poprzez ustawienie właściwości `skin` w ActionScript. Gdy karnacja zostanie w ten sposób zastosowana, nie będzie automatycznie publikowana z plikiem SWF. Należy skopiować plik SWF aplikacji oraz plik SWF karnacji na serwer aplikacji — w przeciwnym wypadku plik SWF karnacji nie będzie dostępny po uruchomieniu aplikacji.

### Przeciąganie składnika FLVPlayback z panelu Składniki

- 1 Na panelu Składniki kliknij przycisk Plus (+), aby otworzyć pozycję wideo.
- 2 Przeciągnij składnik FLVPlayback na stół montażowy.
- 3 Po wybraniu składnika FLVPlayback na stole montażowym należy zlokalizować komórkę Wartość dla parametru `source` na karcie Parametry w inspektorze składników, a następnie wprowadzić łańcuch, który będzie określał:
  - Lokalną ścieżkę do pliku wideo
  - Adres URL do pliku wideo
  - Adres URL pliku SMIL (Synchronized Multimedia Integration Language), który opisuje sposób odtwarzania pliku wideoInformacje o sposobach tworzenia plików SMIL, które opisują jeden lub większą liczbę plików FLV, zawiera sekcja „[Korzystanie z pliku SMIL](#)” na stronie 169.
- 4 Na karcie Parametry w inspektorze składników (po wybraniu składnika FLVPlayback na stole montażowym) kliknij komórkę Wartość dla parametru `skin`.
- 5 Kliknij ikonę szkła powiększającego, aby otworzyć okno dialogowe Wybierz karnację.
- 6 Można wybrać jedną z następujących opcji:
  - Z listy rozwijanej Karnacja wybierz jedną z wstępnie zaprojektowanych karnacji, która posłuży do dołączenia zestawu elementów sterowania odtwarzaniem do składnika.
  - Jeśli utworzono karnację niestandardową, wybierz z menu podręcznego opcję Adres URL do własnej karnacji, a następnie wprowadź do pola URL adres pliku SWF, który zawiera karnację.
  - Wybierz opcję Brak, a następnie przeciągnij niestandardowe składniki UI FLV Playback do stołu montażowego, aby dodać elementy sterowania odtwarzaniem.

**Uwaga:** W dwóch pierwszych przypadkach pojawi się podgląd karnacji, który będzie widoczny na panelu przeglądania powyżej menu podręcznego. W celu zmiany koloru karnacji można użyć obszaru wyboru kolorów.

Aby zmienić kolor niestandardowego elementu sterowania interfejsem użytkownika, należy zmodyfikować ten kolor. Więcej informacji na temat korzystania z niestandardowych elementów sterowania interfejsem użytkownika zawiera sekcja „[Tworzenie karnacji dla poszczególnych niestandardowych składników UI FLV Playback](#)” na stronie 160.

- 7 Kliknij przycisk OK, aby zamknąć okno dialogowe Wybierz karnację.
- 8 Wybierz opcje Sterowanie > Testuj film, aby uruchomić plik SWF i uruchomić wideo.

**Korzystanie ze składnika FLVPlayback**

W poniższej procedurze przedstawiono wykorzystanie kreatora Import wideo w celu dodania składnika FLVPlayback:

**Korzystanie ze składnika Import wideo:**

- 1 Wybierz opcje Plik > Importuj > Importuj wideo.
  - 2 Określ lokalizację pliku wideo, zaznaczając jedną z poniższych opcji:
    - Na komputerze lokalnym
    - Już zainstalowano na serwerze sieci Web, serwerze usługi Flash Video Streaming Service lub serwerze Flash Media Server
  - 3 W zależności od wybranej opcji wprowadź ścieżkę lub adres URL, który będzie określał lokalizację pliku wideo; następnie kliknij przycisk Dalej.
  - 4 Jeśli wybrano ścieżkę do pliku, pojawi się okno dialogowe Instalacja, na którym można wybrać jedną z opcji, aby określić sposób, w jaki plik wideo zostanie zainstalowany:
    - Progresywne pobieranie ze standardowego serwera sieci Web
    - Strumieniowanie z Flash Video Streaming Service
    - Strumieniowanie z serwera Flash Media Server
    - Osadź wideo w pliku SWF i odtwarzaj na osi czasu

***Ważne:** Nie należy wybierać opcji Osadź wideo. Składnik FLVPlayback odtwarza wyłącznie zewnętrzne wideo strumieniowe. Wybranie tej opcji nie spowoduje umieszczenia składnika FLVPlayback na stole montażowym.*
  - 5 Kliknij przycisk Dalej.
  - 6 Można wybrać jedną z następujących opcji:
    - Z listy rozwijanej Karnacja wybierz jedną z wstępnie zaprojektowanych karnacji, która posłuży do dołączenia zestawu elementów sterowania odtwarzaniem do składnika.
    - Jeśli utworzono karnację niestandardową dla składnika, wybierz z menu podręcznego opcję Adres URL do własnej karnacji, a następnie wprowadź do pola URL adres pliku SWF, który zawiera karnację.
    - Wybierz opcję Brak, a następnie przeciągnij niestandardowe składniki UI FLV Playback do stołu montażowego, aby dodać elementy sterowania odtwarzaniem.

***Uwaga:** W dwóch pierwszych przypadkach pojawi się podgląd karnacji, który będzie widoczny na panelu przeglądania powyżej menu podręcznego.*
  - 7 Kliknij przycisk OK, aby zamknąć okno dialogowe Wybierz karnację.
  - 8 Odczytaj zawartość okna dialogowego Zakończ importowanie wideo, aby określić następną operację, a następnie kliknij przycisk Zakończ.
  - 9 Jeśli plik FLA nie został zapisany, pojawi się okno dialogowe Zapisz jako.
  - 10 Wybierz opcje Sterowanie > Testuj film, aby uruchomić plik SWF i uruchomić wideo.
- Poniższa procedura opisuje dodawanie składnika FLVPlayback za pomocą ActionScript.

**Dynamiczne tworzenie instancji za pomocą ActionScript:**

- 1 Przeciągnij składnik FLVPlayback z panelu Składniki do panelu Biblioteka (Okno > Biblioteka).
- 2 Dodaj poniższy kod do panelu Operacje w klatce nr 1 na osi czasu. Zmień *install\_drive* na napęd, na którym zainstalowano program Flash, a następnie zmodyfikuj ścieżkę w celu uwzględnienia lokalizacji folderu Skins dla instalacji:

Na komputerze z systemem Windows:

```
import fl.video.*;
var my_FLVPlybk = new FLVPlayback();
my_FLVPlybk.x = 100;
my_FLVPlybk.y = 100;
addChild(my_FLVPlybk);
my_FLVPlybk.skin = "file:///install_drive|/Program Files/Adobe/Adobe Flash
CS5/en/Configuration/FLVPlayback Skins/ActionScript 3.0/SkinOverPlaySeekMute.swf"
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/water.flv";
```

Na komputerze z systemem Macintosh:

```
import fl.video.*;
var my_FLVPlybk = new FLVPlayback();
my_FLVPlybk.x = 100;
my_FLVPlybk.y = 100;
addChild(my_FLVPlybk);
my_FLVPlybk.skin = "file:///Macintosh HD:Applications:Adobe Flash
CS5:Configuration:FLVPlayback Skins:ActionScript 3.0SkinOverPlaySeekMute.swf"
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/water.flv";
```

**Uwaga:** Jeśli właściwości *source* i *skin* nie zostaną ustawione, wygenerowany klip będzie pusty.

- 3 Wybierz opcje Sterowanie > Testuj film, aby uruchomić plik SWF i uruchomić wideo.

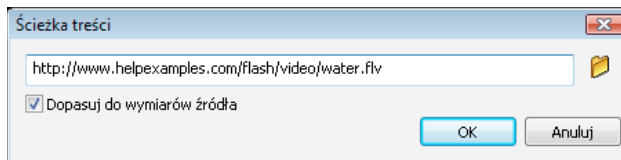
## Parametry składnika FLVPlayback

Dla każdej instancji składnika FLVPlayback można ustawić następujące parametry w inspektorze składników lub inspektorze właściwości: *align*, *autoPlay*, *cuePoints*, *preview*, *scaleMode*, *skin*, *skinAutoHide*, *skinBackgroundAlpha*, *skinBackgroundColor*, *source* oraz *volume*. Dla każdego z tych parametrów istnieje odpowiednia właściwość ActionScript o takiej samej nazwie. Przypisanie wartości dla tych parametrów jest równoważne z ustawieniem wstępnego stanu właściwości w aplikacji. Ustawienie właściwości w ActionScript powoduje zastąpienie wartości ustawionej w parametrze. Więcej informacji na temat możliwych wartości tych parametrów zawiera opis klasy FLVPlayback w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

### Określanie parametru *source* składnika FLVPlayback

Parametr *source* umożliwia określenie nazwy i lokalizacji pliku wideo, które definiują sposób odtwarzania pliku w programie Flash.

Otwórz okno dialogowe Ścieżka treści, klikając dwukrotnie komórkę Wartość dla parametru *source* w inspektorze składników.



Okno dialogowe Ścieżka treści dla składnika FLVPlayback

Okno dialogowe Ścieżka treści udostępnia pole wyboru Dopasuj do wymiarów źródła FLV, które określa, czy wymiary instancji FLVPlayback na stole montażowym mają być zgodne z wymiarami źródłowego pliku wideo. Wysokość i szerokość źródłowego pliku wideo to wymiary preferowane dla odtwarzania. Jeśli ta opcja zostanie wybrana, wymiary instancji FLVPlayback zostaną zmienione w taki sposób, aby były zgodne z wymiarami preferowanymi.

## Źródło

Wprowadź adres URL lub ścieżkę lokalną dla pliku wideo lub pliku XML, który opisuje sposób odtwarzania pliku wideo. Jeśli nie jest znana dokładna lokalizacja pliku wideo, kliknij ikonę folderu, aby otworzyć okno dialogowe przeglądarki w celu znalezienia odpowiedniej lokalizacji. Jeśli wyszukiwany plik wideo znajduje się na poziomie lub poniżej docelowego pliku SWF, program Flash automatycznie ustawi tę ścieżkę jako względną do lokalizacji, dzięki czemu możliwa będzie jej obsługa z serwera WWW. W przeciwnym wypadku ścieżka będzie ustawiona jako bezwzględna w systemie Windows lub Macintosh. Aby wprowadzić nazwę lokalnego pliku XML, wpisz ścieżkę i nazwę.

Jeśli zostanie określony adres URL HTTP, plik wideo będzie odtwarzany jako plik pobierany progresywnie. Jeśli zostanie określony adres URL RTMP, plik wideo będzie pobierany strumieniowo z serwera Flash Media Server lub usługi FVSS. Adres URL pliku XML może również wskazywać plik wideo pobierany strumieniowo z serwera Flash Media Server lub usługi FVSS.

### Ważne:

Możliwe jest również określenie lokalizacji pliku SMIL, który określi sposób odtwarzania wielu strumieni wideo dla różnych przepustowości. W pliku używany jest język Synchronized Multimedia Integration Language (SMIL), który służy do opisywania plików FLV. Opis pliku SMIL, patrz „[Korzystanie z pliku SMIL](#)” na stronie 169.

Nazwę i lokalizację pliku wideo można również określić za pomocą właściwości `FLVPlayback.source` ActionScript oraz za pomocą metod `FLVPlayback.play()` i `FLVPlayback.load()`. Te trzy alternatywy mają priorytet wyższy niż parametr `source` w inspektorze składników. Więcej informacji zawierają pozycje `FLVPlayback.source`, `FLVPlayback.play()` i `FLVPlayback.load()` dla klasy `FLVPlayback` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

## Obsługa trybu pełnoekranowego

Składnik `FLVPlayback` w wersji ActionScript 3.0 obsługuje tryb pełnoekranowy, który wymaga odtwarzacza Flash Player 9.0.28.0 (lub jego późniejszej wersji), a także odpowiedniego ustawienia HTML dla wyświetlania na pełnym ekranie. Niektóre wstępnie zaprojektowane karnacje udostępniają przycisk przełączania, który umożliwia włączanie i wyłączanie trybu pełnoekranowego. Ikona `FullScreenButton` jest widoczna po prawej stronie paska sterowania na poniższej ilustracji.



Ikona trybu pełnoekranowego na pasku sterowania

Funkcje obsługi trybu pełnoekranowego są dostępne tylko wówczas, gdy dla właściwości `fullScreenTakeOver` ustawiona jest wartość `true`, co jest ustawieniem domyślnym.

Obsługa trybu pełnoekranowego może być dostępna w przypadku stosowania, a także niestosowania przyspieszania sprzętowego. Informacje na temat obsługi przyspieszania sprzętowego zawiera sekcja „[Akcelerator sprzętowy](#)” na stronie 147.

### Aby zaimplementować funkcję obsługi trybu pełnoekranowego dla składnika `FLVPlayback`:

- 1 Dodaj składnik `FLVPlayback` do aplikacji i przypisz do niego plik wideo.
- 2 Wybierz karnację dla składnika `FLVPlayback`, który zawiera przycisk trybu pełnoekranowego (np. `SkinUnderPlaySeekFullscreen.swf`) lub dodaj składnik `FullScreenButton` (interfejsu użytkownika) do składnika `FLVPlayback` z sekcji Wideo panelu Składniki.
- 3 Wybierz polecenie Plik > Ustawienia publikowania.
- 4 W oknie dialogowym Ustawienia publikowania kliknij kartę HTML.

- 5 Na karcie HTML z menu podręcznego Szablon wybierz opcję Flash z obsługą pełnego ekranu.
- 6 Na karcie HTML zaznacz pole wyboru Wykryj wersję Flash, a następnie określ wersję 9.0.28 lub nowszą — w zależności od używanej wersji programu Flash Player.
- 7 Wybierz kartę Formaty i upewnij się, że zaznaczone są opcje Flash (.swf) i HTML (.html). Domyślne nazwy plików można zastąpić.
- 8 Kliknij przycisk Publikuj, a następnie przycisk OK.

Zamiast wykonywać krok 7 możesz kliknąć przycisk OK, a następnie wybrać polecenia Plik > Podgląd publikowania > Domyślne - (HTML), aby automatycznie otworzyć wyeksportowany plik HTML w domyślnej przeglądarce. W przeciwnym wypadku otwórz wyeksportowany plik HTML za pomocą przeglądarki, aby przetestować opcję trybu pełnoekranowego.

Aby dodać składnik FLVPlayback z funkcją obsługi trybu pełnoekranowego do strony WWW, otwórz wyeksportowany plik HTML i skopiuj kod, który osadza plik SWF w pliku HTML dla strony WWW. Ten kod powinien przypominać poniższy:

```
//from the <head> section

<script language="javascript"> AC_FL_RunContent = 0; </script>
<script language="javascript"> DetectFlashVer = 0; </script>
<script src="AC_RunActiveContent.js" language="javascript"></script>
<script language="JavaScript" type="text/javascript">
<!--
// -----
// Globals
// Major version of Flash required
var requiredMajorVersion = 9;
// Minor version of Flash required
var requiredMinorVersion = 0;
// Revision of Flash required
var requiredRevision = 28;
// -----
// -->
</script>

//and from the <body> section

<script language="JavaScript" type="text/javascript">
<!--
if (AC_FL_RunContent == 0 || DetectFlashVer == 0) {
    alert("This page requires AC_RunActiveContent.js.");
} else {
    var hasRightVersion = DetectFlashVer(requiredMajorVersion,
        requiredMinorVersion, requiredRevision);
    if(hasRightVersion) { // if we've detected an acceptable version
        // embed the Flash movie
        AC_FL_RunContent(
            &apos;codebase&apos;, &apos;http://download.macromedia.com/pub/
            shockwave/cabs/flash/swflash.cab#version=9,0,28,0&apos;;,
            &apos;width&apos;, &apos;550&apos;;,
            &apos;height&apos;, &apos;400&apos;;,
            &apos;src&apos;, &apos;fullscreen&apos;;,
            &apos;quality&apos;, &apos;high&apos;;,
            &apos;pluginspage&apos;, &apos;http://www.macromedia.com/go/
            getflashplayer&apos;;,
```

```
        &apos;align&apos;;, &apos;middle&apos;;,  
        &apos;play&apos;;, &apos;true&apos;;,  
        &apos;loop&apos;;, &apos;true&apos;;,  
        &apos;scale&apos;;, &apos;showall&apos;;,  
        &apos;wmode&apos;;, &apos;window&apos;;,  
        &apos;devicefont&apos;;, &apos;false&apos;;,  
        &apos;id&apos;;, &apos;fullscreen&apos;;,  
        &apos;bgcolor&apos;;, &apos;#ffffff&apos;;,  
        &apos;name&apos;;, &apos;fullscreen&apos;;,  
        &apos;menu&apos;;, &apos;true&apos;;,  
        &apos;allowScriptAccess&apos;;, &apos;sameDomain&apos;;,  
        &apos;allowFullScreen&apos;;, &apos;true&apos;;,  
        &apos;movie&apos;;, &apos;fullscreen&apos;;,  
        &apos;salign&apos;;, &apos;&apos;; ); //end AC code  
    } else { // Flash is too old or we can&apos;t detect the plug-in.  
        var alternateContent = &apos;Alternative HTML content should be placed  
            here.&apos;;  
        + &apos;This content requires Adobe Flash Player.&apos;;  
        + &apos;<a href=http://www.macromedia.com/go/getflash/>Get Flash</a>  
            &apos;;;  
        document.write(alternateContent); // Insert non-Flash content.  
    }  
}  
// -->  
</script>  
<noscript>  
    // Provide alternative content for browsers that do not support scripting  
    // or for those that have scripting disabled.  
    Alternative HTML content should be placed here. This content requires Adobe Flash Player.  
    <a href="http://www.macromedia.com/go/getflash/">Get Flash</a>  
</noscript>
```

Możliwe jest również użycie wyeksportowanego pliku HTML jako szablonu dla strony WWW oraz dodanie do niego innej treści. W takim przypadku należy jednak zmienić nazwę pliku HTML w taki sposób, aby nie doszło do jej przypadkowego zastąpienia w razie ponownego eksportu pliku HTML FLVPlayback z programu Flash.

W każdym z tych przypadków należy załadować do serwera WWW plik AC\_RunActiveContent.js, który zostanie wyeksportowany do tego samego folderu, co plik HTML.

Obsługa trybu pełnoekranowego w języku ActionScript obejmuje właściwości `fullScreenBackgroundColor`, `fullScreenSkinDelay`, i `fullScreenTakeOver`, a także metodę `enterFullScreenDisplayState()`. Więcej informacji na temat tych elementów języka ActionScript zawiera [Skorowidz języka ActionScript 3.0 dla programu Flash Professional](#).

### Stosowanie metody `enterFullScreenDisplayState()`

Tryb pełnoekranowy można również uruchomić, wywołując metodę `enterFullScreenDisplayState()` ActionScript, co zostało przedstawione w poniższym przykładzie.

```
function handleClick(e:MouseEvent):void {  
    myFLVPlayback.enterFullScreenDisplayState();  
}  
myButton.addEventListener(MouseEvent.CLICK, handleClick);
```

**Korzystanie ze składnika FLVPlayback**

W tym przykładzie tryb pełnoekranowy *nie* jest wywoływany poprzez kliknięcie przycisku przełączania trybu pełnoekranowego na karnacji FLVPlayback — jest uruchamiany poprzez kliknięcie przycisku (MyButton), który został umieszczony przez twórcę strony WWW i służy do wywoływania tego trybu. Kliknięcie przycisku powoduje uruchomienie programu obsługi zdarzenia handleClick, który wywołuje metodę enterFullScreenDisplayState().

Metoda enterFullScreenDisplayState() ustawia dla właściwości Stage.displayState wartość StageDisplayState.FULL\_SCREEN i dlatego obowiązują dla niej te same ograniczenia, co dla właściwości displayState. Więcej informacji na temat metody enterFullScreenDisplayState() i właściwości Stage.displayState zawiera [Skorowidz języka ActionScript 3.0 dla programu Flash Professional](#).

**Akcelerator sprzętowy**

Program Flash Player 9.0.115.0 i jego późniejsze wersje zawierają kod, który korzysta z dostępnych urządzeń wideo w celu poprawy wydajności i wierności plików FLV, które składnik FLVPlayback odtwarza w trybie pełnoekranowym. Jeśli spełnione są wszystkie wymagania wstępne, a dla właściwości fullScreenTakeOver ustawiono wartość true, wówczas w celu skalowania pliku wideo program Flash Player wykorzystuje przyspieszenie sprzętowe zamiast rozwiązań programowych. Jeśli składnik FLVPlayback działa we wcześniejszej wersji programu Flash Player lub jeśli wymagania wstępne dla przyspieszenia sprzętowego nie istnieją, program Flash Player dokona zwiększenia skali pliku wideo, tak jak poprzednio.

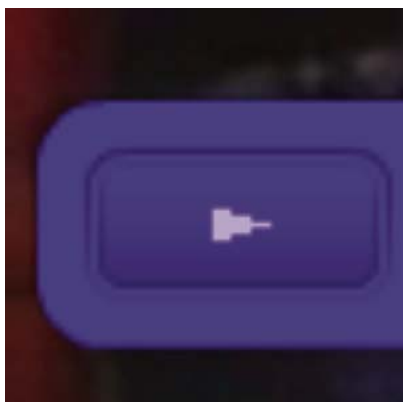
Aby wykorzystać przyspieszenie sprzętowe dla obsługi trybu pełnoekranowego, komputer musi być wyposażony w kartę graficzną zgodną z formatem DirectX 7 z 4 MB lub większą liczbą pamięci VRAM (wideo RAM). Obsługa urządzeń jest dostępna w systemach Windows 2000 lub Mac OS X 10.2, a także w późniejszych wersjach tych systemów operacyjnych. DirectX® udostępnia interfejsy API stanowiące połączenie między oprogramowaniem i sprzętem, które umożliwia między innymi przyspieszenie grafik dwu- i trójwymiarowych.

W celu wykorzystania przyspieszenia sprzętowego należy wywołać tryb pełnoekranowy na jeden z poniższych sposobów:

- Za pomocą przycisku przełączania trybu pełnoekranowego w karnacji FLVPlayback
- Za pomocą elementu do sterowania wideo FullScreenButton
- Za pomocą metody enterFullScreenDisplayState() języka ActionScript. Więcej informacji zawiera sekcja „[Stosowanie metody enterFullScreenDisplayState\(\)](#)” na stronie 146.

Jeśli tryb pełnoekranowy zostanie wywołany poprzez ustawienie dla właściwości Stage.displayState wartości StageDisplayState.FULLSCREEN, wówczas składnik FLVPlayback nie korzysta z przyspieszenia sprzętowego, nawet jeśli dostępne są odpowiednie urządzenia wideo i pamięć.

Jedną konsekwencją korzystania z przyspieszenia sprzętowego dla obsługi trybu pełnoekranowego jest to, że karnacje FLVPlayback są skalowane wraz z odtwarzaczem wideo i plikiem wideo. Poniższy obraz pokazuje wpływ trybu pełnoekranowego z przyspieszeniem sprzętowym na karnację FLVPlayback, której szczegół został przedstawiony w pełnej rozdzielczości.



*Tryb pełnoekranowy na monitorze 1600 x 1200 z wideo o rozdzielczości 320x240 pikseli*

Ten obraz przedstawia wynik korzystania z trybu pełnoekranowego na monitorze 1600 x 1200 z plikiem wideo, którego szerokość wynosi 320, a wysokość 240 pikseli — są to domyślne wymiary dla składnika FLVPlayback. Efekt zniekształcenia karnacji jest bardziej widoczny w przypadku plików FLV o mniejszych rozmiarach lub na większych monitorach. I przeciwnie — efekt zniekształcenia jest mniej widoczny na większych plikach FLV i mniejszych monitorach. Przykład: zmiana wielkości 640 x 480 na 1600 x 1200 powoduje zwiększenie wielkości karnacji, ale wówczas karnacja jest mniej zniekształcona.

Istnieje możliwość ustawienia właściwości `skinScaleMaximum` w celu ograniczenia skalowania karnacji FLVPlayback. Wartością domyślną jest 4,0 lub 400%. Ograniczanie skalowania karnacji wymaga jednak wykorzystania sprzętu oraz oprogramowania w celu skalowania pliku FLV, co może negatywnie wpływać na wydajność wyświetlania plików FLV o większych rozdzielczościach niż zakodowane przy wyższej szybkości transmisji. Jeśli plik wideo jest duży (na przykład: szerokość 640 pikseli lub więcej, wysokość 480 pikseli lub więcej), nie należy ustawiać dla właściwości `skinScaleMaximum` małej wartości, ponieważ może to spowodować znaczne problemy z wydajnością w przypadku dużych monitorów. Właściwość `skinScaleMaximum` umożliwia zarządzanie zmianami (obustronnymi) wydajności i jakości oraz wyglądem dużych karnacji.

## Zamknięcie trybu pełnoekranowego

Aby zamknąć tryb pełnoekranowy, kliknij ponownie przycisk tego trybu lub naciśnij klawisz Esc.

Ustawienie poniższych właściwości i wywołanie poniższych metod może spowodować zmiany układu, które spowodują zamknięcie trybu pełnoekranowego przez składnik FLVPlayback: `height`, `registrationHeight`, `registrationWidth`, `registrationX`, `registrationY`, `scaleX`, `scaleY`, `width`, `x`, `y`, `setScale()` lub `setSize()`.

Jeśli użytkownik ustawi właściwości `align` lub `scaleMode`, składnik FLVPlayback ustawi dla nich wartości `center` i `maintainAspectRatio` do czasu zamknięcia trybu pełnoekranowego.

Zmiana wartości właściwości `fullScreenTakeOver` z `true` na `false` podczas korzystania z trybu pełnoekranowego również sprawia, że program Flash zamyka tryb pełnoekranowy.



## Wyrównanie układu w celu odtwarzania wielu plików wideo

Składnik FLVPlayback języka ActionScript 3.0 udostępnia właściwość `align`, która określa, czy plik wideo, którego wielkość została zmieniona, powinien być ustawiany w środku, u góry, u dołu, po prawej albo lewej stronie składnika. Oprócz właściwości `x`, `y`, `width` i `height` składnik języka ActionScript 3.0 udostępnia również właściwości `registrationX`, `registrationY`, `registrationWidth` i `registrationHeight`. Wstępnie te właściwości są zgodne z właściwościami `x`, `y`, `width` i `height`. Podczas ładowania kolejnych plików wideo automatyczna zmiana układu nie powoduje zmiany tych właściwości, dlatego nowy plik wideo może zostać ustawiony w tym samym miejscu. Jeśli `scaleMode = VideoScaleMode.MAINTAIN_ASPECT_RATIO`, wówczas kolejne pliki FLV mogą otrzymać pierwotne wymiary składnika, a szerokość i wysokość składnika nie zostanie zmieniona.

## Automatyczne odtwarzanie plików wideo pobieranych progresywnie

Podczas progresywnego pobierania pliku wideo składnik FLVPlayback rozpoczyna odtwarzanie pliku wideo tylko wówczas, gdy wystarczająca ilość pliku zostanie pobrana — tylko wtedy możliwe jest odtworzenie pliku wideo od początku do końca.

Jeśli użytkownik chce rozpocząć odtwarzanie pliku zanim wystarczająca jego część zostanie pobrana, musi wywołać metodę `play()` bez żadnych parametrów.

Jeśli użytkownik chce powrócić do ustawienia, w którym program oczekuje na pobranie dostatecznej ilości pliku wideo, należy wywołać metodę `pause()`, a następnie metodę `playWhenEnoughDownloaded()`.

## Korzystanie z punktów sygnalizacji

Punkt sygnalizacji to punkt, w którym odtwarzacz wideo wywołuje zdarzenie `cuePoint` podczas odtwarzania pliku wideo. Punkty sygnalizacji można dodawać do pliku FLV, gdy wymagane jest wykonanie operacji dla innego elementu na stronie WWW. Wymagane może być na przykład wyświetlenie tekstu lub grafiki, zsynchronizowanie z animacją Flash, wstrzymanie pliku FLV, wyszukanie innego punktu w wideo albo przełączenie na inny plik FLV. Punkty sygnalizacji umożliwiają uzyskanie kontroli w kodzie ActionScript w celu synchronizowania punktów w pliku FLV z innymi operacjami na stronie WWW.

Istnieją trzy typy punktów sygnalizacji: nawigacja, zdarzenia i ActionScript. Punkty sygnalizacji typu nawigacja i zdarzenia są również określane jako *osadzone* punkty sygnalizacji, ponieważ są osadzone w strumieniu pliku FLV w pakiecie metadanych pliku FLV.

*Punkt sygnalizacji typu nawigacja* umożliwia wyszukiwanie odpowiednich klatek w pliku FLV, ponieważ punkt tworzy klatkę kluczową w pliku FLV tak blisko określonego czasu, jak to możliwe. *Klatka kluczowa* jest segmentem danych, jaki występuje między klatkami obrazu w strumieniu pliku FLV. Podczas wyszukiwania punktu sygnalizacji typu nawigacja składnik wyszukuje klatkę kluczową i uruchamia zdarzenie `cuePoint`.

*Punkt sygnalizacji typu zdarzenie* umożliwia zsynchronizowanie punktu w czasie w pliku FLV z zewnętrznym zdarzeniem na stronie WWW. Do zdarzenia `cuePoint` dochodzi dokładnie w określonym czasie. Punkty sygnalizacyjne typu nawigacja i zdarzenia można osadzać w plikach FLV za pomocą kreatora Import wideo lub za pomocą programu Flash Video Encoder. Więcej informacji na temat kreatora Import wideo i programu Flash Video Encoder zawiera rozdział 16, „Wideo”, w podręczniku *Korzystanie z programu Flash*.

**Korzystanie ze składnika FLVPlayback**

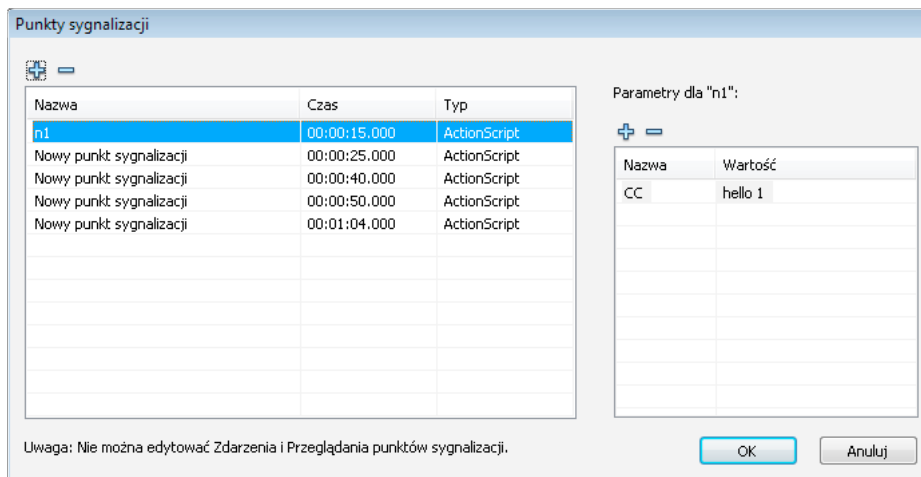
*Punkt sygnalizacji ActionScript* jest zewnętrznym punktem sygnalizacji, który można dodawać za pośrednictwem okna dialogowego punktów sygnalizacji wideo (w programie Flash) lub za pomocą metody `FLVPlayback.addASCuePoint()`. Składnik zapisuje i śledzi punkty sygnalizacji ActionScript poza plikiem FLV i w konsekwencji są one mniej dokładne niż osadzone punkty sygnalizacji. Punkty sygnalizacji ActionScript zachowują dokładność do jednej dziesiątej sekundy. W celu zwiększenia dokładności punktów sygnalizacji ActionScript należy zmniejszyć wartość właściwości `playheadUpdateInterval`, ponieważ składnik generuje zdarzenie `cuePoint` dla punktów sygnalizacji ActionScript, gdy dojdzie do aktualizacji stanu głowicy odtwarzania. Więcej informacji zawiera opis właściwości `FLVPlayback.playheadUpdateInterval` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

W języku ActionScript oraz w metadanych pliku FLV punkt sygnalizacji jest reprezentowany jako obiekt z następującymi właściwościami: `name`, `time`, `type` i `parameters`. Właściwość `name` jest ciągiem znaków, który zawiera nazwę przypisaną do punktu sygnalizacji. Właściwość `time` jest liczbą reprezentującą czas wystąpienia punktu sygnalizacji w godzinach, minutach, sekundach i milisekundach (GG:MM:SS.mmm). Właściwość `type` jest ciągiem znaków, którego wartość to "navigation", "event" lub "actionscript" — w zależności od typu utworzonego punktu sygnalizacji. Właściwość `parameters` jest tablicą określonych par nazwa/wartość.

Gdy dojdzie do zdarzenia `cuePoint`, obiekt punktu sygnalizacji staje się dostępny w obiekcie zdarzenia za pośrednictwem właściwości `info`.

**Korzystanie z okna dialogowego Punkty kontrolne Flash Video**

Otwórz okno dialogowe Punkty kontrolne Flash Video, klikając dwukrotnie komórkę Wartość parametru `cuePoints` w inspektorze właściwości. Okno dialogowe przypomina okno przedstawione poniżej:



Okno dialogowe Punkty sygnalizacji

W tym oknie dialogowym widoczne są osadzone punkty sygnalizacji i punkty sygnalizacji ActionScript. Z tego okna dialogowego można korzystać w celu dodawania i usuwania punktów sygnalizacji ActionScript, a także parametrów punktów sygnalizacji. W tym oknie można również włączać i wyłączać osadzone punkty sygnalizacji. Osadzonych punktów sygnalizacji nie można jednak dodawać, zmieniać ani usuwać.

**Dodawanie punktu sygnalizacji ActionScript:**

- 1 Kliknij dwukrotnie komórkę wartości parametru `cuePoints` w Inspektorze składników, aby otworzyć okno dialogowe punktów sygnalizacji Flash.
- 2 Kliknij znak plus (+) w lewym górnym rogu, powyżej listy punktów sygnalizacji, aby dodać domyślną pozycję punktu sygnalizacji ActionScript.

- 3 Kliknij napis Nowy punkt sygnalizacji w kolumnie Nazwa, a następnie zmodyfikuj napis, aby nazwać punkt sygnalizacji.
- 4 Kliknij wartość Czas 00:00:00:000, aby zmodyfikować ją i przypisać czas wystąpienia punktu sygnalizacji. Czas można określić w godzinach, minutach, sekundach i milisekundach (GG:MM:SS.mmm).  
Jeśli istnieje wiele punktów sygnalizacji, nowy punkt zostanie przesunięty w oknie dialogowym zgodnie z chronologiczną pozycją na liście.
- 5 Aby dodać parametr dla wybranego punktu sygnalizacji, kliknij znak plus (+) powyżej sekcji Parametry, a następnie wprowadź wartości do kolumn Nazwa i Wartość. Powtórz tę czynność dla każdego parametru.
- 6 Aby dodać więcej punktów sygnalizacji ActionScript, powtórz dla każdego z nich czynności od 2 do 5.
- 7 Kliknij przycisk OK, aby zapisać zmiany.

#### Usuwanie punktu sygnalizacji ActionScript:

- 1 Kliknij dwukrotnie komórkę wartości parametru `cuePoints` w Inspektorze składników, aby otworzyć okno dialogowe punktów sygnalizacji Flash.
- 2 Wybierz punkt sygnalizacji, który chcesz usunąć.
- 3 Kliknij znak minus (-) w lewym górnym rogu, powyżej listy punktów sygnalizacji, aby usunąć punkt.
- 4 Powtórz kroki 2 i 3 dla każdego punktu sygnalizacji, który chcesz usunąć.
- 5 Kliknij przycisk OK, aby zapisać zmiany.

#### Aby włączyć lub wyłączyć punkt sygnalizacji w osadzonym pliku FLV:

- 1 Kliknij dwukrotnie komórkę wartości parametru `cuePoints` w Inspektorze składników, aby otworzyć okno dialogowe punktów sygnalizacji Flash.
- 2 Wybierz punkt sygnalizacji, który chcesz włączyć lub wyłączyć.
- 3 Kliknij wartość w kolumnie Typ, aby wywołać menu podręczne, lub kliknij strzałkę w dół.
- 4 Kliknij nazwę typu punktu sygnalizacji (np. Zdarzenie lub Nawigacja), aby go włączyć. Kliknij przycisk Wyłącz, aby wyłączyć.
- 5 Kliknij przycisk OK, aby zapisać zmiany.

#### Korzystanie z punktów sygnalizacji z ActionScript

Język ActionScript może służyć do dodawania punktów sygnalizacji ActionScript; nasłuchiwanie zdarzeń `cuePoint`; wyszukiwania punktów sygnalizacji dowolnego lub określonego typu; wyszukiwania punktów sygnalizacji typu nawigacja; włączania i wyłączania punktów sygnalizacji; sprawdzania, czy punkt sygnalizacji jest włączony oraz do usuwania punktów sygnalizacji.

Przykłady w tej sekcji przedstawiają używanie pliku FLV o nazwie `cuepoints.flv`, który zawiera następujące trzy punkty sygnalizacji:

Nazwa	Godzina	Typ
point1	00:00:00.418	Nawigacja
point2	00:00:07.748	Nawigacja
point3	00:00:16.020	Nawigacja

### Dodawanie punktów sygnalizacji ActionScript

Punkty sygnalizacji ActionScript można dodawać do pliku FLV za pomocą metody `addASCuePoint()`. W poniższym przykładzie przedstawiono dodawanie dwóch punktów sygnalizacji ActionScript do pliku FLV gotowego do odtworzenia. Poniższy przykład dodaje obiekt punktu sygnalizacji, który określa godzinę, nazwę i typ punktu sygnalizacji w jego właściwościach. Drugie wywołanie określa czas i nazwę za pomocą parametrów `time` i `name` metody.

```
// Requires an FLVPlayback instance called my_FLVPlayback on Stage
import fl.video.*;
import fl.video.MetadataEvent;
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var cuePt:Object = new Object(); //create cue point object
cuePt.time = 2.02;
cuePt.name = "ASpt1";
cuePt.type = "actionscript";
my_FLVPlayback.addASCuePoint(cuePt); //add AS cue point
// add 2nd AS cue point using time and name parameters
my_FLVPlayback.addASCuePoint(5, "ASpt2");
```

Więcej informacji zawiera opis metody `FLVPlayback.addASCuePoint()` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

### Nasłuchiwanie zdarzeń cuePoint

Zdarzenie `cuePoint` umożliwia kontrolowanie (za pośrednictwem kodu ActionScript) czasu wystąpienia zdarzenia `cuePoint`. Gdy w poniższym przykładzie wystąpią punkty sygnalizacji, detektor zdarzeń `cuePoint` wywoła funkcję obsługi zdarzeń, która wyświetli wartość właściwości `playheadTime`, a także nazwę i typ punktu sygnalizacji. W celu wyświetlenia rezultatów należy użyć tego przykładu w połączeniu z przykładem z poprzedniej sekcji — Dodawanie punktów sygnalizacji ActionScript.

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

Więcej informacji na temat zdarzenia `cuePoint` zawiera zdarzenie `FLVPlayback.cuePoint` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

### Wyszukiwanie punktów sygnalizacji

Za pomocą języka ActionScript można znaleźć punkt sygnalizacji dowolnego typu, znaleźć punkt sygnalizacji najbliższy punktu czasowego lub znaleźć następny punkt sygnalizacji o określonej nazwie.

Moduł obsługi zdarzeń `ready_listener()` w poniższym przykładzie wywołuje metodę `findCuePoint()` w celu znalezienia punktu sygnalizacji `ASpt1`, a następnie wywołuje metodę `findNearestCuePoint()` w celu znalezienia punktu sygnalizacji typu Nawigacja, który znajduje się najbliższe czasu wystąpienia punktu sygnalizacji `ASpt1`:

**Korzystanie ze składnika FLVPlayback**

```

import fl.video.FLVPlayback;
import fl.video.CuePointType;
import fl.video.VideoEvent;
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var rtn_obj:Object; //create cue point object
my_FLVPlayback.addASCuePoint(2.02, "ASpt1");//add AS cue point
function ready_listener(eventObject:VideoEvent):void {
    rtn_obj = my_FLVPlayback.findCuePoint("ASpt1", CuePointType.ACTIONSCRIPT);
    traceit(rtn_obj);
    rtn_obj = my_FLVPlayback.findNearestCuePoint(rtn_obj.time, CuePointType.NAVIGATION);
    traceit(rtn_obj);
}
my_FLVPlayback.addEventListener(VideoEvent.READY, ready_listener);
function traceit(cuePoint:Object):void {
    trace("Cue point name is: " + cuePoint.name);
    trace("Cue point time is: " + cuePoint.time);
    trace("Cue point type is: " + cuePoint.type);
}

```

W poniższym przykładzie moduł obsługi zdarzeń `ready_listener()` wyszukuje punkt sygnalizacji `ASpt` i wywołuje metodę `findNextCuePointWithName()` w celu znalezienia następnego punktu sygnalizacji o takiej samej nazwie:

```

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

```

Więcej informacji na temat wyszukiwania punktów sygnalizacji zawiera opis metod `FLVPlayback.findCuePoint()`, `FLVPlayback.findNearestCuePoint()` i `FLVPlayback.findNextCuePointWithName()` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

**Wyszukiwanie punktów sygnalizacji typu nawigacja**

Istnieje możliwość wyszukania punktu sygnalizacji typu nawigacja, wyszukiwania kolejnego punktu sygnalizacji od określonego czasu, a także wyszukiwania poprzedniego punktu sygnalizacji typu nawigacja (od określonego czasu). Poniższy przykład prezentuje odtwarzanie pliku FLV `cuepoints.flv`, a ponadto wyszukuje punkt sygnalizacji po upływie 7.748 sekundy od wystąpienia zdarzenia `ready`. Gdy wystąpi zdarzenie `cuePoint` w przykładzie wywoływana jest metoda `seekToPrevNavCuePoint()` w celu znalezienia pierwszego punktu sygnalizacji. Gdy wystąpi to zdarzenie `cuePoint`, w przykładzie wywoływana jest metoda `seekToNextNavCuePoint()` w celu znalezienia ostatniego punktu sygnalizacji poprzez dodanie 10 sekund do wartości `eventObject.info.time`, która jest czasem bieżącego punktu sygnalizacji.

**Korzystanie ze składnika FLVPlayback**

```
import fl.video.*;

my_FLVPlybk.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:Object):void {
    my_FLVPlybk.seekToNavCuePoint("point2");
}
my_FLVPlybk.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace(eventObject.info.time);
    if(eventObject.info.time == 7.748)
        my_FLVPlybk.seekToPrevNavCuePoint(eventObject.info.time - .005);
    else
        my_FLVPlybk.seekToNextNavCuePoint(eventObject.info.time + 10);
}
my_FLVPlybk.source = "http://helpexamples.com/flash/video/cuepoints.flv";
```

Więcej informacji zawiera opis metod `FLVPlayback.seekToNavCuePoint()`, `FLVPlayback.seekToNextNavCuePoint()` i `FLVPlayback.seekToPrevNavCuePoint()` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

**Włączanie i wyłączanie punktów sygnalizacji w osadzonym pliku FLV**

Do włączania i wyłączania punktów sygnalizacji w osadzonym pliku FLV służy metoda `setFLVCuePointEnabled()`. Wyłączone punkty sygnalizacji nie wywołują zdarzeń `cuePoint` i nie działają z metodami `seekToCuePoint()`, `seekToNextNavCuePoint()` oraz `seekToPrevNavCuePoint()`. Wyłączone punkty sygnalizacji można wyszukiwać za pomocą metod `findCuePoint()`, `findNearestCuePoint()` i `findNextCuePointWithName()`.

Aby sprawdzić, czy punkt sygnalizacji w osadzonym pliku FLV jest włączony, należy użyć metody `isFLVCuePointEnabled()`. W poniższym przykładzie przedstawiono wyłączanie punktów sygnalizacji `point2` i `point3`, gdy plik wideo jest gotowy do odtwarzania. Po wystąpieniu pierwszego zdarzenia `cuePoint` moduł obsługi zdarzeń sprawdza, czy punkt sygnalizacji `point3` jest wyłączony. Jeśli jest wyłączony, włącza go.

```
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv";
my_FLVPlybk.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:VideoEvent):void {
    my_FLVPlybk.setFLVCuePointEnabled(false, "point2");
    my_FLVPlybk.setFLVCuePointEnabled(false, "point3");
}
my_FLVPlybk.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace("Cue point time is: " + eventObject.info.time);
    trace("Cue point name is: " + eventObject.info.name);
    trace("Cue point type is: " + eventObject.info.type);
    if (my_FLVPlybk.isFLVCuePointEnabled("point2") == false) {
        my_FLVPlybk.setFLVCuePointEnabled(true, "point2");
    }
}
```

Więcej informacji zawiera opis metod `FLVPlayback.isFLVCuePointEnabled()` i `FLVPlayback.setFLVCuePointEnabled()` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

**Usuwanie punktu sygnalizacji ActionScript**

Punkt sygnalizacji ActionScript można usunąć za pomocą metody `removeASCuePoint()`. W poniższym przykładzie przedstawiono usuwanie punktu sygnalizacji `ASpt2`, gdy wystąpi punkt sygnalizacji `ASpt1`:

**Korzystanie ze składnika FLVPlayback**

```
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
my_FLVPlybk.addASCuePoint(2.02, "ASpt1");//add AS cue point
my_FLVPlybk.addASCuePoint(3.4, "ASpt2");//add 2nd Aspt
my_FLVPlybk.addEventListener(MouseEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace("Cue point name is: " + eventObject.info.name);
    if (eventObject.info.name == "ASpt1") {
        my_FLVPlybk.removeASCuePoint("ASpt1");
        trace("Removed cue point ASpt1");
    }
}
```

Więcej informacji zawiera opis metody `FLVPlayback.removeASCuePoint()` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

**Odtwarzanie wielu plików wideo**

Pliki wideo można odtwarzać kolejno w instancji `FLVPlayback` poprzez ładowanie nowego adresu URL do właściwości `source`, gdy zakończy się odtwarzanie poprzedniego pliku. Na przykład poniższy kod ActionScript wykrywa zdarzenie `complete`, które występuje, gdy zakończy się odtwarzanie pliku wideo. Gdy to zdarzenie wystąpi, kod ustawia nazwę i lokalizację nowego pliku wideo we właściwości `source`, a następnie wywołuje metodę `play()` w celu odtworzenia nowego wideo.

```
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/clouds.flv";
my_FLVPlybk.addEventListener(VideoEvent.COMPLETE, complete_listener);
// listen for complete event; play new FLV
function complete_listener(eventObject:VideoEvent):void {
    if (my_FLVPlybk.source == "http://www.helpexamples.com/flash/video/clouds.flv") {
        my_FLVPlybk.play("http://www.helpexamples.com/flash/video/water.flv");
    }
};
```

**Korzystanie z wielu odtwarzaczy wideo**

W jednej instancji składnika `FLVPlayback` można otworzyć wiele odtwarzaczy wideo, aby odtwarzać wiele plików wideo i przełączać je podczas odtwarzania.

Do utworzenia pierwszego odtwarzacza wideo dochodzi po przeciągnięciu składnika `FLVPlayback` na stół montażowy. Składnik automatycznie przypisuje pierwszemu odtwarzaczowi wideo numer 0 i ustawia go jako odtwarzacz domyślny. W celu utworzenia dodatkowego odtwarzacza wideo ustaw nową liczbę dla właściwości `activeVideoPlayerIndex`. Ustawienie właściwości `activeVideoPlayerIndex` powoduje również ustawienie wybranego odtwarzacza wideo jako *aktywnego* odtwarzacza wideo — na ten odtwarzacz będą wpływać właściwości i metody klasy `FLVPlayback`. Ustawienie właściwości `activeVideoPlayerIndex` nie powoduje jednak wyświetlenia odtwarzacza wideo. W celu wyświetlenia odtwarzacza wideo należy ustawić właściwość `visibleVideoPlayerIndex` na numer odtwarza wideo. Więcej informacji o interakcjach między tymi właściwościami a metodami i właściwościami klasy `FLVPlayback` zawierają opisy właściwości `FLVPlayback.activeVideoPlayerIndex` i `FLVPlayback.visibleVideoPlayerIndex` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

Poniższy kod ActionScript ładuje właściwość `source` w celu odtworzenia pliku wideo w domyślnym odtwarzaczu wideo, a następnie dodaje do niego punkt sygnalizacji. Gdy wystąpi zdarzenie `ready`, moduł obsługi zdarzeń otwiera drugi odtwarzacz wideo, ustawiając dla właściwości `activeVideoPlayerIndex` numer 1. Ten kod określa plik FLV i punkt sygnalizacji dla drugiego odtwarzacza wideo, a następnie ponownie ustawia domyślny odtwarzacz (0) jako aktywny odtwarzacz wideo.

**Korzystanie ze składnika FLVPlayback**

```

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

```

Aby wybrać inny plik FLV podczas odtwarzania jednego pliku, należy ustawić przełączenie w kodzie ActionScript. Punkty sygnalizacji umożliwiają interweniowanie w określonych miejscach pliku FLV za pomocą zdarzenia cuePoint. Poniższy kod tworzy detektor dla zdarzenia cuePoint i wywołuje funkcję modułu obsługi zdarzeń, która wstrzymuje aktywny odtwarzacz wideo (0), przełącza na drugi odtwarzacz (1), a następnie odtwarza jego plik FLV:

```

import fl.video.*;
// add listener for a cuePoint event
my_FLVPlayback.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
// add the handler function for the cuePoint event
function cp_listener(eventObject:MetadataEvent):void {
    // display the no. of the video player causing the event
    trace("Hit cuePoint event for player: " + eventObject.vp);
    // test for the video player and switch FLV files accordingly
    if (eventObject.vp == 0) {
        my_FLVPlayback.pause(); //pause the first FLV file
        my_FLVPlayback.activeVideoPlayerIndex = 1; // make the 2nd player active
        my_FLVPlayback.visibleVideoPlayerIndex = 1; // make the 2nd player visible
        my_FLVPlayback.play(); // begin playing the new player/FLV
    } else if (eventObject.vp == 1) {
        my_FLVPlayback.pause(); // pause the 2nd FLV
        my_FLVPlayback.activeVideoPlayerIndex = 0; // make the 1st player active
        my_FLVPlayback.visibleVideoPlayerIndex = 0; // make the 1st player visible
        my_FLVPlayback.play(); // begin playing the 1st player
    }
}
my_FLVPlayback.addEventListener(VideoEvent.COMPLETE, complete_listener);
function complete_listener(eventObject:VideoEvent):void {
    trace("Hit complete event for player: " + eventObject.vp);
    if (eventObject.vp == 0) {
        my_FLVPlayback.activeVideoPlayerIndex = 1;
        my_FLVPlayback.visibleVideoPlayerIndex = 1;
        my_FLVPlayback.play();
    } else {
        my_FLVPlayback.closeVideoPlayer(1);
    }
};

```



**Korzystanie ze składnika FLVPlayback**

W przypadku utworzenia nowego odtwarzacza wideo instancja FLVPlayback ustawia jego właściwości na wartości zgodne z wartościami domyślnego odtwarzacza wideo — z wyjątkiem właściwości `source`, `totalTime` i `isLive`, dla których instancja FLVPlayback zawsze ustawia wartości domyślne — odpowiednio: pusty ciąg znaków, 0 i `false`. Instancja ustawia właściwość `autoPlay`, która domyślnie przyjmuje wartość `true`, a dla domyślnego odtwarzacza wideo przyjmuje wartość `false`. Właściwość `cuePoints` nie ma wpływu na kolejne operacje ładowania do domyślnego odtwarzacza wideo.

Metody i właściwości, które kontrolują wielkość, położenie, wymiary, widoczność i elementy sterowania na interfejsie użytkownika są zawsze globalne, a na ich działanie nie wpływa ustawienie właściwości `activeVideoPlayerIndex`. Więcej informacji o tych metodach i właściwościach oraz skutkach ustawienia właściwości `activeVideoPlayerIndex` zawiera opis właściwości `FLVPlayback.activeVideoPlayerIndex` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#). Pozostałe właściwości i metody są określone dla odtwarzacza wideo określonego przez wartość właściwości `activeVideoPlayerIndex`.

Właściwości i metody, które kontrolują wymiary, *wchodzą w interakcje* z właściwością `visibleVideoPlayerIndex`. Więcej informacji zawiera opis właściwości `FLVPlayback.visibleVideoPlayerIndex` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

## Strumieniowanie plików FLV z serwera Flash Media Server

Wymagania dotyczące strumieniowego przesyłania plików FLV z serwera Flash Media Server mogą być różne i jest to uzależnione od tego, czy dostępna jest natywna funkcja wykrywania przepustowości przez dostawcę usługi Flash Video Streaming Service. Natywne wykrywanie przepustowości oznacza, że wykrywanie przepustowości jest wstępnie zdefiniowane w serwerze przesyłania strumieniowego i zapewnia wyższą wydajność. W celu sprawdzenia, czy natywne wykrywanie przepustowości jest dostępne należy skontaktować się z dostawcą.

Aby uzyskać dostęp do plików FLV na serwerze Flash Media Server, należy użyć adresu URL, takiego jak `rtmp://nazwa_mojego_serwera/moja_aplikacja/stream.flv`.

Podczas odtwarzania strumienia z serwera Flash Media Server należy ustawić dla właściwości `isLive` składnika FLVPlayback wartość `true`. Więcej informacji zawiera opis właściwości `FLVPlayback.isLive` w [Skorowidzu języka ActionScript 3.0 dla programu Flash Professional](#).

Więcej informacji na temat administrowania serwerem Flash Media Server, łącznie z informacjami o sposobach konfigurowania strumienia aktywnego, zawiera dokumentacja serwera Flash Media Server dostępna pod adresem [www.adobe.com/support/documentation/en/flashmediaserver/](http://www.adobe.com/support/documentation/en/flashmediaserver/).

## Natywne wykrywanie przepustowości i brak wykrywania przepustowości

Klasa `NCManagerNative` jest podklasą `NCManager`, która obsługuje natywne wykrywanie przepustowości, które jest obsługiwane przez niektórych dostawców usługi Flash Video Streaming Service. Jeśli używana jest klasa `NCManagerNative`, na serwerze Flash Media Server nie są wymagane żadne pliki specjalne. Klasa `NCManagerNative` umożliwia również połączenie z dowolną wersją serwera Flash Media Server, bez pliku `main.asc`, pod warunkiem że wykrywanie przepustowości nie jest wymagane.

W celu użycia klasy `NCManagerNative` zamiast domyślnej klasy `NCManager` należy dodać poniższe linie kodu do pierwszej klatki pliku FLA:

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

## Wykrywanie przepustowości inne niż natywne

Jeśli dostawca usługi Flash Video Streaming Service nie udostępnia natywnego wykrywania przepustowości, a wykrywanie przepustowości jest wymagane, wówczas należy dodać plik `main.asc` do aplikacji FLV serwera Flash Media Server. Plik `main.asc` można znaleźć w sieci WWW pod adresem [www.adobe.com/go/learn\\_fl\\_samples\\_pl](http://www.adobe.com/go/learn_fl_samples_pl). Znajduje się w pliku `Samples.zip`, w katalogu `Samples\ComponentsAS2\FLVPlayback`.

### W celu skonfigurowania serwera Flash Media Server dla strumieniowego przesyłania plików FLV:

- 1 Utwórz folder w folderze aplikacji Flash Media Server i nadaj mu nazwę, taką jak `moja_aplikacja`.
- 2 Skopiuj plik `main.asc` do folderu `moja_aplikacja`.
- 3 W folderze `moja_aplikacja` utwórz folder o nazwie `strumienie`.
- 4 W folderze „strumienie” utwórz folder o nazwie `_definst_`.
- 5 Umieść pliki FLV w folderze `_definst_`.

## Dostosowywanie składnika FLVPlayback

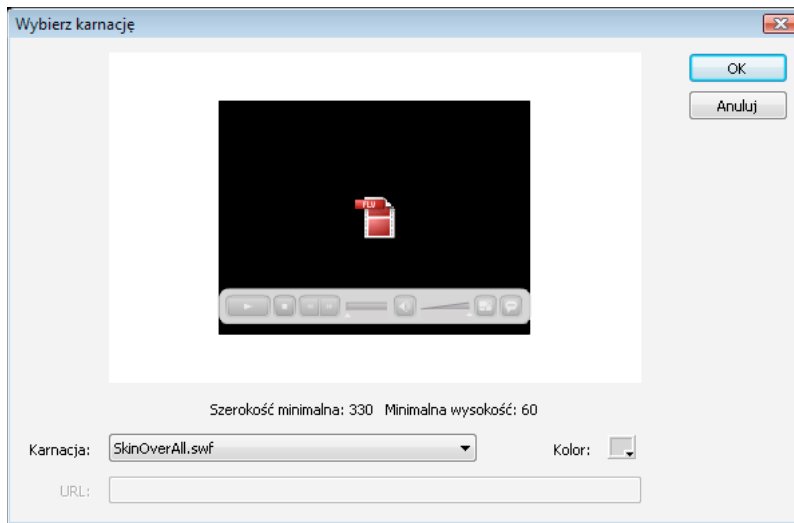
W niniejszej sekcji opisano sposób dostosowywania składnika FLVPlayback. W większości sposobów służących do dostosowywania innych składników nie jest wykorzystywany składnik FLVPlayback. W celu dostosowania składnika FLVPlayback należy stosować wyłącznie techniki opisane w niniejszej sekcji.

Istnieją następujące możliwości dostosowania składnika FLVPlayback: wybranie wstępnie zaprojektowanej karnacji, utworzenie karnacji dla poszczególnych niestandardowych składników UI FLV Playback lub utworzenie nowej karnacji. Można również użyć właściwości FLVPlayback w celu zmodyfikowania działania karnacji.

***Uwaga:** Plik SWF karnacji należy załadować na serwer sieci Web wraz z plikiem SWF aplikacji dla karnacji, która będzie działała ze składnikiem FLVPlayback.*

### Wybór wstępnie zaprojektowanej karnacji

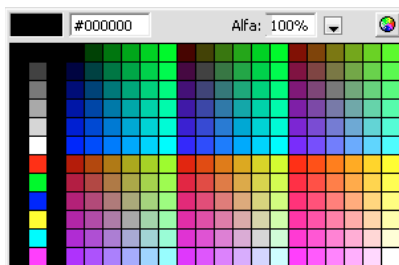
W celu wybrania karnacji dla składnika FLVPlayback kliknij komórkę `value` dla parametru `skin` w inspektorze składników. Następnie kliknij ikonę lupy, aby otworzyć poniższe okno dialogowe Wybierz karnację w którym można wybrać karnację lub wprowadzić adres URL, który będzie określał lokalizację pliku SWF karnacji.



Okno dialogowe Wybierz karnację dla składnika FLVPlayback

Karnacje widoczne w menu podręcznym Karnacja są zlokalizowane w folderze /Flash Configuration/FLVPlayback Skins/ActionScript 3.0 aplikacji. Aby uwidocznić nową karnację w tym oknie dialogowym, należy utworzyć ją i umieścić plik SWF w tym folderze. Nazwa pojawi się w menu podręcznym z rozszerzeniem .swf. Więcej informacji na temat tworzenia zestawu karnacji zawiera sekcja „[Tworzenie nowej karnacji](#)” na stronie 165.

Dla karnacji przypisanych poprzez ustawienie właściwości `skin`, poprzez ustawienie parametru `skin` podczas tworzenia lub przypisanych w języku ActionScript w środowisku wykonawczym, można przypisać wartości koloru i alfa (przezroczystości) niezależnie od tego, która karnacja została wybrana. Aby podczas tworzenia przypisać wartości koloru i alfa, należy otworzyć obszar wyboru kolorów w oknie dialogowym Wybierz karnację, co zostało przedstawione poniżej.



Selektor koloru w oknie dialogowym Wybierz karnację

W celu wybrania koloru kliknij próbkę na panelu lub wprowadź jej wartość liczbową do pola tekstowego. Aby wybrać wartość alfa, użyj suwaka lub wpisz wartość procentową do pola tekstowego alfa.

Aby przypisać wartości koloru i alfa w środowisku wykonawczym, ustaw właściwości `skinBackgroundColor` i `skinBackgroundAlpha`. Ustaw dla właściwości `skinBackgroundColor` wartość 0xRRGGBB (czerwony, zielony, niebieski). Ustaw dla właściwości `skinBackgroundAlpha` numer z zakresu od 0.0 do 1.0. W poniższym przykładzie przedstawiono ustawienie dla właściwości `skinBackgroundColor` wartości 0xFF0000 (czerwony) i wartości 0,5 dla właściwości `skinBackgroundAlpha`.

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

Wartości domyślne to ostatnie wartości, jakie wybrał użytkownik.

Jeśli dla składnika FLVPlayback wymagane jest określenie karnacji za pomocą niestandardowych składników UI FLV Playback, z menu podręcznego wybierz wartość Brak.

## Tworzenie karnacji dla poszczególnych niestandardowych składników UI FLV Playback

Niestandardowe składniki UI FLV Playback umożliwiają dostosowywanie elementów sterowania FLVPlayback w pliku FLA oraz umożliwiają wyświetlanie rezultatów na podglądzie strony WWW. Te składniki nie mogą być jednak skalowane. Klip filmowy oraz jego treść należy edytować, aby określić odpowiednią wielkość klipu. Z tego względu zwykle najlepiej jest ustawić na stole montażowym składnik FLVPlayback żądanej wielkości, a dla właściwości `scaleMode` ustawić wartość `exactFit`.

W celu rozpoczęcia przeciągnij żądane niestandardowe składniki UI FLV Playback z panelu Składniki, umieść je w żądanym miejscu na stole montażowym, a następnie nadaj im nazwy.

Te składniki mogą działać bez kodu ActionScript. Jeśli zostaną umieszczone na tej samej osi czasu i w tej samej klatce, co składnik FLVPlayback, a w składniku nie ustawiono żadnej karnacji, wówczas składnik FLVPlayback połączy się automatycznie z tymi składnikami. Jeśli na stole montażowym istnieje wiele składników FLVPlayback lub jeśli niestandardowy element sterowania i instancja FLVPlayback nie znajdują się na tej samej osi czasu, wówczas wymagane jest wybranie opcji Operacja.

Składniki umieszczone na stole montażowym można edytować w taki sam sposób, jak każdy inny symbol. Po otwarciu składników widoczne jest, że każdy z nich jest ustawiony w sposób inny niż pozostałe.

### Składniki Button

Składniki Button mają podobną strukturę. Do przycisków należą: BackButton, ForwardButton, MuteButton, PauseButton, PlayButton, PlayPauseButton i StopButton. Dla większości z nich w klatce 1 istnieje pojedynczy klip filmowy o nazwie instancji `placeholder_mc`. Jest to zwykle instancja stanu normal dla przycisku, jednak nie musi tak być. Na klatce 2 istnieją cztery klipy filmowe na stole montażowym dla każdego stanu wyświetlania: normal, over, down i disabled. (W czasie wykonywania składnik nigdy nie przechodzi do klatki 2; te klipy filmowe są dostępne, ponieważ ułatwiają edycję i mogą być ładowane do pliku SWF bez konieczności zaznaczania pola wyboru Eksport w pierwszej klatce w oknie dialogowym Właściwości symbolu. Konieczne jest jednak wybranie opcji Eksportuj do ActionScript).

W celu określenia karnacji dla przycisku należy przeprowadzić edycję każdego z tych klipów filmowych. Możliwa jest zmiana wielkości, a także wyglądu.

Część kodu ActionScript zwykle znajduje się w klatce 1. Zmiana tego skryptu nie powinna być konieczna. Skrypt zatrzymuje głośnicę odtwarzania na klatce 1 i określa klipy filmowe, które będą używane dla określonych stanów.

### Przyciski PlayPauseButton, MuteButton, FullScreenButton i CaptionButton

Przyciski PlayPauseButton, MuteButton, FullScreenButton i CaptionButton są skonfigurowane inaczej niż inne przyciski; dla nich istnieje tylko jedna klatka z dwoma warstwami i bez skryptu. Ta klatka — w przypadku składnika PlayPauseButton (przyciski Odtwórz i Wstrzymaj) — zawiera dwa przyciski: jeden powyżej drugiego; w przypadku składnika MuteButton są to dwa przyciski: Wycisz i Włącz dźwięk; w przypadku składnika FullScreenButton są to dwa przyciski: przycisk włączenia i wyłączenia trybu pełnoekranowego; w przypadku składnika CaptionButton są to przyciski włączenia i wyłączenia napisów. W celu ustawienia karnacji dla tych przycisków należy zdefiniować karnację dla każdego z dwóch przycisków wewnętrznych, co zostało opisane w sekcji „[Tworzenie karnacji dla poszczególnych niestandardowych składników UI FLV Playback](#)” na stronie 160; wykonywanie innych czynności nie jest wymagane.

Składnik CaptionButton jest przeznaczony dla składnika FLVPlaybackCaptioning i musi być dołączony do tego składnika, a nie do składnika FLVPlayback.

### Przyciski BackButton i ForwardButton

Przyciski BackButton i ForwardButton również ustawione są w sposób inny niż inne przyciski. W klatce 2 istnieją dla nich dodatkowe klipy filmowe, które mogą być używane jako klatka wokół jednego lub obydwu tych przycisków. Te klipy filmowe nie są wymagane i nie udostępniają żadnych specjalnych funkcji; zostały wprowadzone wyłącznie w celu ułatwienia korzystania z programu. W celu użycia klipu wystarczy przeciągnąć go na stół montażowy z panelu Biblioteka, a następnie umieścić je w żądanym miejscu. Jeśli klipy nie są wymagane, nie należy ich używać lub należy je usunąć z panelu Biblioteka.

Większość dostępnych przycisków jest oparta na wspólnym zestawie klipów filmowych, dlatego możliwa jest zmiana wyglądu wszystkich przycisków jednocześnie. Użytkownik może wykorzystać tę zależność lub zastąpić te wspólne klipy i zdefiniować inny wygląd dla każdego przycisku.

### Składnik BufferingBar

Składnik paska buforowania działa bardzo prosto: zawiera animację, która zostaje wyświetlona, gdy składnik przechodzi do stanu buforowania i nie jest wymagany specjalny kod ActionScript w celu skonfigurowania tego składnika. Domyślnie jest to pasek z segmentami, który jest przesuwany z lewej na prawą stronę. Dla paska zdefiniowana jest prostokątna maska, dzięki której uzyskuje on wygląd obrotowego słupa z paskiem, ale w tej konfiguracji nie ma niczego szczególnego.

Paski buforowania w plikach SWF karnacji korzystają z 9-warstwowej skali, ponieważ muszą być skalowane w środowisku wykonawczym — niestandardowy składnik UI BufferingBar FLV nie korzysta i *nie może* korzystać z 9-plastrowej skali, ponieważ zawiera on zagnieżdżone klipy filmowe. Jeśli wymagane jest poszerzenie i zwiększenie wysokości składnika BufferingBar, wówczas zamiast skalować go należy zmienić jego zawartość.

### Składniki SeekBar i VolumeBar

Składniki SeekBar i VolumeBar są do siebie podobne, ale mają inne funkcje. Każdy z nich zawiera uchwyty, korzysta z tych samych mechanizmów śledzenia uchwytu i obsługuje klipy zagnieżdżone w celu śledzenia postępu i stopnia ukończenia.

Istnieje wiele miejsc, w których kod ActionScript w składniku FLVPlayback przyjmuje, że punkt odniesienia (zwany również *punktem wyjściowym* lub *punktem zero*) na składniku SeekBar lub VolumeBar, znajduje się w lewym górnym rogu, dlatego ważne jest zachowanie takiej konwencji. W przeciwnym wypadku mogą pojawić się problemy z uchwytami oraz ze śledzeniem postępu i stopnia ukończenia klipów filmowych.

Paski wyszukiwania w plikach SWF karnacji korzystają z 9-warstwowej skali, ponieważ muszą być skalowane w środowisku wykonawczym — niestandardowy składnik UI SeekBar FLV nie korzysta i *nie może* korzystać z 9-plastrowej skali, ponieważ zawiera on zagnieżdżone klipy filmowe. Jeśli wymagane jest poszerzenie i zwiększenie wysokości składnika SeekBar, wówczas zamiast skalować go należy zmienić jego zawartość.

### Uchwyt

Instancja klipu filmowego uchwytu znajduje się w klatce 2. Ten składnik — podobnie jak składniki BackButton i ForwardButton — nigdy nie przechodzi do klatki 2; te klipy filmowe są dostępne, ponieważ ułatwiają edycję i mogą być ładowane do pliku SWF bez konieczności zaznaczania pola wyboru Eksport w pierwszej klatce w oknie dialogowym Właściwości symbolu. Konieczne jest jednak wybranie opcji Eksportuj do ActionScript.

Widoczne jest, że klip filmowy uchwytu zawiera prostokąt w tle, dla którego wartość alfa wynosi 0. Ten prostokąt zwiększa obszar docelowy uchwytu, dzięki czemu łatwiej jest chwycić uchwyt bez zmiany jego wyglądu (podobnie jak w przypadku obszaru klikania przycisku). Uchwyt jest tworzony w sposób dynamiczny w środowisku wykonawczym, dlatego musi być klipem filmowym, a nie przyciskiem. Prostokąt z wartością alfa ustawioną na 0 nie jest wymagany z innych powodów i zwykle wewnątrz uchwytu można wypełnić dowolnym obrazem. Najlepiej jest jednak, gdy środek punktu odniesienia znajduje się na tym samym poziomie, co linia środkowa klipu filmowego uchwytu.

**Korzystanie ze składnika FLVPlayback**

Poniższy kod ActionScript znajduje się w klatce 1 składnika SeekBar i umożliwia zarządzanie uchwytem:

```
stop();  
handleLinkageID = "SeekBarHandle";  
handleLeftMargin = 2;  
handleRightMargin = 2;  
handleY = 11;
```

Wywołanie funkcji `stop()` jest niezbędne z powodu treści klatki 2.

Druga linia określa symbol, który będzie używany jako uchwyt i w przypadku prostej edycji instancji klipu filmowego uchwytu w klatce 2 zmiana tego symbolu nie powinna być konieczna. W środowisku wykonawczym składnik FLVPlayback tworzy instancję określonego klipu filmowego na stole montażowym jako element równorzędny instancji składnika Bar, co oznacza, że istnieje dla nich ten sam nadrzędny klip filmowy. Jeśli więc pasek znajduje się na poziomie głównym, uchwyt również musi być na poziomie głównym.

Zmienna `handleLeftMargin` określa początkowe położenie uchwytu (0%), a zmienna `handleRightMargin` określa jego położenie na końcu (100%). Liczby określają przesunięcie od lewego i prawego końca elementu sterowania paskiem. Liczby dodatnie określają granice na pasku, a ujemne określają granice poza paskiem. Te przesunięcia określają miejsca, w których uchwyt może się znaleźć w zależności od punktu odniesienia. Jeśli punkt odniesienia znajdzie się w środku uchwytu, prawa i lewa strona uchwytu znajdzie się za marginesami. Punkt odniesienia paska wyszukiwania musi znajdować się w lewym górnym rogu treści — tylko wówczas działa poprawnie.

Zmienna `handleY` określa położenie *y* uchwytu względem instancji paska. Jest to uzależnione od punktów odniesienia każdego klipu filmowego. Punkt odniesienia na przykładowym uchwycie znajduje się na końcu trójkąta — jest widoczny względem widocznej części bez względu na niewidoczny prostokąt docelowy. Ponadto punkt odniesienia klipu filmowego musi znajdować się w lewym górnym rogu treści — tylko wówczas działa poprawnie.

Dlatego jeśli przy uwzględnieniu tych granic element sterujący zostanie ustawiony w pozycji (100, 100) i ma 100 pikseli szerokości, wówczas uchwyt może mieć wielkość od 102 do 198 w poziomie i pozostawać na wysokości 111. Jeśli wartości `handleLeftMargin` i `handleRightMargin` zostaną zmienione na -2, a wartość `handleY` na -11, wówczas uchwyt może mieć wielkość od 98 do 202 w poziomie i pozostawać na wysokości 89.

**Klipy filmowe postępu i stopnia ukończenia**

Składnik SeekBar zawiera klip filmowy *progress*, a składnik VolumeBar zawiera klip filmowy *fullness*, ale w praktyce składniki SeekBar i VolumeBar mogą korzystać z jednego, żadnego lub obydwu tych klipów. Strukturalnie wyglądają tak samo i zachowują się podobnie, ale śledzą inne wartości. Klip filmowy postępu jest wypełniany podczas pobierania pliku FLV (jest to użyteczne jedynie dla pobierania z użyciem protokołu HTTP, ponieważ dla strumieniowania z FMS plik jest zawsze pełny), a klip filmowy stopnia ukończenia wypełniany jest podczas przesuwania się uchwytu od lewej do prawej.

Składnik FLVPlayback znajduje instancje tych klipów filmowych, wyszukując konkretnych nazw instancji, dlatego instancja klipu filmowego postępu musi mieć pasek klipu taki, jak jej element nadrzędny i nazwę instancji `progress_mc`. Instancja klipu filmowego stopnia ukończenia musi mieć nazwę instancji `fullness_mc`.

Klipy filmowe postępu i stopnia ukończenia można ustawić tak, aby klip filmowy `fill_mc` był w nich zagnieżdżony lub nie był zagnieżdżany. Klip filmowy VolumeBar `fullness_mc` wyświetla metodę z klipem filmowym `fill_mc`, a klip SeekBar `progress_mc` wyświetla metodę *bez* klipu `fill_mc`.

Metoda z zagnieżdżonym klipem filmowym `fill_mc` jest użyteczna, gdy konieczne jest wypełnienie, którego nie można przeskalać bez zniekształcenia wyglądu.

W klipie filmowym VolumeBar fullness\_mc zagnieżdżona instancja klipu fill\_mc jest maskowana. Maskę można nałożyć podczas tworzenia klipu filmowego lub zostanie ona utworzona dynamicznie w czasie wykonania. Jeśli klip maskowany będzie klipem filmowym, instancji należy nadać nazwę **mask\_mc** i ustawić ją w celu wyświetlania fill\_mc tak, jakby osiągnął wartość 100%. Jeśli fill\_mc nie będzie maskowany, dynamicznie utworzona maska będzie prostokątna o rozmiarze takim samym jak fill\_mc dla wartości 100%.

Klip filmowy fill\_mc odsłaniany jest z maską na dwa sposoby w zależności od tego, czy właściwość fill\_mc.slideReveal ma wartość true czy false.

Jeśli fill\_mc.slideReveal ma wartość true, klip fill\_mc zostanie przesunięty od lewej do prawej w celu uwidocznienia go przez maskę. Dla wartości 0% jest to cała lewa strona, dlatego żaden z klipów nie jest widoczny przez maskę. Wraz ze wzrostem wartości procentowej, klip przesuwany jest w prawą stronę, aż do osiągnięcia wartości 100%, a następnie powraca do miejsca na stole montażowym, w którym został utworzony.

Jeśli właściwość fill\_mc.slideReveal ma wartość false lub jest niezdefiniowana (zachowanie domyślne), rozmiar maski zostanie zmieniony w celu odsłonięcia większego fragmentu klipu fill\_mc. Dla wartości 0% maska jest skalowana do wartości 0.5 w poziomie i ze wzrostem wartości procentowej, zwiększa się parametr scaleX aż do 100%; odsłonięty zostanie wówczas cały klip fill\_mc. Ustawienie właściwości scaleX = 100 nie jest niezbędne, ponieważ klip mask\_mc mógł zostać przeskalowany w momencie utworzenia.

Metoda bez klipu fill\_mc jest prostsza niż metoda z klipem fill\_mc, ale powoduje zniekształcenie wypełnienia w poziomie. Aby uniknąć zniekształcenia, należy użyć klipu fill\_mc. Proces ten ilustruje klip progress\_mc składnika SeekBar.

Klip filmowy postępu lub stopnia ukończenia skalowany jest w poziomie w oparciu o wartość procentową. Dla wartości 0% właściwość scaleX instancji ustawiona jest na 0, co powoduje, że jest ona niewidoczna. Wraz ze wzrostem wartości procentowej właściwość scaleX jest uzgadniana, aż do osiągnięcia wartości 100%, wówczas klip ma rozmiar taki, jaki w chwili utworzenia. Ponownie, ustawienie właściwości scaleX = 100 nie jest niezbędne, ponieważ instancja klipu mask\_mc mogłaby zostać przeskalowana w chwili utworzenia.

## Łączenie niestandardowych składników UI FLV Playback

Jeśli niestandardowe składniki UI zostaną umieszczone na tej samej osi czasu i w tej samej klatce, co składnik FLVPlayback, a w składniku nie ustawiono właściwości skin, wówczas składnik FLVPlayback automatycznie połączy się z nimi bez konieczności użycia kodu ActionScript.

Jeśli na stole montażowym istnieje wiele składników FLVPlayback lub jeśli niestandardowy element sterowania i FLVPlayback nie znajdują się na tej samej osi czasu, należy wówczas napisać kod ActionScript w celu połączenia niestandardowych składników UI z instancją składnika FLVPlayback. Najpierw należy przypisać nazwę do instancji FLVPlayback, a następnie użyć kodu ActionScript, aby przypisać instancje niestandardowego składnika UI FLV Playback do odpowiednich właściwości FLVPlayback. W poniższym przykładzie instancją FLVPlayback jest my\_FLVPlybk, nazwy właściwości FLVPlayback są poprzedzone kropkami (.), a instancje elementu sterującego niestandardowego UI FLV Playback znajdują się po prawej stronie znaków równości (=):

```
//FLVPlayback instance = my_FLVPlybk
my_FLVPlybk.playButton = playbtn; // set playButton prop. to playbtn, etc.
my_FLVPlybk.pauseButton = pausebtn;
my_FLVPlybk.playPauseButton = playpausebtn;
my_FLVPlybk.stopButton = stopbtn;
my_FLVPlybk.muteButton = mutebtn;
my_FLVPlybk.backButton = backbtn;
my_FLVPlybk.forwardButton = forbtn;
my_FLVPlybk.volumeBar = volbar;
my_FLVPlybk.seekBar = seekbar;
my_FLVPlybk.bufferingBar = bufbar;
```

Wykonanie poniższych kroków powoduje utworzenie niestandardowych elementów sterujących StopButton, PlayPauseButton, MuteButton oraz SeekBar:

- 1 Przeciągnij składnik FLVPlayback na stół montażowy i nadaj instancji nazwę **my\_FLVPlybk**.
- 2 Ustaw parametr `source` za pomocą Inspektora składników na <http://www.helpexamples.com/flash/video/cuepoints.flv>.
- 3 Ustaw parametr `Karnacja` na wartość Brak.
- 4 Przeciągnij na stół montażowy składniki StopButton, PlayPauseButton, oraz MuteButton i umieść je nad instancją FLVPlayback, układając je pionowo w stos po lewej stronie. Nadaj każdemu z przycisków nazwę instancji w Inspektorze właściwości (np. **my\_stopbtn**, **my\_plypausbbtn** i **my\_mutebtt**).
- 5 W panelu Biblioteka otwórz folder FLVPlayback Skins, a następnie otwórz folder SquareButton.
- 6 Wybierz klip filmowy SquareBgDown i kliknij go dwukrotnie w celu otwarcia na stole montażowym.
- 7 Kliknij prawym przyciskiem (Windows) lub kliknij przytrzymując klawisz Control (Macintosh), wybierz z menu opcję Zaznacz wszystko i usuń symbol.
- 8 Wybierz narzędzie Ował, narysuj ował w tym samym miejscu i ustaw wypełnienie na niebieskie (**#0033FF**).
- 9 W Inspektorze właściwości ustaw szerokość (Sz:) na **40** i wysokość (W:) na **20**. Ustaw współrzędną x (X:) na **0.0** i współrzędną y (Y:) na **0.0**.
- 10 Powtórz kroki od 6 do 8 dla SquareBgNormal, zmieniając wypełnienie na żółte (**#FFFF00**).
- 11 Powtórz kroki od 6 do 8 dla SquareBgOver, zmieniając wypełnienie na zielone (**#006600**).
- 12 Dokonaj edycji klipów filmowych dla różnych ikon symboli w przyciskach (PauseIcon, PlayIcon, MuteOnIcon, MuteOffIcon i StopIcon). Pliki te można znaleźć w panelu Biblioteka w folderach FLV Playback Skins/Label Button/Assets, gdzie *Label* jest nazwą przycisku np. Odtwórz, Wstrzymaj itd. Dla każdego składnika wykonaj następujące czynności:
  - a Wybierz opcję Zaznacz wszystko.
  - b Zmień kolor na czerwony (**#FF0000**).
  - c Przeskaluj składnik do 300%.
  - d Zmień współrzędną X: położenie treści na **7.0**, aby zmienić ułożenie poziome ikony dla każdego stanu przycisku.

*Uwaga: Poprzez zmianę położenia w ten sposób użytkownik uniknie otwierania każdego stanu dla przycisku i przesuwania instancji klipu filmowego ikony.*
- 13 Kliknij niebieską strzałkę Wstecz powyżej osi czasu, aby powrócić do sceny nr 1, klatki nr 1.
- 14 Przeciągnij na stół montażowy składnik SeekBar i umieść go w prawym dolnym rogu instancji FLVPlayback.
- 15 W panelu Biblioteka kliknij dwukrotnie składnik SeekBar, aby otworzyć go na stole montażowym.
- 16 Przeskaluj go do wartości 400%.
- 17 Zaznacz kontur i ustaw kolor na czerwony (**#FF0000**).
- 18 Kliknij dwukrotnie SeekBarProgress w folderze FLVPlayback Skins/Seek Bar i ustaw kolor na żółty (**#FFFF00**).
- 19 Kliknij dwukrotnie SeekBarHandle w folderze FLVPlayback Skins/Seek Bar i ustaw kolor na czerwony (**#FF0000**).
- 20 Kliknij niebieską strzałkę Wstecz powyżej osi czasu, aby powrócić do sceny nr 1, klatki nr 1.
- 21 Zaznacz instancję SeekBar na stole montażowym i nadaj instancji nazwę **my\_seekbar**.



**Korzystanie ze składnika FLVPlayback**

**22** W panelu Operacje do klatki nr 1 na osi czasu dodaj instrukcję importowania dla klas wideo i przypisz nazwy dla przycisku i paska wyszukiwania do odpowiednich właściwości FLVPlayback tak, jak przedstawiono to w poniższym przykładzie:

```
import fl.video.*;
my_FLVPlybk.stopButton = my_stopbbtn;
my_FLVPlybk.playPauseButton = my_plypausbbtn;
my_FLVPlybk.muteButton = my_mutebbtn;
my_FLVPlybk.seekBar = my_seekbar;
```

**23** Naciśnij klawisze Control+Enter, aby przetestować film.

## Tworzenie nowej karnacji

Najlepszym sposobem na utworzenie pliku SWF karnacji jest skopiowanie jednego z plików karnacji dostarczonego z programem Flash i użycie go jako punktu wyjściowego. Pliki FLA dla tych karnacji można znaleźć w folderze Configuration/FLVPlayback Skins/FLA/ActionScript 3.0/ w aplikacji Flash. Aby udostępnić ukończony plik SWF karnacji jako opcję w oknie dialogowym Wybierz karnację, należy umieścić ją w folderze Configuration/FLVPlayback Skins/ActionScript 3.0 aplikacji Flash lub w folderze lokalnym użytkownika Configuration/FLVPlayback Skins/ActionScript 3.0.

Ponieważ kolor karnacji można ustawić niezależnie od wybieranej karnacji, edycja pliku FLA w celu modyfikacji koloru nie jest konieczna. Jeśli utworzona zostanie karnacja z konkretnym kolorem, który ma być nieedytowalny w oknie dialogowym Wybierz karnację należy w kodzie ActionScript pliku FLA ustawić `this.border_mc.colorMe = false;`. Informacje o ustawianiu koloru karnacji zawiera sekcja „[Wybór wstępnie zaprojektowanej karnacji](#)” na stronie 158.

Podczas przeglądania zainstalowanych plików FLA karnacji Flash, może wydawać się, że pewne rzeczy na stole montażowym są niepotrzebne, ale wiele z nich zostało umieszczonych na liniach pomocniczych. Za pomocą podglądu aktywnego skalowania 9-plastowego możliwy jest szybki podgląd tego, co rzeczywiście zostanie wyświetlone w pliku SWF.

Następujące sekcje zawierają bardziej złożone dopasowania i zmiany klipów filmowych SeekBar, BufferingBar i VolumeBar.

## Korzystanie z układu karnacji

Po otwarciu pliku FLA karnacji Flash na głównej osi czasu ułożone będą klipy filmowe karnacji. Klipy te wraz z kodem ActionScript tej samej klatki definiują sposób, w jaki elementy sterujące układane będą w czasie wykonywania.

Mimo że warstwa Układ wygląda podobnie, jak karnacja w czasie wykonywania, treść tego układu w czasie wykonywania jest niewidoczna. Używany jest on jedynie do obliczania miejsc, w których rozmieszczone zostaną elementy sterujące. Pozostałe elementy sterujące na stole montażowym używane są w czasie wykonywania.

W warstwie Układ znajduje się element zastępczy o nazwie video\_mc dla składnika FLVPlayback. Wszystkie pozostałe elementy sterujące układane są względem video\_mc. Jeśli użytkownik rozpocznie pracę z jednym z plików FLA programu Flash i zmieni rozmiar elementów sterujących, przesunięcie klipów zastępczych umożliwi zwykle naprawienie układu.

Każdy z klipów elementów zastępczych ma konkretną nazwę instancji. Nazwami elementów zastępczych są: playpause\_mc, play\_mc, pause\_mc, stop\_mc, captionToggle\_mc, fullScreenToggle\_mc, back\_mc, bufferingBar\_mc, bufferingBarFill\_mc, seekBar\_mc, seekBarHandle\_mc, seekBarProgress\_mc, volumeMute\_mc, volumeBar\_mc oraz volumeBarHandle\_mc. Fragment, który jest ponownie kolorowany podczas wybierania koloru dla karnacji, nosi nazwę border\_mc.

Nie jest ważne, który klip używany jest dla elementu sterującego. Zazwyczaj do przycisków używany jest klip stanu normalnego. Jedynie dla wygody dla pozostałych elementów sterujących używany jest klip danego elementu. Najważniejszą rzeczą są położenia  $x$  (poziome) i  $y$  (pionowe) oraz wysokość i szerokość elementu zastępczego.

Poza standardowymi elementami sterującymi użytkownik może zdefiniować dowolną ilość dodatkowych klipów. Jedynym wymaganym dla klipów jest, aby w oknie dialogowym Powiązanie dla ich symboli biblioteki zaznaczona była opcja Eksportuj do ActionScript. Niestandardowe klipy warstwy Układ mogą mieć dowolną nazwę instancji, poza zarezerwowanymi nazwami instancji wymienionymi powyżej. Nazwa instancji wymagana jest tylko do ustawiania dla klipów kodu ActionScript w celu określenia układu.

Klip `border_mc` jest klipem specjalnym. Jeśli właściwość `FLVPlayback.skinAutoHide` zostanie ustawiona na wartość `true`, karnacja zostanie wyświetlona, gdy mysz umieszczona zostanie nad klipem `border_mc`. Dla karnacji istotne jest wyświetlanie poza obwiednią odtwarzacza wideo. Informacje o właściwości `skinAutoHide` zawiera sekcja „[Modyfikowanie zachowania karnacji](#)” na stronie 169.

W plikach FLA programu Flash klip `border_mc` używany jest dla efektu chromowania i krawędzi wokół przycisków Do przodu i Wstecz.

Klip `border_mc` jest także częścią karnacji, której kanał alfa i kolor zmienione zostały przez właściwości `skinBackgroundAlpha` i `skinBackgroundColor`. Aby umożliwić dostosowywanie koloru oraz kanału alfa, do kodu ActionScript należy dołączyć następujący fragment:

```
border_mc.colorMe = true;
```

### Kod ActionScript i układ karnacji

Poniższy kod ActionScript dotyczy wszystkich elementów sterujących. Niektóre elementy sterujące zawierają konkretny kod ActionScript, który określa dodatkowe zachowanie i który wyjaśniony jest w poświęconej jemu sekcji.

Wstęp do języka ActionScript jest dużą sekcją, w której określone zostały nazwy klas dla każdego stanu wszystkich składników. Wszystkie nazwy klas znajdują się w pliku `SkinOverAll fla`. Dla przycisków Wstrzymaj i Odtwórz przykładowy kod wyglądu następująco:

```
this.pauseButtonDisabledState = "fl.video.skin.PauseButtonDisabled";  
this.pauseButtonDownState = "fl.video.skin.PauseButtonDown";  
this.pauseButtonNormalState = "fl.video.skin.PauseButtonNormal";  
this.pauseButtonOverState = "fl.video.skin.PauseButtonOver";  
this.playButtonDisabledState = "fl.video.skin.PlayButtonDisabled";  
this.playButtonDownState = "fl.video.skin.PlayButtonDown";  
this.playButtonNormalState = "fl.video.skin.PlayButtonNormal";  
this.playButtonOverState = "fl.video.skin.PlayButtonOver";
```

Nazwy klas nie mają rzeczywistych plików klas zewnętrznych; są one określone w oknie dialogowym Połączenie dla wszystkich klipów filmowych w bibliotece.

W składniku w języku ActionScript 2.0 były klipami filmowymi stołu montażowego używanymi w czasie wykonywania. W składniku w języku ActionScript 3.0 dla wygody edytowania klipy filmowe pozostają w pliku FLA. W chwili obecnej wszystkie klipy znajdują się na liniach pomocniczych i nie są eksportowane. Wszystkie zasoby karnacji w bibliotece ustawione są do eksportu w pierwszej klatce i tworzone są dynamicznie wraz z kodem, jak w poniższym przykładzie:

```
new fl.video.skin.PauseButtonDisabled();
```

Ta sekcja zawiera kod ActionScript, w którym definiowana jest minimalna szerokość i wysokość karnacji. Wartości te wyświetlane są w oknie dialogowym Wybierz karnację i używane są w czasie wykonywania w celu zapobiegania skalowaniu karnacji poniżej jej minimalnego rozmiaru. Jeśli minimalny rozmiar nie ma być określany, należy pozostawić go jako niezdefiniowany lub określić jako mniejszy bądź równy zero.

**Korzystanie ze składnika FLVPlayback**

```
// minimum width and height of video recommended to use this skin,
// leave as undefined or <= 0 if there is no minimum
this.minWidth = 270;
this.minHeight = 60;
```

Każdy element zastępczy może mieć następujące, zastosowane do siebie właściwości:

Właściwość	Opis
anchorLeft	Boolean. Ustawia położenie elementu sterowania względem lewej strony instancji FLVPlayback. Wartością domyślną jest <code>true</code> chyba, że właściwość <code>anchorRight</code> została jawnie ustawiona na wartość <code>true</code> , wówczas wartością domyślną jest <code>false</code> .
anchorRight	Boolean. Ustawia położenie elementu sterowania względem prawej strony instancji FLVPlayback. Wartością domyślną jest <code>false</code> .
anchorBottom	Boolean. Ustawia położenie elementu sterowania względem dolnej części instancji FLVPlayback. Wartością domyślną jest <code>true</code> chyba, że właściwość <code>anchorTop</code> została jawnie ustawiona na wartość <code>true</code> , wówczas wartością domyślną jest <code>false</code> .
anchorTop	Boolean. Ustawia położenie elementu sterowania względem górnej części instancji FLVPlayback. Wartością domyślną jest <code>false</code> .

Jeśli obie właściwości `anchorLeft` i `anchorRight` mają wartość `true`, w czasie wykonywania element sterujący skalowany jest poziomo. Jeśli obie właściwości `anchorTop` i `anchorBottom` mają wartość `true`, w czasie wykonywania element sterujący skalowany jest pionowo.

Aby zobaczyć efekt działania właściwości, należy zobaczyć, w jaki sposób właściwości używane są w karnacjach Flash. Elementy sterujące `BufferingBar` i `SeekBar` są jedynymi elementami skalowanymi; układane są jeden na drugim i mają właściwości `anchorLeft` i `anchorRight` ustawione na wartość `true`. Wszystkie elementy sterujące na lewo od `BufferingBar` i `SeekBar` mają właściwość `anchorLeft` ustawioną na wartość `true`, a wszystkie elementy sterujące po prawej stronie mają `anchorRight` ustawioną na wartość `true`. Wszystkie elementy sterujące mają właściwość `anchorBottom` ustawioną na wartość `true`.

Aby utworzyć karnację powyżej, a nie poniżej, miejsca, w którym umieszczone są elementy sterujące, można dokonać edycji na warstwie Układ. Należy jedynie przesunąć elementy sterujące do góry względem klipu `video_mc`, i ustawić właściwość `anchorTop` na wartość `true` dla wszystkich elementów sterujących.

**Pasek buforowania**

Pasek buforowania zawiera dwa klipy filmowe: `bufferingBar_mc` i `bufferingBarFill_mc`. Położenie każdego klipu na stole montażowym względem innego jest istotne ze względu na pozycjonowanie względne. Pasek buforowania używa dwóch oddzielnych klipów, ponieważ składnik skaluje klip `bufferingBar_mc`, ale nie skaluje `bufferingBarFill_mc`.

W klipie `bufferingBar_mc` stosowane jest skalowanie 9-plastrowe, dlatego obwódka nie zostanie podczas skalowania zniekształcona. Klip `bufferingBarFill_mc` jest bardzo szeroki, dlatego zawsze będzie dostatecznie szeroki bez potrzeby skalowania. Jest automatycznie maskowany w czasie wykonania w celu wyświetlenia tylko fragmentu powyżej rozciągniętego klipu `bufferingBar_mc`. Domyślnie dokładne wymiary maski zachowają równe marginesy po lewej i prawej stronie w klipie `bufferingBar_mc`, w oparciu o różnicę między położeniami  $x$  (poziomymi) klipu `bufferingBar_mc` i `bufferingBarFill_mc`. Pozycjonowanie można dostosować za pomocą kodu ActionScript.

Jeśli pasek buforowania nie potrzebuje skalowania lub nie używa skalowania 9-plastrowego, można go ustawić tak, jak składnik `BufferingBar` niestandardowego UI FLV Playback. Więcej informacji zawiera „[Składnik BufferingBar](#)” na stronie 161.

Pasek buforowania zawiera następującą dodatkową właściwość:

Właściwość	Opis
fill_mc:MovieClip	Określa nazwę instancji wypełnienia paska buforowania. Wartością domyślną jest bufferingBarFill_mc.

## Pasek wyszukiwania i pasek głośności

Pasek wyszukiwania również zawiera dwa klipy filmowe : seekBar\_mc i seekBarProgress\_mc. Położenie każdego klipu w warstwie Układ względem innego jest istotne ze względu na pozycjonowanie względne. Mimo że oba klipy są skalowane, klip seekBarProgress\_mc nie może być zagnieżdżony w klipie seekBar\_mc, ponieważ seekBar\_mc używa skalowania 9-plastrowego, które nie działa poprawnie z zagnieżdżonymi klipami filmowymi.

W klipie seekBar\_mc stosowane jest skalowanie 9-plastrowe, dlatego obwiednia nie zostanie podczas skalowania zniekształcona. Klip seekBarProgress\_mc jest również skalowany, ale jest on zniekształcany. Nie jest w nim używane skalowanie 9-plastrowe, ponieważ jest to wypełnienie, które wygląda dobrze podczas zniekształcania.

Klip seekBarProgress\_mc działa bez klipu fill\_mc, w podobny sposób, w jaki klip progress\_mc działa w niestandardowych składnikach UI FLV Playback. Innymi słowy nie jest maskowany i skalowany w poziomie. Dokładne wymiary klipu seekBarProgress\_mc w 100% zdefiniowane zostały przez lewe i prawe marginesy w klipie seekBarProgress\_mc. Domyślnie wymiary te są równe i oparte na różnicy między położeniami  $x$  (poziomymi) klipów seekBar\_mc i seekBarProgress\_mc. Wymiary można dostosować w kodzie ActionScript klipu filmowego paska wyszukiwania tak, jak zostało to przedstawione w poniższym przykładzie:

```
this.seekBar_mc.progressLeftMargin = 2;
this.seekBar_mc.progressRightMargin = 2;
this.seekBar_mc.progressY = 11;
this.seekBar_mc.fullnessLeftMargin = 2;
this.seekBar_mc.fullnessRightMargin = 2;
this.seekBar_mc.fullnessY = 11;
```

Kod ten można wstawić na osi czasu klipu filmowego składnika lub razem z innym kodem ActionScript na głównej osi czasu. Jeśli dostosowywanie przeprowadzane będzie za pomocą kodu, zamiast modyfikacji układu, wypełnienie nie musi znajdować się na stole montażowym. Wypełnienie musi znajdować się w bibliotece ustawione do eksportu do ActionScript w klatce nr 1 z poprawną nazwą klasy.

Tak jak w składniku SeekBar niestandardowego UI FLV Playback dla paska wyszukiwania możliwe jest utworzenie klipu filmowego stopnia ukończenia. Jeśli pasek wyszukiwania nie potrzebuje skalowania lub jest skalowany, ale nie używa skalowania 9-plastrowego, klip progress\_mc lub fullness\_mc można ustawić za pomocą dowolnej metody użytej dla niestandardowych składników UI FLV Playback. Więcej informacji zawiera sekcja .

Ponieważ pasek głośności w karnacjach Flash nie jest skalowany, zbudowany jest on w ten sam sposób, co składnik VolumeBar niestandardowego UI FLV Playback. Więcej informacji zawiera sekcja „Składniki SeekBar i VolumeBar” na stronie 161. Wyjątkiem jest inna implementacja uchwytu.

## Uchwyty Seekbar i VolumeBar

Uchwyty SeekBar i VolumeBar umieszczane są na warstwie Układ obok paska. Domyślnie lewy i prawy margines oraz wartości na osi  $y$  uchwytu ustawiane są na podstawie ich położenia względem klipu filmowego paska. Lewy margines ustawiony jest jako różnica między położeniem  $x$  (poziomym) uchwytu, a położeniem  $x$  (poziomym) paska, natomiast prawy margines jest równy lewemu. Wartości te można dostosować z użyciem kodu ActionScript w klipie SeekBar lub VolumeBar. Poniższy przykład jest tym samym kodem ActionScript, którego użyto z niestandardowymi składnikami UI FLV Playback:

```
this.seekBar_mc.handleLeftMargin = 2;
this.seekBar_mc.handleRightMargin = 2;
this.seekBar_mc.handleY = 11;
```

**Korzystanie ze składnika FLVPlayback**

Kod ten można wstawić na osi czasu klipu filmowego składnika lub razem z innym kodem ActionScript na głównej osi czasu. Jeśli dostosowywanie przeprowadzane będzie za pomocą kodu, zamiast modyfikacji układu, uchwyt nie musi znajdować się na stole montażowym. Wypełnienie musi znajdować się w bibliotece ustawione do eksportu do ActionScript w klatce nr 1 z poprawną nazwą klasy.

Poza tymi właściwościami uchwyty są prostymi klipami filmowymi ustawionymi w ten sam sposób, co w niestandardowych składnikach UI FLV Playback. Oba mają prostokątne tła z właściwością `alpha` ustawioną na 0. Są one wyświetlane jedynie w celu powiększenia obszaru klikania i nie są wymagane.

**Klipy tła i pierwszego planu**

Klipy filmowe `chrome_mc` i `forwardBackBorder_mc` są implementowane jako klipy tła.

Spośród klipów filmowych `ForwardBackBorder`, `ForwardBorder` i `BackBorder` na stole montażowym oraz zastępczych przycisków `Do przodu` i `Wstecz` jedynym, który *nie* znajduje się na linii pomocniczej, jest `ForwardBackBorder`. Sytuacja taka ma miejsce tylko w karnacjach, które w danej chwili używają przycisków `Do przodu` i `Wstecz`.

Jedynym wymaganiem dla klipów jest, aby były eksportowane do kodu ActionScript w klatce nr 1 w bibliotece.

**Modyfikowanie zachowania karnacji**

Właściwość `bufferingBarHidesAndDisablesOthers` oraz `skinAutoHide` umożliwiają dostosowanie zachowania karnacji FLVPlayback.

Ustawienie `bufferingBarHidesAndDisablesOthers` na wartość `true` powoduje, że w składniku FLVPlayback ukryty zostanie składnik `SeekBar` oraz jego uchwyt, a także przyciski `Odtwórz` i `Wstrzymaj`, gdy składnik przejdzie w stan buforowania. Może to się okazać użyteczne, gdy plik FLV strumieniowany jest z serwera FMS z użyciem wolnego połączenia z wysokim ustawieniem we właściwości `bufferTime` (np. 10). W tej sytuacji niecierpliwy użytkownik może rozpocząć przeszukiwanie, klikając przyciski `Odtwórz` lub `Wstrzymaj`, które mogą opóźnić odtworzenie pliku. Działania takiego można uniknąć, ustawiając właściwość `bufferingBarHidesAndDisablesOthers` na wartość `true` i wyłączając element `SeekBar` oraz przyciski `Wstrzymaj` i `Odtwórz`, gdy składnik jest w stanie buforowania.

Właściwość `skinAutoHide` dotyczy jedynie wcześniej zaprojektowanych plików SWF karnacji i nie elementów sterowania, które nie zostały utworzone z niestandardowych składników UI FLV Playback. Jeśli ustawiono dla niej wartość `true`, karnacja składnika FLVPlayback zostanie ukryta, gdy mysz znajdzie się poza obszarem wyświetlania. Wartością domyślną tej właściwości jest wartość `false`.

**Korzystanie z pliku SMIL**

Aby obsłużyć strumienie dla wielu przepustowości, w klasie `VideoPlayer` używana jest klasa pomocnicza (`NCManager`), w której obsługiwany jest podzbiór SMIL. SMIL używany jest w celu identyfikacji położenia strumienia wideo, układu (szerokości i wysokości) pliku FLV oraz źródłowych plików FLV, które odpowiadają różnym przepustowościom. Może być on także używany do określania szybkości transmisji oraz czasu trwania pliku FLV.

Do określania położenia pliku SMIL służy parametr `source` lub właściwość `FLVPlayback.source` (ActionScript). Więcej informacji zawiera sekcja oraz opis właściwości `FLVPlayback.source` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

W poniższym przykładzie przedstawiony został plik SMIL, który strumieniuje z serwera FMS pliki FLV o różnej przepustowości za pomocą protokołu RTMP:

```
<smil>
<head>
<meta base="rtmp://myserver/myapp/" />
<layout>
<root-layout width="240" height="180" />
</layout>
</head>
<body>
    <switch>
        <ref src="myvideo_cable.flv" dur="3:00.1"/>
        <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
        <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1"/>
    </switch>
</body>
</smil>
```

Znacznik `<head>` może zawierać znaczniki `<meta>` oraz `<layout>`. W znaczniku `<meta>` obsługiwany jest tylko atrybut `base`, który używany jest do określania adresu URL strumienia wideo (protokół RTMP z serwera FMS).

W znaczniku `<layout>` obsługiwany jest tylko element `root-layout`, który używany jest do ustawiania atrybutów `height` oraz `width`, dlatego też określany jest w nim rozmiar okna, w którym renderowany jest plik FLV. W atrybutach tych akceptowane są tylko wartości pikseli, nie zaś procenty.

Do ciała pliku SMIL można dołączyć pojedyncze łącze do źródłowego pliku FLV lub, jeśli z serwera FMS strumieniowanych jest wiele plików o różnych przepustowościach (tak, jak w poprzednim przykładzie), można użyć znacznika `<switch>` w celu wyświetlenia plików źródłowych.

Znaczniki `video` i `ref` w tagu `<switch>` są synonimami — oba znaczniki mogą używać atrybutu `src` do określania plików FLV. Każdego z atrybutów `region`, `system-bitrate` i `dur` można użyć do określenia obszaru, minimalnej wymaganej przepustowości oraz czasu trwania pliku FLV.

W znaczniku `<body>` dozwolone jest tylko jedno wystąpienie znacznika `<video>`, `<src>` lub `<switch>`.

W poniższym przykładzie przedstawione zostało pobieranie progresywne dla pojedynczego pliku FLV, w którym nie jest używane wykrywanie przepustowości:

```
<smil>
    <head>
        <layout>
            <root-layout width="240" height="180" />
        </layout>
    </head>
    <body>
        <video src="myvideo.flv" />
    </body>
</smil>
```

## <smil>

### Dostępność

Flash Professional 8.

### Zastosowanie

```
<smil>
...
child tags
...
</smil>
```

### Atrybuty

Brak.

### Znaczniki podrzędne

<head>, <body>

### Znacznik nadrzędny

Brak.

### Opis

Znacznik najwyższego poziomu, który identyfikuje plik SMIL.

### Przykład

W poniższym przykładzie przedstawiono plik SMIL, w który określa trzy pliki FLV:

```
<smil>
<head>
<meta base="rtmp://myserver/myapp/" />
<layout>
<root-layout width="240" height="180" />
</layout>
</head>
<body>
<switch>
    <ref src="myvideo_cable.flv" dur="3:00.1"/>
    <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
    <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1"/>
</switch>
</body>
</smil>
```

### <head>

### Dostępność

Flash Professional 8.

### Zastosowanie

```
<head>
...
child tags
...
</head>
```

**Korzystanie ze składnika FLVPlayback****Atrybuty**

Brak.

**Znaczniki podrzędne**

<meta>, <layout>

**Znacznik nadrzędny**

<smil>

**Opis**

Obsługuje znaczniki <meta> i <layout> i określa położenie i domyślny układ (wysokość i szerokość) źródłowych plików FLV.

**Przykład**

Poniższy przykład ustawia układ podstawowy na 240 x 180 pikseli:

```
<head>
  <meta base="rtmp://myserver/myapp/" />
  <layout>
    <root-layout width="240" height="180" />
  </layout>
</head>
```

**<meta>****Dostępność**

Flash Professional 8.

**Zastosowanie**

<meta/>

**Atrybuty**

base

**Znaczniki podrzędne**

<layout>

**Znacznik nadrzędny**

Brak.

**Opis**

Zawiera atrybut base, który określa położenie (adres URL dla protokołu RTMP) źródłowych plików FLV.

**Przykład**

W poniższym przykładzie przedstawiony został znacznik meta dla podstawowego położenia na serwerze myserver:

```
<meta base="rtmp://myserver/myapp/" />
```



## <layout>

### Dostępność

Flash Professional 8.

### Zastosowanie

```
<layout>
...
child tags
...
</layout>
```

### Atrybuty

Brak.

### Znaczniki podrzędne

```
<root-layout>
```

### Znacznik nadrzędny

```
<meta>
```

### Opis

Określa szerokości i wysokość pliku FLV.

### Przykład

W poniższym przykładzie układ określony został na 240 na 180 pikseli:

```
<layout>
  <root-layout width="240" height="180" />
</layout>
```

## <root-layout>

### Dostępność

Flash Professional 8.

### Zastosowanie

```
<root-layout...attributes.../>
```

### Atrybuty

Width, height

### Znaczniki podrzędne

Brak.

### Znacznik nadrzędny

```
<layout>
```

## Opis

Określa szerokości i wysokość pliku FLV.

## Przykład

W poniższym przykładzie układ określony został na 240 na 180 pikseli:

```
<root-layout width="240" height="180" />
```

## <body>

### Dostępność

Flash Professional 8.

### Zastosowanie

```
<body>
...
child tags
...
</body>
```

### Atrybuty

Brak.

### Znaczniki podrzędne

<video>, <ref>, <switch>

### Znacznik nadrzędny

<smil>

## Opis

Zawiera znaczniki <video>, <ref> i <switch>, które określają nazwę źródłowego pliku FLV, minimalną przepustowość oraz czas trwania pliku FLV. Atrybut `system-bitrate` obsługiwany jest tylko, gdy używany jest znacznik <switch>. W znaczniku <body> dozwolona jest tylko jedna instancja znacznika <switch>, <video> lub <ref>.

## Przykład

W poniższym przykładzie określone zostały trzy pliki FLV, dwa za pomocą znacznika `video` i jeden za pomocą znacznika `ref`:

```
<body>
  <switch>
    <ref src="myvideo_cable.flv" dur="3:00.1"/>
    <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
    <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1"/>
  </switch>
</body>
```

## <video>

### Dostępność

Flash Professional 8.

### Zastosowanie

```
<video...attributes.../>
```

### Atrybuty

src, system-bitrate, dur

### Znaczniki podrzędne

Brak.

### Znacznik nadrzędny

```
<body>
```

### Opis

Synonim ze znacznikiem <ref>. Obsługuje atrybuty src i dur , które określają nazwę źródłowego pliku FLV oraz czas jego trwania. Atrybut dur obsługuje pełny (00:03:00:01) i częściowy format czasu (03:00:01).

### Przykład

W poniższym przykładzie dla video ustawiane jest źródło i czas trwania:

```
<video src="myvideo_mdm.flv" dur="3:00.1"/>
```

## <ref>

### Dostępność

Flash Professional 8.

### Zastosowanie

```
<ref...attributes.../>
```

### Atrybuty

src, system-bitrate, dur

### Znaczniki podrzędne

Brak.

### Znacznik nadrzędny

```
<body>
```

### Opis

Synonim ze znacznikiem <video>. Obsługuje atrybuty src i dur , które określają nazwę źródłowego pliku FLV oraz czas jego trwania. Atrybut dur obsługuje pełny (00:03:00:01) i częściowy format czasu (03:00:01).

### Przykład

W poniższym przykładzie dla wideo ustawiane jest źródło i czas trwania:

```
<ref src="myvideo_cable.flv" dur="3:00.1"/>
```

## <switch>

### Dostępność

Flash Professional 8.

### Zastosowanie

```
<switch>  
...  
child tags  
...  
</switch/>
```

### Atrybuty

Brak.

### Znaczniki podrzędne

<video>, <ref>

### Znacznik nadrzędny

<body>

### Opis

Używany z podrzędnymi znacznikami <video> lub <ref> do wyświetlania plików FLV dla strumieniowania wideo o różnych przepustowościach. Znacznik <switch> obsługuje atrybut `system-bitrate`, który określa minimalną przepustowość, a także atrybuty `src` oraz `dur`.

### Przykład

W poniższym przykładzie określone zostały trzy pliki FLV, dwa za pomocą znacznika `video` i jeden za pomocą znacznika `ref`:

```
<switch>  
  <ref src="myvideo_cable.flv" dur="3:00.1"/>  
  <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>  
  <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1" />  
</switch>
```

# Rozdział 7: Korzystanie ze składnika FLVPlaybackCaptioning

Składnik FLVPlayback umożliwia łatwe dołączenie odtwarzacza wideo do aplikacji Adobe Flash CS5 Professional w celu odtwarzania pobieranych plików Adobe Flash Video (FLV lub F4V) oraz strumieniowego przesyłania plików FLV lub F4V. Więcej informacji na temat składnika FLVPlayback zawiera sekcja „Korzystanie ze składnika FLVPlayback” na stronie 139.

Składnik FLVPlaybackCaptioning umożliwia dołączanie funkcji obsługi podpisów do plików wideo. Składnik podpisów obsługuje standardowy format XML Timed Text i zawiera następujące funkcje:

**Podpisy z osadzonymi punktami sygnalizacji o zdarzeniach** Kojarzenie osadzonych punktów sygnalizacji o zdarzeniach w pliku FLV z kodem XML w celu udostępnienia podpisów — zamiast korzystania z pliku XML Timed Text.

**Wiele składników FLVPlaybackCaptioning** Tworzenie wielu instancji składników FLVPlaybackCaptioning dla wielu instancji FLVPlayback.

**Sterowanie za pomocą przycisku przełączania** Udostępnianie podpisów podczas interakcji użytkownika za pośrednictwem przycisku przełączania podpisów.

## Korzystanie ze składnika FLVPlaybackCaptioning

Składnik FLVPlaybackCaptioning może być używany z jednym lub większą liczbą składników FLVPlayback. W najprostszym scenariuszu należy przeciągnąć składnik FLVPlayback na stół montażowy, przeciągnąć składnik FLVPlaybackCaptioning na ten sam stół, określić adres URL podpisów i ustawić podpisy do wyświetlenia. Ponadto możliwe jest również ustawienie różnych parametrów w celu dostosowania składnika FLVPlaybackCaptioning.

### Dodawanie podpisów do składnika FLVPlayback

Składnik FLVPlaybackCaptioning można dodać do dowolnego składnika FLVPlayback. Informacje o dodawaniu składników FLVPlayback do aplikacji zawiera sekcja „Tworzenie aplikacji za pomocą składnika FLVPlayback” na stronie 141.

#### Dodawanie składnika FLVPlaybackCaptioning z panelu Składniki:

- 1 W panelu Składniki otwórz folder Video.
- 2 Przeciągnij (lub kliknij dwukrotnie) składnik FLVPlaybackCaptioning, a następnie dodaj go do tego stołu montażowego, na którym znajduje się składnik FLVPlayback, do którego dodane zostaną podpisy.


***Uwaga:** Firma Adobe udostępnia dwa pliki, na których można łatwo nauczyć się korzystania ze składnika FLVPlaybackCaptioning: `caption_video.flv` (przykład ze składnikiem FLVPlayback) i `caption_video.xml` (przykład z podpisami). Pliki te są dostępne pod adresami: [www.helpexamples.com/flash/video/caption\\_video.flv](http://www.helpexamples.com/flash/video/caption_video.flv) oraz [www.helpexamples.com/flash/video/caption\\_video.xml](http://www.helpexamples.com/flash/video/caption_video.xml).*

- 3 (Opcjonalnie) Przeciągnij składnik CaptionButton do stołu montażowego, na którym znajdują się składniki FLVPlayback i FLVPlaybackCaptioning. Składnik CaptionButton umożliwia użytkownikowi włączanie i wyłączanie podpisów.

**Uwaga:** Aby włączyć składnik `CaptionButton`, przeciągnij go na ten stół montażowy, na którym znajdują się składniki `FLVPlayback` i `FLVPlaybackCaptioning`.

- 4 Po wybraniu składnika `FLVPlaybackCaptioning` na stole montażowym określ następujące informacje na karcie Parametry w Inspektorze właściwości:

- Dla `showCaptions` ustaw wartość `true`.
- Określ wartość `source` dla pliku XML Timed Text, który zostanie pobrany.

 Podczas testowania podpisów w programie Flash należy ustawić dla właściwości `showCaptions` wartość `true`. Jeśli jednak zostanie dołączony składnik `CaptionButton`, który będzie umożliwiał użytkownikom włączanie i wyłączenie podpisów, wówczas dla właściwości `showCaptions` należy ustawić wartość `false`.

Dostępne są również inne parametry przeznaczone do dostosowywania składnika `FLVPlaybackCaptioning`. Więcej informacji zawiera sekcja „Dostosowywanie składnika `FLVPlaybackCaptioning`” na stronie 189 oraz *Skorowidz języka ActionScript 3.0 dla programu Flash Professional*.

- 5 Wybierz polecenia Sterowanie > Testuj film, aby uruchomić plik wideo.

### Dynamiczne tworzenie wystąpień za pomocą języka ActionScript:

- 1 Przeciągnij składnik `FLVPlayback` z panelu Składnik do panelu Biblioteka (Okno > Biblioteka).
- 2 Przeciągnij składnik `FLVPlaybackCaptioning` z panelu Składnik do panelu Biblioteka.
- 3 Dodaj poniższy kod do panelu Operacje w klatce 1 na osi czasu.

```
import fl.video.*;
var my_FLVPlybk = new FLVPlayback();
my_FLVPlybk.x = 100;
my_FLVPlybk.y = 100;
addChild(my_FLVPlybk);
my_FLVPlybk.skin = "install_drive:/Program Files/Adobe/Adobe Flash
CS5/en/Configuration/FLVPlayback Skins/ActionScript 3.0/SkinUnderPlaySeekCaption.swf";
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/caption_video.flv";
var my_FLVPlybkcap = new FLVPlaybackCaptioning();
addChild(my_FLVPlybkcap);
my_FLVPlybkcap.source = "http://www.helpexamples.com/flash/video/caption_video.xml";
my_FLVPlybkcap.showCaptions = true;
```

- 4 Zmień `install_drive` na napęd, na którym zainstalowano program Flash, a następnie zmodyfikuj ścieżkę w celu uwzględnienia lokalizacji folderu `Skins` dla instalacji:

**Uwaga:** Jeśli instancja `FLVPlayback` jest tworzona za pomocą `ActionScript`, należy również dynamicznie przypisać do niej karnację, poprzez ustawienie właściwości `skin` w `ActionScript`. Gdy karnacja zostanie w ten sposób zastosowana za pomocą `ActionScript`, nie będzie automatycznie publikowana z plikiem `SWF`. Skopiuj karnację `SWF` oraz plik `SWF` aplikacji na serwer — w przeciwnym wypadku plik `SWF` karnacji nie będzie dostępny, gdy użytkownik go uruchomi.

## Ustawianie parametrów składnika `FLVPlaybackCaptioning`

Dla każdej instancji składnika `FLVPlaybackCaptioning` można ustawić w Inspektorze właściwości lub Inspektorze składników następujące parametry w celu dalszego dostosowywania składnika. Poniższa lista zawiera nazwy i krótkie opisy właściwości:

**autoLayout** Określa, czy składnik `FLVPlaybackCaptioning` kontroluje wielkość obszaru podpisu. Wartością domyślną jest `true`.

**Korzystanie ze składnika FLVPlaybackCaptioning**

**captionTargetName** Określa nazwy instancji TextField lub MovieClip, które zawierają podpisy. Domyślnie: auto.

**flvPlaybackName** Określa nazwę instancji obiektu FLVPlayback, która ma być opatrzona podpisami. Domyślnie: auto.

**simpleFormatting** Jeśli ma wartość true, ogranicza formatowanie pliku XML Timed Text. Wartość domyślna to false.

**showCaptions** Określa, czy podpisy są wyświetlane. Wartością domyślną jest true.

**source** Określa lokalizację pliku XML Timed Text.

Więcej informacji na temat wszystkich parametrów FLVPlaybackCaptioning zawiera *Skorowidz składników ActionScript 3.0 dla programu Flash Professional*.

**Określanie parametru source składnika**

Parametr `source` służy do określania nazwy i lokalizacji pliku XML Timed Text, który zawiera podpisy dla filmu. Ścieżkę URL należy wprowadzić bezpośrednio do komórki `source` w Inspektorze składników.

**Wyświetlanie podpisów**

W celu wyświetlenia podpisów należy ustawić dla parametru `showCaptions` wartość `true`.

Więcej informacji na temat wszystkich parametrów składnika FLVPlaybackCaptioning zawiera *Skorowidz składników ActionScript 3.0 dla programu Flash Professional*.

W poprzednich przykładach przedstawiono sposób tworzenia i włączania składnika FLVPlaybackCaptioning w celu wyświetlania podpisów. Istnieją dwa źródła, które mogą być wykorzystywane dla podpisów: (1) plik XML Timed Text XML zawierający podpisy lub (2) plik XML zawierający tekst podpisu, który należy skojarzyć z osadzonymi punktami sygnalizacji zdarzeń.

## Korzystanie z podpisów Timed Text

Składnik FLVPlaybackCaptioning umożliwia tworzenie podpisów dla skojarzonego składnika FLVPlayback — w tym celu należy pobrać plik XML Timed Text (TT). Więcej informacji o formacie Timed Text zawiera sekcja poświęcona formatowi AudioVideo Timed Text na stronie <http://www.w3.org>.

Ta sekcja zawiera ogólne informacje o obsługiwanych znacznikach Timed Text, wymaganych znacznikach pliku podpisów, a także zawiera przykładowy plik XML Timed Text. Szczegółowe informacje na temat wszystkich obsługiwanych znaczników z synchronizacją czasową zawiera sekcja „[Znaczniki Timed Text](#)” na stronie 181.

Składnik FLVPlaybackCaptioning obsługuje następujące znaczniki Timed Text:

**Korzystanie ze składnika FLVPlaybackCaptioning**

Kategoria	Zadanie
Obsługa formatowania akapitów	Wyrównanie akapitu do prawej lub lewej, wyśrodkowanie
Obsługa formatowania tekstu	<ul style="list-style-type: none"> <li>• Ustawienie wielkości tekstu za bezwzględnie za pomocą wielkości pikseli lub przyrostowo (np. +2, -4).</li> <li>• Ustawianie koloru i czcionki tekstu</li> <li>• Pogrubianie czcionki i zmiana na kursywę</li> <li>• Ustawianie justowania tekstu</li> </ul>
Obsługa innych operacji formatowania	<ul style="list-style-type: none"> <li>• Ustawianie koloru tła pola TextField dla podpisów</li> <li>• Ustawianie przezroczystego tła pola TextField dla podpisów (alfa 0)</li> <li>• Ustawianie zawijania słów w polu TextField dla podpisów (włączanie/wyłączanie)</li> </ul>

Składnik FLVPlaybackCaptioning jest zsynchronizowany z kodem czasowym pliku FLV. Każdy podpis musi mieć atrybut `begin`, który określa pożądaną punkt czasowy pojawienia się podpisu. Jeśli podpis nie ma atrybutu `dur` ani `end`, znika w momencie pojawienia się następnego podpisu lub po zakończeniu pliku FLV.

Poniżej przedstawiono przykładowy plik XML w formacie Timed Text. Plik ten (`caption_video.xml`) zawiera napisy dla pliku `caption_video.flv`. Pliki te są dostępne pod adresami [www.helpexamples.com/flash/video/caption\\_video.flv](http://www.helpexamples.com/flash/video/caption_video.flv) oraz [www.helpexamples.com/flash/video/caption\\_video.xml](http://www.helpexamples.com/flash/video/caption_video.xml).



**Korzystanie ze składnika FLVPlaybackCaptioning**

```

<?xml version="1.0" encoding="UTF-8"?>
  <tt xml:lang="en"
xmlns="http://www.w3.org/2006/04/ttaf1"xmlns:tts="http://www.w3.org/2006/04/ttaf1#styling">
<head>
  <styling>
<style id="1" tts:textAlign="right"/>
<style id="2" tts:color="transparent"/>
<style id="3" style="2" tts:backgroundColor="white"/>
<style id="4" style="2 3" tts:fontSize="20"/>
  </styling>
</head>
<body>
  <div xml:lang="en">
<p begin="00:00:00.00" dur="00:00:03.07">I had just joined <span
tts:fontFamily="monospaceSansSerif,proportionalSerif,TheOther"tts:fontSize="+2">Macromedia</span> in 1996,</p>
<p begin="00:00:03.07" dur="00:00:03.35">and we were trying to figure out what to do about the internet.</p>
<p begin="00:00:06.42" dur="00:00:03.15">And the company was in dire straights at the time.</p>
<p begin="00:00:09.57" dur="00:00:01.45">We were a CD-ROM authoring company,</p>
<p begin="00:00:11.42" dur="00:00:02.00">and the CD-ROM business was going away.</p>
<p begin="00:00:13.57" dur="00:00:02.50">One of the technologies I remember seeing was Flash.</p>
<p begin="00:00:16.47" dur="00:00:02.00">At the time, it was called <span
tts:fontWeight="bold" tts:color="#ccc333">FutureSplash</span>.</p>
<p begin="00:00:18.50" dur="00:00:01.20">So this is where Flash got its start.</p>
<p begin="00:00:20.10" dur="00:00:03.00">This is smart sketch running on the <span
tts:fontStyle="italic">EU-pin computer</span>,</p>
<p begin="00:00:23.52" dur="00:00:02.00">which was the first product that FutureWave did.</p>
<p begin="00:00:25.52" dur="00:00:02.00">So our vision for this product was to</p>
<p begin="00:00:27.52" dur="00:00:01.10">make drawing on the computer</p>
<p begin="00:00:29.02" dur="00:00:01.30" style="1">as <span tts:color="#ccc333">easy</span>
as drawing on paper.</p>
</div>
</body>
</tt>

```

**Znaczniki Timed Text**

Składnik FLVPlaybackCaptioning obsługuje znaczniki Timed Text w plikach XML definiujących podpisy. Więcej informacji na temat znaczników Timed Text służących do synchronizacji tekstu z materiałami audio i wideo można znaleźć pod adresem [www.w3.org](http://www.w3.org). W poniższej tabeli wymieniono znaczniki obsługiwane i nieobsługiwane.

Funkcja	Znacznik/wartość	Zastosowanie/opis	Przykład
Znaczniki ignorowane	metadata	Ignorowany/dozwolony na każdym poziomie dokumentu	
	set	Ignorowany/dozwolony na każdym poziomie dokumentu	
	xml:lang	Ignorowany	
	xml:space	Ignorowany / wymuszone jest zachowanie: xml:space="default"	

## Korzystanie ze składnika FLVPlaybackCaptioning

Funkcja	Znacznik/wartość	Zastosowanie/opis	Przykład
	layout	Ignorowany/łącznie ze wszelkimi znacznikami region w sekcji znacznika layout.	
	Znacznik br	Wszystkie atrybuty, wraz z zawartością, są ignorowane.	
Synchronizacja czasowa multimediiów i podpisów	Atrybuty begin	Dozwolone tylko w znacznikach p. Wymagane do określania czasu pojawiania się podpisów.	<p begin="3s">
	Atrybuty dur	Dozwolone tylko w znacznikach p. Zalecane. Pominięcie tego atrybutu spowoduje, że podpis zniknie dopiero w momencie zakończenia pliku FLV lub rozpoczęcia następnego podpisu.	
	Atrybuty end	Dozwolone tylko w znacznikach p. Zalecane. Pominięcie tego atrybutu spowoduje, że podpis zniknie dopiero w momencie zakończenia pliku FLV lub rozpoczęcia następnego podpisu.	
Format zegara dla podpisów	00:03:00.1	Pełny format zegara	
	03:00.1	Częściowy format zegara	
	10	Czas przesunięcia bez jednostek. Przesunięcie wyrażone w sekundach.	
	00:03:00:05 00:03:00:05.1 30f 30t	Nieobsługiwany. Formaty czasu obejmujące klatki lub podziałkę (ang. tick) nie są obsługiwane.	
Znacznik części głównej pliku	body	Wymagany / dozwolony tylko jeden znacznik body.	<body><div>...</div></body>
Znacznik treści	Znacznik div	Dozwolone zero lub więcej znaczników. Używany jest pierwszy znacznik.	
	Znacznik p	Dozwolone zero lub więcej znaczników..	
	Znacznik span	Logiczny kontener szeregu jednostek treści tekstowej. Zagnieżdżone znaczniki span nie są obsługiwane. Obsługiwane są znaczniki stylu atrybutów.	
	Znacznik br	Oznacza jawny podział wiersza.	
Znaczniki opisujące style  (Wszystkie znaczniki stylów są używane wewnątrz znacznika p)	style	Odwołanie do jednego lub większej liczby elementów stylu. Może być używany zarówno jako znacznik, jak i w charakterze atrybutu. W przypadku znacznika wymagany jest atrybut ID (styl może być wielokrotnie używany w dokumencie). Wewnątrz znacznika style może wystąpić jeden lub wiele innych znaczników style.	

## Korzystanie ze składnika FLVPlaybackCaptioning

Funkcja	Znacznik/wartość	Zastosowanie/opis	Przykład
	tts:background Color	Określa właściwość stylu definiującą kolor tła regionu. Wartość alfa jest ignorowana, chyba że zostanie ustawiona na zero (alfa 0), co powoduje, że tło staje się przezroczyste. Format koloru to #RRGGBBAA.	
	tts:color	Określa właściwość stylu definiującą kolor pierwszego planu. Wartość alfa nie jest obsługiwana dla jakichkolwiek kolorów. Wartość <code>transparent</code> jest interpretowana jako kolor czarny.	<pre>&lt;style id="3" style="2" tts:backgroundColor="white"/&gt; "transparent" = #00000000 "black"=#000000FF "silver"=#C0C0C0FF "grey"=#808080FF "white"=#FFFFFFFF "maroon"=#800000FF "red"=#FF0000FF "purple"=#800080FF "fuchsia("magenta")= #FF00FFFF "green"=#008000FF "lime"=#00FF00FF "olive"=#808000FF "yellow"=#FFFF00FF "navy"=#000080FF "blue"=#0000FFFF "teal"=#008080FF "aqua("cyan")=#00FFFFFF</pre>
	tts:fontFamily	Określa właściwość stylu definiującą rodzinę czcionek.	<pre>"default" = _serif "monospace" = _typewriter "sansSerif" = _sans "serif" = _serif "monospaceSansSerif" = _typewriter "monospaceSerif" = _typewriter "proportionalSansSerif" = _sans</pre>
	tts:fontSize	Określa właściwość stylu definiującą rozmiar czcionki. W wypadku podania dwóch wartości, używana jest tylko pierwsza (rozmiar w pionie). Wartości procentowe i jednostki są ignorowane. Obsługiwane są rozmiary w pikselach podane bezwzględnie (np. 12) oraz względnie (np. +2).	

Funkcja	Znacznik/wartość	Zastosowanie/opis	Przykład
	tts:fontStyle	Określa właściwość stylu definiującą styl czcionki.	"normal" "italic" "inherit"* * Zachowanie domyślne; dziedziczy styl ze znacznika okalającego.
	tts:fontWeight	Określa właściwość stylu definiującą grubość czcionki.	"normal" "bold" "inherit"* * Zachowanie domyślne; dziedziczy styl ze znacznika okalającego.

Funkcja	Znacznik/wartość	Zastosowanie/opis	Przykład
	tts:textAlign	Określa właściwość stylu definiującą sposób wyrównania obszarów wewnątrz wierszy z obszarem bloku, w którym te wiersze są zawarte.	"left" "right" "center" "start" ("=left") "end" ("=right") "inherit" *Dziedziczy styl ze znacznika okalającego. Jeśli znacznik textAlign nie jest ustawiony, przyjmowany jest styl domyślny "left".
	tts:wrapOption	Określa właściwość stylu, która definiuje, czy w kontekście danego elementu ma zastosowanie automatyczne zawijanie (dzielenie) wierszy. To ustawienie ma wpływ na wszystkie akapity w elemencie caption.	"wrap" "noWrap" "inherit" *Dziedziczy styl ze znacznika okalającego. Jeśli znacznik wrapOption nie jest ustawiony, przyjmowany jest styl domyślny "wrap".
Atrybuty nieobsługiwane	tts:direction tts:display tts:displayAlign tts:dynamicFlow tts:extent tts:lineHeight tts:opacity tts:origin tts:overflow tts:padding tts:showBackground tts:textOutline tts:unicodeBidi tts:visibility tts:writingMode tts:zIndex		

## Korzystanie z punktów sygnalizacji razem z podpisami

Punkty sygnalizacji umożliwiają interakcję z materiałem wideo; można na przykład wpływać na odtwarzanie pliku FLV lub wyświetlać tekst w określonych punktach czasowych materiału wideo. Jeśli nie jest dostępny plik XML w formacie Timed Text przeznaczony do użycia z plikiem FLV, można osadzić punkty sygnalizacji zdarzeń w pliku FLV, a następnie skojarzyć te punkty z tekstem. W niniejszej sekcji zamieszczono informacje o standardach punktów sygnalizacji stosowanych w składniku FLVPlaybackCaptioning oraz krótki przegląd metody kojarzenia punktów sygnalizacji z tekstem podpisów. Więcej informacji na temat osadzania punktów sygnalizacji zdarzeń za pomocą kreatora importu wideo lub programu kodującego wideo Flash zawiera Rozdział 16, „Wideo”, w podręczniku *Korzystanie z programu Flash*.

### Omówienie standardów punktów sygnalizacji używanych w składniku FLVPlaybackCaptioning

W metadanych pliku FLV punkt sygnalizacji jest reprezentowany przez obiekt z następującymi właściwościami: `name`, `time`, `type` i `parameters`. Punkty sygnalizacji stosowane w składniku FLVPlaybackCaptioning języka ActionScript mają następujące atrybuty:

**name** Właściwość `name` jest ciągiem znaków, który zawiera nazwę przypisaną do punktu sygnalizacji. Właściwość `name` musi rozpoczynać się od przedrostka `fl.video.caption.2.0.`, po którym musi następować ciąg. Ciąg jest szeregiem dodatkich liczb całkowitych zwiększanych o jeden w każdym wystąpieniu, tak aby nazwy punktów były unikatowe. Przedrostek zawiera numer wersji zgodny z numerem wersji składnika FLVPlayback. W przypadku programu Adobe Flash CS4 i nowszych wersji należy ustawić numer wersji na `2.0`.

**time** Właściwość `time` to czas, w którym podpis powinien zostać wyświetlony.

**type** Właściwość `type` to ciąg znaków o wartości `"event"`.

**parameters** Właściwość `parameters` to tablica, która może zawierać następujące pary nazwa-wartość:

- **text:String** Tekst podpisu w formacie HTML. Tekst jest przekazywany bezpośrednio do właściwości `TextField.htmlText`. Składnik FLVPlaybackCaptioning obsługuje opcjonalną właściwość `text:n` umożliwiającą używanie wielu wersji językowych. Więcej informacji zawiera sekcja „[Obsługa wielu wersji językowych przy użyciu osadzonych punktów sygnalizacji](#)” na stronie 188.
- **endTime:Number** Czas, w którym podpis powinien zniknąć. Jeśli wartość tej właściwości *nie zostanie określona*, składnik FLVPlaybackCaptioning przyjmuje, że nie jest ona liczbą (NaN), a podpis jest wyświetlany aż do zakończenia pliku FLV (momentu, w którym instancja FLVPlayback wywoła zdarzenie `VideoEvent.COMPLETE`). Wartość właściwości `endTime:Number` należy podać w sekundach.
- **backgroundColor:uint** Ten parametr ustawia właściwość `TextField.backgroundColor`. Ta właściwość jest opcjonalna.
- **backgroundColorAlpha:Boolean** Jeśli składnik alfa wartości `backgroundColor` jest równy 0%, dokonywane jest przypisanie `TextField.background = !backgroundColor`. Ta właściwość jest opcjonalna.
- **wrapOption:Boolean** Ten parametr ustawia właściwość `TextField.wordWrap`. Ta właściwość jest opcjonalna.

## Podpisy realizowane przy użyciu osadzonych punktów sygnalizacji zdarzeń

Jeśli nie jest dostępny plik XML w formacie Timed Text zawierający podpisy dla pliku FLV, można utworzyć podpisy, korzystając ze skojarzonego pliku XML zawierającego podpisy sterowane przez osadzone punkty sygnalizacji zdarzeń. Przykładowy plik XML opracowano przy założeniu, że użytkownik wykonał następujące czynności w celu utworzenia osadzonych punktów sygnalizacji zdarzeń w pliku wideo:

- Dodaj pliki sygnalizacji zdarzeń (zgodne ze standardami stosowanymi w składniku FLVPlaybackCaptioning) i zakoduj wideo.
- W programie Flash przeciągnij składnik FLVPlayback oraz składnik FLVPlaybackCaptioning na stół montażowy.
- Ustaw właściwości źródła składnika FLVPlayback i FLVPlaybackCaptioning (lokalizację pliku FLV i lokalizację pliku XML).
- Dokonaj publikacji.

Poniższy przykład importuje plik XML do kodera:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<FLVCoreCuePoints>

  <CuePoint>
    <Time>9136</Time>
    <Type>event</Type>
    <Name>fl.video.caption.2.0.index1</Name>
    <Parameters>
      <Parameter>
        <Name>text</Name>
        <Value><![CDATA[Captioning text for the first cue point]]></Value>
      </Parameter>
    </Parameters>
  </CuePoint>

  <CuePoint>
    <Time>19327</Time>
    <Type>event</Type>
    <Name>fl.video.caption.2.0.index2</Name>
    <Parameters>
      <Parameter>
        <Name>text</Name>
        <Value><![CDATA[Captioning text for the second cue point]]></Value>
      </Parameter>
    </Parameters>
  </CuePoint>

  <CuePoint>
    <Time>24247</Time>
    <Type>event</Type>
    <Name>fl.video.caption.2.0.index3</Name>
    <Parameters>
```

```

        <Parameter>
            <Name>text</Name>
            <Value><![CDATA[Captioning text for the third cue point]]></Value>
        </Parameter>
    </Parameters>
</CuePoint>

<CuePoint>
    <Time>36546</Time>
    <Type>event</Type>
    <Name>fl.video.caption.2.0.index4</Name>
    <Parameters>
        <Parameter>
            <Name>text</Name>
            <Value><![CDATA[Captioning text for the fourth cue point]]></Value>
        </Parameter>
    </Parameters>
</CuePoint>

</FLVCoreCuePoints>

```

Składnik FLVPlaybackCaptioning obsługuje także wiele wersji językowych napisów realizowanych przy użyciu osadzonych punktów sygnalizacji. Więcej informacji zawiera sekcja „[Obsługa wielu wersji językowych przy użyciu osadzonych punktów sygnalizacji](#)” na stronie 188.

## Obsługa wielu wersji językowych przy użyciu osadzonych punktów sygnalizacji

Właściwość `track` składnika FLVPlaybackCaptioning umożliwia utworzenie wielu wersji językowych przy użyciu osadzonych punktów sygnalizacji, o ile tylko plik XML w formacie Timed Text jest zgodny ze standardami punktów sygnalizacji stosowanymi w składniku FLVPlaybackCaptioning. (Więcej informacji zawiera sekcja „[Omówienie standardów punktów sygnalizacji używanych w składniku FLVPlaybackCaptioning](#)” na stronie 186.) Jednak FLVPlaybackCaptioning nie obsługuje wielu wersji językowych podpisów w odrębnych plikach XML. Aby użyć właściwości `track`, należy ustawić ją na wartość różną od 0. Na przykład, w wypadku ustawienia właściwości na 1 (`track == 1`), składnik FLVPlaybackCaptioning będzie szukał parametrów punktów sygnalizacji. W razie nieznaledzenia dopasowanych parametrów, użyta zostanie właściwość `text` z parametrów punktu sygnalizacji. Więcej informacji zawiera opis właściwości `track` w *Skorowidzu języka ActionScript 3.0 dla programu Flash Professional*.

## Odtwarzanie wielu plików FLV z podpisami

W jednej instancji składnika FLVPlayback można otworzyć wiele odtwarzaczy wideo, aby odtwarzać wiele plików wideo i przełączać je podczas odtwarzania. Można również skojarzyć wiele podpisów z każdym odtwarzaczem wideo w składniku FLVPlayback. Więcej informacji o otwieraniu wielu odtwarzaczy wideo zawiera sekcja „[Korzystanie z wielu odtwarzaczy wideo](#)” na stronie 155. Aby użyć podpisów w wielu odtwarzaczach wideo, należy utworzyć jedną instancję składnika FLVPlaybackCaptioning dla każdego obiektu `VideoPlayer` i ustawić właściwość `videoPlayerIndex` składnika FLVPlaybackCaptioning na odpowiedni indeks. Indeks obiektu `VideoPlayer` jest domyślnie równy 0, gdy istnieje tylko jeden obiekt `VideoPlayer`.

Poniższej przedstawiono przykładowy kod, który przypisuje unikalne podpisy do unikalnych plików wideo. Aby można było uruchomić ten przykład, fikcyjne adresy URL w przykładzie należy zastąpić działającymi adresami URL.



**Korzystanie ze składnika FLVPlaybackCaptioning**

```
captioner0.videoPlayerIndex = 0;
captioner0.source = "http://www.[yourDomain].com/mytimedtext0.xml";
flvPlayback.play("http://www.[yourDomain].com/myvideo0.flv");
captioner1.videoPlayerIndex = 1;
captioner1.source = "http://www.[yourDomain].com/mytimedtext1.xml";
flvPlayback.activeVideoIndex = 1;
flvPlayback.play("http://www.[yourDomain].com/myvideo1.flv");
```

## Dostosowywanie składnika FLVPlaybackCaptioning

Aby szybko rozpocząć korzystanie ze składnika FLVPlaybackCaptioning, można używać jego ustawień domyślnych, które powodują nałożenie podpisów bezpośrednio na składnik FLVPlayback. Aby przesunąć podpisy poza obraz wideo, można dostosować składnik FLVPlaybackCaptioning.

Poniższy kod ilustruje dynamiczne tworzenie obiektu FLVPlayback z przyciskiem przełączania podpisów:

- 1 Umieść składnik FLVPlayback na stole montażowym w pozycji 0,0 i nadaj instancji nazwę **player**.
- 2 Umieść składnik FLVPlaybackCaptioning na stole montażowym w pozycji 0,0 i nadaj instancji nazwę **captioning**.
- 3 Umieść składnik CaptionButton na stole montażowym.
- 4 W poniższym przykładzie przypisz zmiennej `testVideoPath:String` plik FLV (podając ścieżkę bezwzględną lub względną).

**Uwaga:** Przykładowy kod przypisuje zmiennej `testVideoPath` przykładowy plik wideo Flash o nazwie `caption_video.flv`. Zmień tę wartość na ścieżkę składnika wideo z podpisami, do którego dodawany jest składnik przycisku.

- 5 W poniższym przykładzie przypisz zmiennej `testCaptioningPath:String` odpowiedni plik XML w formacie Timed Text (podając ścieżkę bezwzględną lub względną).

**Uwaga:** Przykładowy kod przypisuje zmiennej `testCaptioningPath` plik XML Timed Text o nazwie `caption_video.xml`. Zmień tę wartość na ścieżkę pliku XML w formacie Timed Text, który zawiera podpisy dla pliku wideo.

- 6 Zapisz poniższy kod jako plik `FLVPlaybackCaptioningExample.as` w tym samym katalogu, co plik FLA.
- 7 Ustaw właściwość `DocumentClass` pliku FLA na `FLVPlaybackCaptioningExample`.

```
package
{
    import flash.display.Sprite;
    import flash.text.TextField;
    import fl.video.FLVPlayback;
    import fl.video.FLVPlaybackCaptioning;

    public class FLVPlaybackCaptioningExample extends Sprite {

        private var testVideoPath:String =
"http://www.helpexamples.com/flash/video/caption_video.flv";
        private var testCaptioningPath:String =
"http://www.helpexamples.com/flash/video/caption_video.xml";

        public function FLVPlaybackCaptioningExample() {
            player.source = testVideoPath;
            player.skin = "SkinOverAllNoCaption.swf";
            player.skinBackgroundColor = 0x666666;
            player.skinBackgroundAlpha = 0.5;

            captioning.flvPlayback = player;
            captioning.source = testCaptioningPath;
            captioning.autoLayout = false;
            captioning.addEventListener("captionChange", onCaptionChange);
        }
        private function onCaptionChange(e:*) :void {
            var tf:* = e.target.captionTarget;
            var player:FLVPlayback = e.target.flvPlayback;

            // move the caption below the video
            tf.y = 210;
        }
    }
}
```

Więcej informacji na temat wszystkich parametrów FLVPlaybackCaptioning zawiera *Skorowidz składników ActionScript 3.0 dla programu Flash Professional*.