

Uso de componentes do ACTIONSCRIPT® 3.0

Avisos legais

Para ver os avisos legais, consulte http://help.adobe.com/pt_BR/legalnotices/index.html.

Conteúdo

Capítulo 1: Introdução

| | |
|----------------------------------|---|
| Público-alvo | 1 |
| Requisitos do sistema | 1 |
| Sobre a documentação | 2 |
| Convenções tipográficas | 2 |
| Termos usados neste manual | 2 |
| Recursos adicionais | 2 |

Capítulo 2: Sobre os componentes do ActionScript 3.0

| | |
|--|----|
| Vantagens do uso de componentes | 4 |
| Tipos de componentes | 6 |
| Adicionar e excluir de um documento | 7 |
| Localizar a versão do componente | 9 |
| Modelo de manipulação de eventos do ActionScript 3.0 | 10 |
| Um aplicativo simples | 11 |

Capítulo 3: Trabalho com componentes

| | |
|--|----|
| Arquitetura do componente | 18 |
| Trabalho com arquivos de componente | 20 |
| Depuração dos aplicativos de componente | 22 |
| Definição de parâmetros e propriedades | 22 |
| A biblioteca | 24 |
| Dimensionamento de componentes | 24 |
| Visualização ao vivo | 25 |
| Manipulação de eventos | 25 |
| Trabalho com a lista de exibição | 27 |
| Trabalho com o FocusManager | 29 |
| Trabalho com componentes baseados em lista | 31 |
| Trabalho com um DataProvider | 31 |
| Trabalho com um CellRenderer | 39 |
| Tornar os componentes acessíveis | 46 |

Capítulo 4: Uso dos componentes de UI

| | |
|--|----|
| Uso do componente Button | 47 |
| Uso do componente CheckBox | 49 |
| Uso do componente ColorPicker | 52 |
| Uso do componente ComboBox | 55 |
| Uso do componente DataGrid | 58 |
| Uso do componente Label | 64 |
| Uso do componente List | 66 |
| Uso do componente NumericStepper | 70 |
| Uso do componente ProgressBar | 73 |
| Uso do componente RadioButton | 78 |

Conteúdo

| | |
|-------------------------------------|----|
| Uso do componente ScrollPane | 81 |
| Uso do componente Slider | 84 |
| Uso do componente TextArea | 87 |
| Uso do componente TextInput | 90 |
| Uso do componente TileList | 93 |
| Uso do componente UILoader | 96 |
| Uso do componente UIScrollBar | 98 |

Capítulo 5: Personalização dos componentes de UI

| | |
|--|-----|
| Sobre a personalização dos componentes de UI | 101 |
| Definição de estilos | 101 |
| Sobre capas | 104 |
| Personalizar o componente Button | 107 |
| Personalizar o componente CheckBox | 109 |
| Personalizar o componente ColorPicker | 110 |
| Personalizar o componente ComboBox | 112 |
| Personalizar o componente DataGrid | 114 |
| Personalizar o componente Label | 119 |
| Personalizar o componente List | 120 |
| Personalizar o componente NumericStepper | 123 |
| Personalizar o componente ProgressBar | 124 |
| Personalizar o componente RadioButton | 126 |
| Personalizar o componente ScrollPane | 128 |
| Personalizar o componente Slider | 129 |
| Personalizar o componente TextArea | 130 |
| Personalizar o componente TextInput | 132 |
| Personalizar o componente TileList | 134 |
| Personalizar o componente UILoader | 136 |
| Personalizar o componente UIScrollBar | 136 |

Capítulo 6: Uso do componente FLVPlayback

| | |
|---|-----|
| Usar o componente FLVPlayback | 139 |
| Personalizar o componente FLVPlayback | 157 |
| Usar um arquivo SMIL | 169 |

Capítulo 7: Uso do componente de legenda FLVPlayback

| | |
|--|-----|
| Uso do componente FLVPlaybackCaptioning | 177 |
| Uso de legendas Timed Text | 179 |
| Uso de pontos de sinalização com legendas | 184 |
| Execução de vários arquivos FLV com legendas | 187 |
| Personalização do componente FLVPlaybackCaptioning | 187 |

Capítulo 1: Introdução

O Adobe® Flash® CS5 Professional é a ferramenta de criação padrão para produzir experiências de alto impacto na Web. Componentes são os blocos de criação de aplicativos avançados da Internet que oferecem essas experiências. Um *componente* é um clipe de filme com parâmetros que permitem personalizar o componente durante a criação em Flash ou em tempo de execução com métodos, propriedades e eventos do Adobe® ActionScript®. Os componentes foram desenvolvidos para permitir que desenvolvedores reutilizem e compartilhem código, e para integrar funcionalidades complexas que podem ser usadas e personalizadas por designers sem o uso do ActionScript.

Os componentes permitem criar aplicativos de maneira fácil e rápida, com aparência e comportamento consistentes. Este manual descreve como criar aplicativos com componentes do Adobe ActionScript 3.0. A *Referência de Componentes e Linguagem do Adobe® ActionScript® 3.0* descreve cada API (Interface de programação de aplicativos) do componente.

Você pode usar componentes criados pela Adobe®, fazer download de componentes criados por outros desenvolvedores ou criar seus próprios componentes.

Público-alvo

Este manual destina-se a desenvolvedores que estejam criando aplicativos Flash e desejem utilizar componentes que agilizem o desenvolvimento. Você já deve estar familiarizado com o desenvolvimento de aplicativos em Flash e com a escrita de códigos ActionScript.

Se tiver menos experiência em escrever ActionScript, você pode adicionar componentes a um documento, ajustar os parâmetros no Inspetor de propriedade ou componente, e usar o painel Comportamentos para manipular os eventos. Por exemplo, você poderia anexar um comportamento Ir para página da Web para um componente Botão que abre um URL em um navegador quando se clica no botão sem escrever nenhum código ActionScript.

Se você for um programador que deseja criar aplicativos mais robustos, poderá criar os componentes dinamicamente, usar o ActionScript para ajustar as propriedades e métodos de chamada no momento da execução, e usar o modelo de ouvinte de evento para manipular os eventos.

Para obter mais informações, consulte [“Trabalho com componentes”](#) na página 18.

Requisitos do sistema

Os componentes Flash não têm nenhum requisito de sistema além dos requisitos padrão do Flash.

Qualquer arquivo SWF que utilize componentes do Flash CS3 ou superior deve ser visualizado com o Adobe® Flash® Player 9.0.28.0 ou superior, e deve ser publicado para o ActionScript 3.0 (você pode definir isso em Arquivo > Configurações de publicação, na guia Flash).

Sobre a documentação

Este documento explica os detalhes do uso de componentes para desenvolvimento de aplicativos Flash. Ele presume que você tenha um conhecimento geral de Flash e ActionScript 3.0. A documentação específica sobre o Flash e produtos relacionados está disponível separadamente.

Este documento está disponível como arquivo em PDF e como ajuda on-line. Para exibir a ajuda on-line, inicie o Flash e selecione Ajuda > Ajuda do Flash > Utilização dos componentes do Adobe ActionScript 3.0.

Para obter informações sobre o Flash, consulte os seguintes documentos:

- *Uso do Flash*
- *Guia do desenvolvedor do ActionScript 3.0*
- *Referência do Adobe ActionScript 3.0 para Flash Professional*

Convenções tipográficas

As seguintes convenções tipográficas são usadas neste manual:

- *Fonte em itálico* indica um valor que deve ser substituído (por exemplo, em um caminho de pasta).
- `Fonte de código` indica código ActionScript, incluindo nomes de método e propriedade.
- *Código com fonte em itálico* indica um item de código que deve ser substituído (por exemplo, um parâmetro do ActionScript).
- **Fonte em negrito** indica o valor que você insere.

Termos usados neste manual

Os seguintes termos são usados neste manual:

em tempo de execução Quando o código está sendo executado no Flash Player.

durante criação Enquanto você está trabalhando no ambiente de criação do Flash.

Recursos adicionais

Além do conteúdo desses manuais, a Adobe fornece artigos, conceitos de design e exemplos no Adobe Developer Center e no Adobe Design Center atualizados regularmente.

Você pode encontrar amostras adicionais de componentes em www.adobe.com/go/learn_fl_samples_br.

Adobe Developer Center

O Adobe Developer Center é o seu recurso para obter informações atualizadas sobre o ActionScript, artigos sobre desenvolvimento de aplicativos reais e informações sobre problemas importantes. Consulte o Developer Center em www.adobe.com/go/flash_devcenter_br.

Adobe Design Center

Saiba das novidades em design digital e gráficos de movimento. Navegue nos trabalhos dos principais artistas, descubra novas tendências de design e aprimore suas habilidades com tutoriais, fluxos de trabalho importantes e técnicas avançadas. Confira duas vezes por mês os tutoriais e artigos recentes, além de peças de galeria inspiradoras. Consulte o Design Center em www.adobe.com/go/fl_designcenter_br.

Capítulo 2: Sobre os componentes do ActionScript 3.0

Os componentes do Adobe® Flash® Professional CS5 são clipes de filmes com parâmetros que permitem modificar sua aparência e comportamento. Um componente pode ser um controle simples de interface do usuário, como um RadioButton ou uma CheckBox, ou então ter conteúdo, como uma List ou uma DataGrid.

Os componentes permitem criar com rapidez e facilidade Flash aplicativos robustos com comportamento e aparência consistentes. Em vez de criar botões, caixas de combinação e listas personalizadas, você pode usar os componentes do Flash que implementam esses controles. Basta arrastá-los do painel Componentes para o documento do seu aplicativo. Também é fácil personalizar a aparência desses componentes, ajustando-os ao design do seu aplicativo.

Embora seja possível fazer tudo isso sem um conhecimento avançado do ActionScript, você também pode usar o ActionScript 3.0 para modificar o comportamento de um componente ou implementar um novo comportamento. Cada componente tem um conjunto exclusivo de métodos, propriedades e eventos do ActionScript que compõem sua API (*interface de programação de aplicativo*). A API permite criar e manipular componentes com o aplicativo em execução.

A API também permite que você crie seus novos componentes personalizados. Você pode baixar componentes criados por membros da comunidade Flash no Adobe Exchange em www.adobe.com/go/flash_exchange_br. Para obter mais informações sobre a criação de um componente, consulte www.adobe.com/go/learn_fl_creating_components_br.

A arquitetura de componentes do ActionScript 3.0 inclui classes nas quais todos os componentes se baseiam, capas e estilos que permitem personalizar sua aparência, um modelo de manipulação de eventos, gerenciamento de foco, uma interface de acessibilidade etc.

Nota: O Adobe Flash CS5 inclui componentes do ActionScript 2.0 e também do ActionScript 3.0. Você não pode combinar esses dois conjuntos de componentes. É preciso usar um dos dois conjuntos em um determinado aplicativo. O Flash CS5 apresenta os componentes do ActionScript 2.0 ou do ActionScript 3.0, com base na abertura de um arquivo do ActionScript 2.0 ou do ActionScript 3.0. Quando você cria um novo documento Flash, precisa especificar Arquivo Flash (ActionScript 3.0) ou Arquivo Flash (ActionScript 2.0). Quando você abre um documento existente, o Flash examina as Configurações de publicação para determinar qual o conjunto de componentes a ser usado. Para obter informações sobre os componentes do ActionScript 2.0, consulte *Utilização dos componentes do Adobe® ActionScript® 2.0*.

Para obter uma lista completa dos componentes Flash do ActionScript 3.0, consulte “[Tipos de componentes](#)” na página 6.

Vantagens do uso de componentes

Os componentes permitem estabelecer uma separação entre o processo de design e o processo de codificação do seu aplicativo. Eles permitem aos desenvolvedores criar funções que os designers podem usar nos aplicativos. Os desenvolvedores podem encapsular funções usadas com frequência em componentes, e os designers podem personalizar o tamanho, a localização e o comportamento dos componentes alterando seus parâmetros. Podem também alterar a aparência de um componente editando seus elementos gráficos, ou capas.

Os componentes compartilham funções essenciais, como estilos, capas e gerenciamento de foco. Quando você adiciona o primeiro componente a um aplicativo, essa função essencial responde por cerca de 20 kilobytes do tamanho. Quando você adiciona outros componentes, essa alocação inicial de memória é compartilhada pelos componentes adicionados, reduzindo o aumento de tamanho do aplicativo.

Esta seção descreve algumas vantagens dos componentes do ActionScript 3.0.

A potência do ActionScript 3.0 proporciona uma linguagem de programação eficaz e orientada a objeto, que representa uma etapa importante na evolução dos recursos do Flash Player. A linguagem foi projetada para a criação de aplicativos avançados de Internet com uma base de código reutilizável. O ActionScript 3.0 é baseado no ECMAScript, a linguagem padronizada internacional para scripts, e é compatível com a especificação ECMAScript (ECMA-262) 3ª edição. Para obter uma introdução completa ao ActionScript 3.0, consulte o *Guia do Desenvolvedor do ActionScript 3.0 Developer's Guide*. Para obter informações de referência sobre a linguagem, consulte a [Referência do ActionScript® 3.0 para Adobe Flash® Professional CS5](#).

Os componentes de interface do usuário baseados em FLA permitem o acesso fácil às capas, simplificando a personalização durante a autoria. Esses componentes também fornecem estilos, inclusive estilos de capas, que permitem personalizar aspectos da aparência dos componentes e carregar capas em tempo de execução. Para obter mais informações, consulte “[Personalização dos componentes de UI](#)” na página 101 e a [Referência do ActionScript® 3.0 para Adobe Flash® Professional CS5](#).

O novo componente FVLPlayback adiciona o componente FLVPlaybackCaptioning, além de suporte a tela cheia, visualização ao vivo aperfeiçoada, capas que permitem adicionar configurações de cor e alfa e recursos aperfeiçoados de download e layout de FLV.

O Inspetor de propriedades e o Inspetor de componentes permitem alterar parâmetros de componentes durante a autoria no Flash. Para obter mais informações, consulte “[Trabalho com arquivos de componente](#)” na página 20 e “[Definição de parâmetros e propriedades](#)” na página 22.

A nova caixa de diálogo de conjunto para os componentes ComboBox, List e TileList permite preencher a propriedade `dataProvider` por meio da interface do usuário. Para obter mais informações, consulte “[Criar um DataProvider](#)” na página 31.

O modelo de eventos do ActionScript 3.0 permite que seu aplicativo monitore eventos e invoque manipuladores de eventos em resposta. Para obter mais informações, consulte “[Modelo de manipulação de eventos do ActionScript 3.0](#)” na página 10 e “[Manipulação de eventos](#)” na página 25.

As classes de gerenciadores são uma forma simples de manipular o foco e gerenciar os estilos em um aplicativo. Para obter mais informações, consulte a [Referência do ActionScript® 3.0 para Adobe Flash® Professional CS5](#).

A classe base UIComponent fornece métodos, propriedades e eventos essenciais para os componentes que a estendem. Todos os componentes de interface do usuário do ActionScript 3.0 herdam da classe UIComponent. Para obter mais informações, consulte a classe UIComponent [Referência do ActionScript® 3.0 para Adobe Flash® Professional CS5](#).

O uso de SWC nos componentes de UI baseados em FLA proporciona ao ActionScript definições como um ativo dentro da Linha do tempo do componente, agilizando a compilação.

Uma hierarquia de classes facilmente extensível usando o ActionScript 3.0 permite criar namespaces exclusivos, importar classes conforme necessário e criar subclasses com facilidade para estender componentes.

Para obter mais informações, consulte a [Referência do ActionScript® 3.0 para Adobe Flash® Professional CS5](#).

Nota: O Flash CS5 oferece suporte a componentes baseados em FLA e em SWC. Para obter mais informações, consulte “[Arquitetura do componente](#)” na página 18.

Tipos de componentes

Você instala os componentes Flash ao instalar o Flash CS5.

Entre os componentes do ActionScript 3.0 estão os seguintes componentes de UI (interface do usuário):

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

Além dos componentes de interface do usuário, os componentes Flash do ActionScript 3.0 incluem os seguintes componentes e classes de suporte:

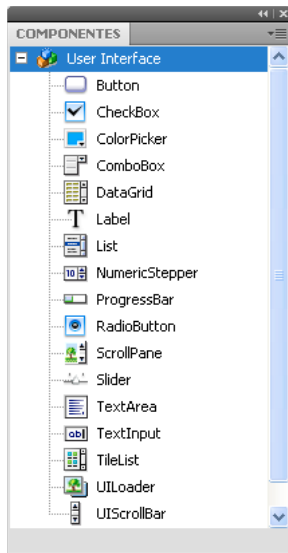
- Componente FLVPlayback (fl.video.FLVPlayback), baseado em SWC.
O componente FLVPlayback permite incluir imediatamente um player de vídeo no seu aplicativo Flash para reproduzir fluxo progressivo de vídeo via HTTP, a partir de um Adobe® FVSS (Flash® Video Streaming Service), ou então de um Macromedia® FMS (Flash® Media Server) da Adobe. Para obter mais informações, consulte “[Uso do componente FLVPlayback](#)” na página 139.
- Os componentes de UI personalizada FLVPlayback, que são baseados em FLA e funcionam com as versões ActionScript 2.0 e ActionScript 3.0 do componente FLVPlayback. Para obter mais informações, consulte “[Uso do componente FLVPlayback](#)” na página 139.
- O componente FLVPlayback Captioning, que fornece legendas para FLVPlayback. Consulte “[Uso do componente de legenda FLVPlayback](#)” na página 177.

Para obter uma lista completa dos componentes do ActionScript 3.0 e de suas classes de suporte, consulte a [Referência do ActionScript® 3.0 para Adobe Flash® Professional CS5](#).

Exibir componentes Flash:

Você pode exibir os componentes Flash do ActionScript 3.0 no painel Componentes seguindo estas etapas.

- 1 Inicie o Flash.
- 2 Crie um novo arquivo Flash (ActionScript 3.0) ou abra um documento Flash existente no qual as Configurações de publicação especifiquem o ActionScript 3.0.
- 3 Selecione Janela > Componentes para abrir o painel Componentes, se ainda não estiver aberto.



painel Componentes com componentes da interface do usuário

Você também pode baixar componentes adicionais do Adobe Exchange em www.adobe.com/go/flash_exchange_br. Para instalar componentes baixados do Exchange, baixe e instale o Adobe® Extension Manager em www.adobe.com/go/exchange_br. Clique no link Início do Adobe Exchange e procure o link do Extension Manager.

Qualquer componente pode aparecer no painel Componentes do Flash. Siga estas etapas para instalar componentes em um computador Windows® ou Macintosh®.

Instalar componentes em um computador baseado em Windows ou Macintosh:

- 1 Saia do Flash.
- 2 Insira o arquivo SWC ou FLA que contém o componente na seguinte pasta do seu disco rígido:
 - No Windows:
C:\Arquivos de programas\Adobe\Adobe Flash CS5\idioma\Configuration\Components
 - No Macintosh:
Macintosh HD:Aplicativos:Adobe Flash CS5:Configuration:Components
- 3 Inicie o Flash.
- 4 Selecione Janela > Componentes para exibir o componente no painel Componentes, se ele ainda não estiver aberto.
Para obter mais informações sobre arquivos de componentes, consulte “[Trabalho com arquivos de componente](#)” na página 20

Adicionar e excluir de um documento

Quando você arrasta um componente baseado em FLA do painel Componentes para o Palco, o Flash importa um clipe de filme editável para a biblioteca. Quando você arrasta um componente baseado em SWC para o Palco, o Flash importa um clipe compilado para a biblioteca. Depois que um componente tiver sido importado para a biblioteca, você poderá arrastar ocorrências dele para o Palco do painel Biblioteca ou do painel Componentes.

Adicionar componentes durante a autoria

Você pode adicionar um componente a um documento arrastando-o do painel Componentes. É possível definir propriedades para cada ocorrência de um componente no Inspetor de propriedades ou no Inspetor de componentes, na guia Parâmetros.

- 1 Selecione Janela > Componentes.
- 2 Clique duas vezes no componente no painel Componentes ou arraste-o para o Palco.
- 3 Selecione o componente no Palco.
- 4 Se o Inspetor de propriedades não estiver visível, selecione Janela > Propriedades > Propriedades.
- 5 No Inspetor de propriedades, digite um nome de ocorrência para a ocorrência do componente.
- 6 Selecione Janela > Inspetor de componentes e selecione a guia Parâmetros para especificar os parâmetros da ocorrência.
Para obter mais informações, consulte [“Definição de parâmetros e propriedades”](#) na página 22.
- 7 Altere o tamanho do componente conforme desejado, editando os valores de largura (W:) e altura (H:).
Para obter mais informações sobre o dimensionamento de tipos de componentes específicos, consulte [“Personalização dos componentes de UI”](#) na página 101.
- 8 Selecione Controlar > Testar filme ou pressione Control+Enter para compilar o documento e ver os resultados das suas configurações.

Você também pode alterar a cor e a formatação de texto de um componente definindo suas propriedades de estilo, ou então personalizar sua aparência editando as capas. Para obter mais informações sobre esses tópicos, consulte [“Personalização dos componentes de UI”](#) na página 101.

Se você arrastar um componente para o Palco durante a autoria, poderá fazer referência a ele usando o nome da ocorrência (por exemplo, `myButton`).

Adicionar componentes em tempo de execução com o ActionScript

Para adicionar um componente a um documento em tempo de execução com o ActionScript, é preciso primeiro que o componente esteja na biblioteca do aplicativo (Janela > Biblioteca) quando o arquivo SWF for compilado. Para adicionar um componente à biblioteca, arraste-o do painel Componentes para o painel Biblioteca. Para obter mais informações sobre a biblioteca, consulte [“A biblioteca”](#) na página 24.

Você precisa também importar o arquivo de classe do componente para tornar sua API disponível para o aplicativo. Os arquivos de classe de componentes são instalados em *pacotes* contendo uma ou mais classes. Para importar a classe de um componente, use a instrução `import` e especifique o nome do pacote e o nome da classe. Por exemplo, você importaria a classe `Button` com a seguinte instrução `import`:

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

Para obter informações sobre o pacote em que um componente se encontra, consulte a [Referência do ActionScript® 3.0 para Adobe Flash® Professional CS5](#). Para obter informações sobre a localização de arquivos de origem de componentes, consulte [“Trabalho com arquivos de componente”](#) na página 20.

Para criar uma ocorrência do componente, você precisa invocar o método do construtor do ActionScript para o componente. Por exemplo, a instrução a seguir cria uma ocorrência de `Button` chamada `aButton`:

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

A etapa final consiste em chamar o método `addChild()` estático para adicionar a ocorrência do componente ao Palco ou ao contêiner do aplicativo. Por exemplo, a instrução a seguir adiciona a ocorrência `aButton`:

```
addChild(aButton);
```

Neste ponto, você pode usar a API do componente para especificar dinamicamente o tamanho do componente e sua posição no Palco, monitorar eventos e definir propriedades para modificar seu comportamento. Para obter mais informações sobre a API de um determinado componente, consulte a [Referência do ActionScript® 3.0 para Adobe Flash® Professional CS5](#).

Para obter mais informações sobre o método `addChild()`, consulte “[Trabalho com a lista de exibição](#)” na página 27.

Excluir um componente

Para excluir uma ocorrência de componente do Palco durante a autoria, basta selecioná-la e pressionar a tecla Delete. Isso removerá a ocorrência do Palco, mas não removerá o componente do seu aplicativo.

Para excluir um componente do seu documento Flash depois de inseri-lo no Palco ou na biblioteca, você precisa excluir o componente e seus ativos associados da biblioteca. Não basta excluir o componente do Palco. Se não for removido da biblioteca, ele será incluído no seu aplicativo quando você o compilar.

- 1 No painel Biblioteca, selecione o símbolo do componente.
- 2 Clique no botão Excluir, na parte inferior do painel Biblioteca, ou então selecione Excluir no menu do painel Biblioteca.

Repita essas etapas para excluir todos os ativos associados ao componente.

Para obter mais informações sobre como remover um componente de seu contêiner com o aplicativo em execução, consulte “[Remoção de um componente da lista de exibição](#)” na página 28.

Localizar a versão do componente

Os componentes Flash do ActionScript 3.0 têm uma propriedade de versão que você pode exibir, se precisar fornecê-la ao Suporte técnico da Adobe, ou então se precisar saber qual versão do componente está usando.

Exibir o número da versão de um componente de interface do usuário:

- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste o componente para o Palco e atribua a ele um nome de ocorrência. Por exemplo, arraste uma ComboBox para o Palco e chame-a de **aCb**.
- 3 Pressione a tecla **F9** ou selecione Janela > Ações para abrir o painel Ações.
- 4 Clique no Quadro 1 da Linha do tempo principal e adicione o seguinte código ao painel Ações:

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

O número da versão, semelhante ao mostrado na ilustração a seguir, deverá aparecer no painel Saída.

No caso dos componentes `FLVPlayback` e `FLVPlaybackCaptioning`, é preciso fazer referência ao nome da classe, e não ao nome da ocorrência, porque o número da versão é armazenado em uma constante de classe.

Exibir o número da versão dos componentes `FLVPlayback` e `FLVPlaybackCaptioning`:

- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste os componentes `FLVPlayback` e `FLVPlaybackCaptioning` para o painel Biblioteca.
- 3 Pressione a tecla **F9** ou selecione Janela > Ações para abrir o painel Ações.

4 Clique no Quadro 1 da Linha do tempo principal e adicione o código a seguir ao painel Ações.

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

Os números de versão aparecerão no painel Saída.

Modelo de manipulação de eventos do ActionScript 3.0

O ActionScript 3.0 introduz um único modelo de manipulação de eventos que substitui os diferentes mecanismos de manipulação de eventos existentes nas versões anteriores do ActionScript. O novo modelo de evento baseia-se na especificação de eventos DOM nível 3.

Para desenvolvedores familiarizados com o método `addListener()` do ActionScript 2.0, talvez seja útil descrever as diferenças entre o modelo de ouvinte de evento do ActionScript 2.0 e o modelo de evento do ActionScript 3.0. A lista a seguir descreve algumas das principais diferenças entre os dois modelos de eventos:

- Para adicionar ouvintes de eventos no ActionScript 2.0, utiliza-se `addListener()` em alguns casos e `addEventListener()` em outros. Já no ActionScript 3.0, utiliza-se `addEventListener()` em todos os casos.
- Não há fluxo de eventos no ActionScript 2.0. Isso significa que o método `addListener()` só pode ser chamado no objeto que transmite o evento, enquanto no ActionScript 3.0 o método `addEventListener()` pode ser chamado em qualquer objeto que faça parte do fluxo de eventos.
- No ActionScript 2.0, os ouvintes de evento podem ser funções, métodos ou objetos e, no ActionScript 3.0, apenas funções ou métodos podem ser ouvintes de evento.
- A sintaxe `on(event)` não tem mais suporte no ActionScript 3.0. Então, não é possível anexar um código de evento do ActionScript a um clipe de filme. Você só pode usar `addEventListener()` para adicionar um ouvinte de eventos.

O exemplo a seguir, que monitora um evento `MouseEvent.CLICK` em um componente `Button` chamado `aButton`, ilustra o modelo básico de manipulação de eventos do 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);
}
```

Para obter mais informações sobre a manipulação de eventos do ActionScript 3.0, consulte *Programação do ActionScript 3.0*. Para obter mais informações sobre a manipulação de eventos do ActionScript 3.0 para componentes, consulte [“Manipulação de eventos”](#) na página 25.

Um aplicativo simples

Esta seção descreve as etapas para a criação de um aplicativo simples do ActionScript 3.0 usando componentes Flash e a ferramenta de autoria do Flash. O exemplo é fornecido como um arquivo FLA, com o código do ActionScript incluído na Linha do tempo, e também como um arquivo de classe externo do ActionScript, com um arquivo FLA que contém somente os componentes na biblioteca. De forma geral, é provável que você queira desenvolver aplicativos maiores usando arquivos de classe externos, para poder compartilhar código entre classes e aplicativos e facilitar a manutenção dos seus aplicativos. Para obter mais informações sobre a programação com o ActionScript 3,0, consulte *Programação do ActionScript 3.0*.

Design do aplicativo

Nosso primeiro exemplo de aplicativo com componentes do ActionScript é uma variação do aplicativo “Hello World” padrão, com um design bastante simples:

- O aplicativo será chamado de Greetings.
- Ele utiliza uma TextArea para exibir uma saudação que é, inicialmente, Hello World.
- Ele utiliza um ColorPicker que permite alterar a cor do texto.
- Ele utiliza três RadioButtons que permitem definir o tamanho do texto como pequeno, grande ou muito grande.
- Ele utiliza uma ComboBox que permite selecionar uma saudação diferente em uma lista suspensa.
- O aplicativo utiliza componentes do painel Componentes, e também cria elementos do aplicativo com o código do ActionScript.

A partir dessa definição, você pode começar a criar o aplicativo.

Criar o aplicativo Greetings

As etapas a seguir criam o aplicativo Greetings usando a ferramenta de autoria do Flash para criar um arquivo FLA, inserir componentes no Palco e adicionar código do ActionScript à Linha do tempo.

Criar o aplicativo Greetings em um arquivo FLA:

- 1 Selecione Arquivo > Novo.
- 2 Na caixa de diálogo Novo documento, selecione Arquivo Flash (ActionScript 3.0) e clique em OK.
Uma nova janela do Flash é aberta.
- 3 Selecione Arquivo > Salvar, nomeie o arquivo Flash como **Greetings.fla** e clique no botão Salvar.
- 4 No painel Componentes do Flash, selecione um componente TextArea e arraste-o para o Palco.
- 5 Na janela Propriedades, com a TextArea selecionada no Palco, digite **aTa** como nome da ocorrência e insira as seguintes informações:
 - Digite **230** como valor L (largura).
 - Digite **44** como valor A (altura).
 - Digite **165** como valor X (posição horizontal).
 - Digite **57** como valor Y (posição vertical).
 - Digite **Hello World!** para o parâmetro de texto, na guia Parâmetros.

- 6 Arraste um componente `ColorPicker` para o Palco, insira-o à esquerda da `TextArea` e atribua a ele um nome de ocorrência de `txtCp`. Digite as seguintes informações no Inspetor de propriedades:
 - Digite **96** como valor X.
 - Digite **72** como valor Y.
- 7 Arraste três componentes `RadioButton` para o Palco, um de cada vez, e atribua a eles os nomes de ocorrência de `smallRb`, `largerRb` e `largestRb`. Digite as seguintes informações no Inspetor de propriedades:
 - Digite **100** como valor L e **22** para o valor A de cada um deles.
 - Digite **155** como valor X.
 - Digite **120** como valor Y de `smallRb`, **148** para `largerRb` e **175** para `largestRb`.
 - Digite `fontRbGrp` para o parâmetro `groupName` de cada um deles.
 - Digite na guia Parâmetros os rótulos **Small**, **Larger**, **Largest**.
- 8 Arraste uma `ComboBox` para o Palco e atribua a ela o nome de ocorrência de `msgCb`. Digite as seguintes informações no Inspetor de propriedades:
 - Digite **130** como valor L.
 - Digite **265** como valor X.
 - Digite **120** como valor Y.
 - Na guia Parâmetros, digite **Greetings** para o parâmetro do prompt.
 - Clique duas vezes no campo de texto do parâmetro `dataProvider` para abrir a caixa de diálogo Valores.
 - Clique no sinal de adição e substitua o valor do rótulo por **Hello World!**
 - Repita a etapa anterior para adicionar os valores de rótulos **Have a nice day!** e **Top of the Morning!**
 - Clique em OK para fechar a caixa de diálogo Valores.
- 9 Salve o arquivo.
- 10 Abra o painel Ações, se ainda não estiver aberto, pressionando **F9** ou selecionando Ações no menu Janela. Clique no Quadro 1 da Linha do tempo principal e insira o seguinte código no painel Ações:

```
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);
```

As três primeiras linhas importam as classes de eventos usadas pelo aplicativo. Um evento ocorre quando um usuário interage com um dos componentes. As cinco linhas seguintes registram manipuladores de eventos para os eventos que o aplicativo deseja monitorar. Um evento `click` ocorre em um `RadioButton` quando um usuário clica nele. Um evento `change` ocorre quando um usuário seleciona uma cor diferente no `ColorPicker`. Um evento `change` ocorre na `ComboBox` quando um usuário escolhe uma saudação diferente na lista suspensa.

A quarta linha importa a classe `RadioButtonGroup`, de forma que o aplicativo possa atribuir um ouvinte de eventos ao grupo de `RadioButtons`, em vez de atribuir o ouvinte a cada botão individualmente.

- 11 Adicione a linha de código a seguir ao painel Ações para criar o objeto `TextFormat tf`, que o aplicativo usa para alterar as propriedades de estilo `size` e `color` do texto na `TextArea`.


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

- 12 Adicione o código a seguir para criar a função de manipulação de eventos `rbHandler`. Essa função manipula um evento `click` quando um usuário clica em um dos componentes `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);
}
```

Essa função usa uma instrução `switch` para examinar a propriedade `target` do objeto `event`, a fim de determinar qual `RadioButton` acionou o evento. A propriedade `currentTarget` contém o nome do objeto que acionou o evento. Dependendo do `RadioButton` no qual o usuário clicou, o aplicativo altera o tamanho do texto na `TextArea` para 14, 18 ou 24 pontos.

- 13 Adicione o código a seguir para implementar a função `cpHandler()`, que manipula uma alteração no valor do `ColorPicker`:

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

Essa função define a propriedade `color` do objeto `TextFormat` `tf` como a cor selecionada no `ColorPicker`, e depois chama `setStyle()` para aplicá-la ao texto na ocorrência de `TextArea` `aTa`.

- 14 Adicione o código a seguir para implementar a função `cbHandler()`, que manipula uma alteração na seleção na `ComboBox`:

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

Essa função simplesmente substitui o texto na `TextArea` pelo texto selecionado na `ComboBox`, `event.target.selectedItem.label`.

- 15 Selecione `Controlar > Testar filme` ou pressione `Control+Enter` para compilar o código e testar o aplicativo `Greetings`.

A próxima seção mostra como criar o mesmo aplicativo com uma classe `ActionScript` externa e um arquivo `FLA` que tem apenas os componentes necessários na biblioteca.

Criar o aplicativo `Greetings2` com um arquivo de classe externo:

- 1 Selecione `Arquivo > Novo`.
- 2 Na caixa de diálogo `Novo documento`, selecione `Arquivo Flash (ActionScript 3.0)` e clique em `OK`.
Uma nova janela do `Flash` é aberta.
- 3 Selecione `Arquivo > Salvar`, nomeie o arquivo `Flash` como **`Greetings2 fla`** e clique no botão `Salvar`.

4 Arraste cada um dos seguintes componentes do painel Componentes para a biblioteca:

- ColorPicker
- ComboBox
- RadioButton
- TextArea

O arquivo SWF compilado utilizará todos esses ativos. Então, você precisa adicioná-los à biblioteca. Arraste os componentes para a parte inferior do painel Biblioteca. Conforme você adiciona esses componentes à biblioteca, outros ativos (como List, TextInput e UIScrollBox) são adicionados automaticamente.

5 Na janela Propriedades, em Classe do documento, digite **Greetings2**.

Se o Flash exibir um aviso informando que " não foi possível localizar uma definição para a classe do documento", ignore-o. Você definirá a classe de Greetings2 nas etapas seguintes. Essa classe define a função principal do aplicativo.

6 Salve o arquivo Greetings2.fla.

7 Selecione Arquivo > Novo.

8 Na caixa de diálogo Novo documento, selecione Arquivo ActionScript e clique em OK.

Uma nova janela de script é aberta.

9 Adicione o código a seguir à janela de script:

```
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() {
```

O script define uma classe do ActionScript 3.0 denominada Greetings2. O script faz o seguinte:

- Importa classes que serão usadas no arquivo. Normalmente, você adicionaria essas instruções `import` à medida que fizesse referência às diferentes classes no código. Mas, para abreviar o processo, este exemplo importa todas em uma só etapa.
- Declara variáveis representando os diversos tipos de objetos componentes que serão acrescentados ao código. Outra variável cria o objeto `TextFormat` `tf`.

- Defina uma função de construtor, `Greetings2()`, para a classe. Nas etapas seguintes, adicionaremos linhas a essa função e outros métodos à classe.

10 Selecione Arquivo > Salvar, nomeie o arquivo como **Greetings2.fla** e clique no botão Salvar.

11 Adicione as seguintes linhas de código à função `Greeting2()`:

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

Agora, a função deve ter a seguinte aparência:

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

12 Adicione as linhas de código a seguir depois da chave de fechamento do método `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);
    }
```

Essas linhas fazem o seguinte:

- Instanciam os componentes usados no aplicativo.
- Definem o tamanho, a posição e as propriedades de cada componente.
- Adicionam todos os componentes ao Palco usando o método `addChild()`.

13 Depois da chave de fechamento do método `blDRadioButton()`, adicione o seguinte código ao método `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;
}
}
}
```

Essas funções definem manipuladores de eventos para os componentes.

14 Selecione Arquivo > Salvar para salvar o arquivo.

15 Selecione Controlar > Testar filme ou pressione Control+Enter para compilar o código e testar o aplicativo Greetings2.

Desenvolver e executar os exemplos subsequentes

Tendo desenvolvido e executado o aplicativo Greetings, você conta com o conhecimento básico necessário para executar os outros exemplos de código apresentados neste livro. O código relevante do ActionScript 3.0 em cada exemplo será destacado e debatido, e você deverá ser capaz de recortar e colar em um arquivo FLA, compilar e executar cada um dos exemplos deste livro.

Capítulo 3: Trabalho com componentes

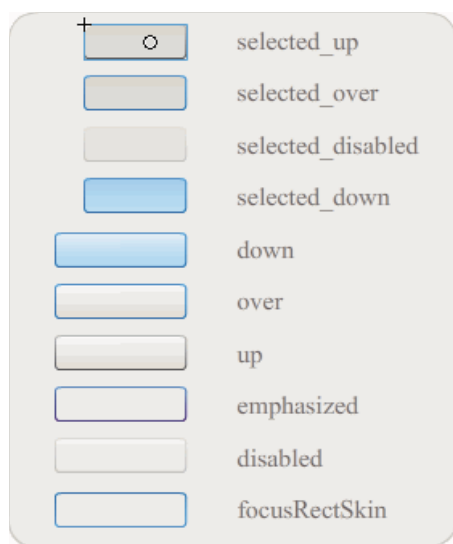
Arquitetura do componente

Os componentes do Adobe® ActionScript® 3.0 são suportados pelo Adobe® Flash Player 9.0.28.0 e superior. Esses componentes não são compatíveis com os componentes criados antes do Flash CS4. Para obter informações sobre o uso dos componentes do Adobe® ActionScript® 2.0, consulte *Utilização dos componentes do Adobe® ActionScript® 2.0 e Referência de Componentes e Linguagem do Adobe® ActionScript® 2.0*.

Os componentes da interface do usuário (UI) do Adobe ActionScript 3.0 são implementados como componentes baseados em FLA, mas o Flash CS5 oferece suporte componentes baseados tanto em SWC quanto em FLA. Os componentes FLVPlayback e FLVPlaybackCaptioning são baseados em SWC, por exemplo. Você pode colocar qualquer tipo de componente no painel Componentes para que ele apareça nesse painel. Esses dois tipos de componentes são criados de forma diferente; por isso, eles são descritos separadamente aqui.

Componentes do ActionScript 3.0 baseados em FLA

Os componentes de interface de usuário do ActionScript 3.0 são arquivos baseados em FLA (.fla) com capas internas que você pode acessar para fins de edição clicando duas vezes no componente no Palco. As capas do componente e os outros ativos são colocados no Quadro 2 da Linha do tempo. Quando você clica duas vezes no componente, o Flash salta automaticamente para o Quadro 2 e abre uma paleta das capas do componente. A ilustração a seguir mostra a paleta com capas exibidas para o componente Button.



Capas do componente Button

Para obter mais informações sobre as capas dos componentes e a personalização dos componentes, consulte [“Personalização dos componentes de UI”](#) na página 101 e [“Personalizar o componente FLVPlayback”](#) na página 157.

Para agilizar a compilação de seus aplicativos e evitar conflitos com as configurações do ActionScript 3.0, os componentes da interface de usuário do Flash CS5 baseados em FLA também contêm um SWC que possui o código do ActionScript já compilado do componente. O ComponentShim SWC é colocado no Palco no Quadro 2 em cada componente da interface de usuário para disponibilizar as definições pré-compiladas. Para ser disponibilizado para o ActionScript, um componente deve estar no Palco ou estar na biblioteca com a opção Exportar no primeiro quadro selecionada em suas propriedades de ligação. Para criar um componente usando o ActionScript, você também deve importar a classe com uma instrução `import` para acessá-la. Para obter informações sobre a instrução `import`, consulte a [Referência do ActionScript 3.0 para Flash Professional](#).

Componentes baseados em SWC

Os componentes baseados em SWC têm um arquivo FLA e um arquivo de classe do ActionScript, mas eles foram compilados e exportados como um SWC. Um arquivo SWC é um pacote de símbolos Flash e código do ActionScript pré-compilados que evita a recompilação dos símbolos e do código que não serão alterados.

Os componentes FLVPlayback e FLVPlaybackCaptioning são baseados em SWC. Eles têm capas externas, e não internas. O componente FLVPlayback tem uma capa padrão que você pode alterar selecionando a que deseja dentre um conjunto de capas predefinidas, personalizando controles a partir dos controles de UI do painel Componentes (BackButton, BufferingBar, etc.) ou criando uma capa personalizada. Para obter mais informações, consulte “[Personalizar o componente FLVPlayback](#)” na página 157.

No Flash, você pode converter um clipe de filme em um clipe compilado, da seguinte maneira:

Compilação de um clipe de filme

- Clique com o botão direito do mouse (Windows) ou mantenha pressionada a tecla Control enquanto clica (Macintosh) no clipe de filme no painel Biblioteca e selecione Converter em clipe compilado.

O clipe compilado tem o mesmo comportamento do clipe de filme a partir do qual ele foi compilado, mas os cliques compilados são exibidos e publicados com muito mais rapidez do que os cliques de filme comuns. Os cliques compilados não podem ser editados, mas suas propriedades podem aparecer no Inspetor de propriedades e no Inspetor de componentes.

Os componentes SWC contêm um clipe compilado, as definições pré-compiladas do ActionScript do componente e outros arquivos que descrevem o componente. Se você criar seu próprio componente, poderá exportá-lo como um arquivo SWC para distribuí-lo.

Exportação de um arquivo SWC

- Selecione o clipe de filme no painel Biblioteca e clique com o botão direito do mouse (Windows) ou mantenha pressionada a tecla Control enquanto clica (Macintosh) e selecione Exportar arquivo SWC.

Nota: O formato de um arquivo Flash CS4 ou de um arquivo SWC posterior é compatível com o formato Flex SWC a fim de que os arquivos SWC possam ser trocados entre os dois produtos, mas não necessariamente sem modificações.

Para obter informações sobre a criação de componentes baseados em SWC, consulte www.adobe.com/go/learn_fl_creating_components_br.

A API dos componentes do ActionScript 3.0

Cada componente do ActionScript 3.0 é criado em uma classe do ActionScript 3.0 que está localizada em uma pasta de pacote e tem um nome no formato `fl.pacotename.classname`. O componente Button, por exemplo, é uma ocorrência da classe Button e tem o nome de pacote `fl.controls.Button`. Você deve fazer referência ao nome do pacote ao importar uma classe de componente no aplicativo. Você importará a classe Button com a seguinte instrução:

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

Para obter mais informações sobre o local dos arquivos de classe do componente, consulte “[Trabalho com arquivos de componente](#)” na página 20.

A classe de um componente define os métodos, as propriedades, os eventos e os estilos que permitem a você interagir com ela no seu aplicativo. Os componentes de UI do ActionScript 3.0 são subclasses das classes Sprite e UIComponent; eles herdam propriedades, métodos e eventos dessas classes. A classe Sprite é o bloco de construção básico de lista de exibição; ele é similar a um MovieClip, mas não tem uma linha do tempo. A classe UIComponent é a classe base para todos os componentes visuais, tanto interativos quanto não interativos. O caminho de herança de cada componente, bem como suas propriedades, métodos, eventos e estilos são descritos na *Referência do ActionScript 3.0 para Flash Professional*, da Adobe.

Todos os componentes do ActionScript 3.0 usam o modelo de manipulação de eventos do ActionScript 3.0. Para obter mais informações sobre a manipulação de eventos, consulte “[Manipulação de eventos](#)” na página 25 e *Programação em ActionScript 3.0*.

Trabalho com arquivos de componente

Esta seção mostra em que local os arquivos de componente são armazenados, em que local os arquivos de origem do ActionScript podem ser encontrados e como adicionar e remover componentes do painel Componentes.

Onde os arquivos de componente são armazenados

Os componentes Flash são armazenados na pasta Configuração no nível do aplicativo.

Nota: Para obter informações sobre essas pastas, consulte “*Pastas de configuração instaladas com o Flash*” em *Uso do Flash*.

Os componentes são instalados nos seguintes locais:

- Windows 2000 ou Windows XP: C:\Arquivos de Programas\Adobe\Adobe Flash CS5\idioma\Configuration\Components
- Mac OS X: Macintosh HD:Aplicativos:Adobe Flash CS5:Configuration:Components

Na pasta de componentes, os componentes da interface de usuário (UI) estão no arquivo User Interface.fla e os componentes FLVPlayback (FLVPlaybackAS3.swc) e FLVPlaybackCaptioning estão na pasta de vídeo.

Você também pode armazenar componentes nos seguintes locais baseados em usuário:

- Windows 2000 ou Windows XP: C:\Documents and Settings\nomedousuário\Configurações locais\Dados de aplicativos\Adobe\Adobe Flash CS5\en\Configuration\Components
- Windows Vista: C:\Users\nomedousuário\Configurações Locais\Dados de Aplicativos\Adobe\Adobe Flash CS5\en\Configuration\Components

Nota: No Windows, a pasta *Application Data* fica oculta, por padrão. Para mostrar pastas e arquivos ocultos, selecione *Meu Computador para abrir o Windows Explorer*, selecione *Ferramentas > Opções de Pasta* e selecione a *guia Exibir*. Na *guia Exibir*, selecione o botão de opção *Mostrar pastas e arquivos ocultos*.

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

Onde os arquivos de origem do componente são armazenados

Os arquivos de classe do ActionScript (.as) (ou *arquivos de origem*) dos componentes são instalados nas seguintes pastas de aplicativo para Windows 2000 ou Windows XP:

Componentes da interface do usuário C:\Arquivos de Programas\Adobe\Adobe Flash CS5\en\Configuration\Component Source\ActionScript 3.0\User Interface\fl

FLVPlayback C:\Arquivos de Programas\Adobe\Adobe Flash CS5\en\Configuration\Component Source\ActionScript 3.0\FLVPlayback\fl\video

FLVPlaybackCaptioning C:\Arquivos de Programas\Adobe\Adobe Flash CS5\en\Configuration\Component Source\ActionScript 3.0\FLVPlaybackCaptioning\fl\video

Para Mac OS X, os arquivos de origem do componente ficam no seguinte local:

Componentes da interface do usuário Macintosh HD:Aplicativos:Adobe Flash CS5:Configuration:Component Source>ActionScript 3.0>User Interface:fl

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

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

Arquivos de origem do componente e Classpath

Como o código dos componentes do ActionScript 3.0 é compilado, você não deve especificar o local dos arquivos de classe do ActionScript na variável Classpath. Se você incluir esse local na Classpath, ele aumentará o tempo necessário para compilar os aplicativos. No entanto, se o Flash localizar os arquivos de classe de componente na configuração Caminho de classe, o arquivo de classe sempre terá precedência sobre o código compilado do componente.

Uma ocasião em que você talvez queira adicionar o local dos arquivos de origem de componente à configuração Caminho de classe é quando estiver depurando um aplicativo com componentes. Para obter mais informações, consulte [“Depuração dos aplicativos de componente”](#) na página 22.

Modificação dos arquivos de componente

Se você atualizar, adicionar ou remover componentes baseados em SWC ou adicionar novos componentes baseados em FLA, deverá recarregá-los no painel Componentes para disponibilizá-los. Você pode recarregar os componentes reiniciando o Flash ou selecionando Recarregar no menu do painel Componentes. Isso fará com que o Flash selecione qualquer componente que você adicionou à pasta Componentes.


Recarregamento dos componentes no painel Componentes enquanto o Flash está em execução:

- Selecione Recarregar no menu do painel Componentes.

Remoção de um componente do painel Componentes:

- Remova o arquivo FLA, SWC ou MXP da pasta de componentes e reinicie o Flash ou selecione Recarregar no menu do painel Componentes. Um arquivo MXP é um arquivo de componente baixado do Adobe Exchange.

Você pode remover e substituir os componentes baseados em SWC enquanto o Flash está em execução, e o recarregamento refletirá as alterações; mas, se você alterar ou excluir os componentes baseados em FLA, as alterações só serão refletidas quando você terminar e reiniciar o Flash. No entanto, você pode adicionar componentes baseados em FLA e carregá-los com o comando Recarregar.

 A Adobe recomenda que, primeiro, você faça uma cópia de qualquer arquivo de componente Flash (.fla ou .as) que pretende alterar. Em seguida, você poderá restaurá-lo, se necessário.

Depuração dos aplicativos de componente

Os componentes do ActionScript 3.0 contêm todo o código de origem para reduzir o tempo de compilação do aplicativo. O depurador Flash, no entanto, não pode inspecionar o código nos clipes compilados. Portanto, se você deseja depurar o aplicativo para o código de origem dos componentes, adicione os arquivos de origem do componente à configuração Caminho de classe.

O local das pastas de pacote do componente é relativo ao local dos arquivos de origem do tipo de componente. Para fazer referência a todos os arquivos de origem do ActionScript 3.0 de todos os componentes de UI, adicione o seguinte local ao caminho de classe dos pacotes de interface de usuário:

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

Nota: Isso substituirá o código compilado de todos os componentes de UI e aumentará o tempo de compilação do aplicativo. Se você tiver alterado o arquivo de origem de um componente por qualquer motivo, esse componente poderá demonstrar um comportamento diferente como resultado.

Para definir o caminho de classe, selecione Preferências no menu Editar, selecione ActionScript na lista Categoria e clique no botão Configurações do ActionScript 3.0. Para adicionar uma nova entrada, clique no botão de adição acima da janela que exibe as configurações atuais.

A variável `$(AppConfig)` se refere à pasta de configuração do Flash CS5 no local em que você instalou o Flash CS5. Geralmente, o caminho é o seguinte:

- Windows 2000 ou Windows XP: C:\Arquivos de Programas\Adobe\Adobe Flash CS5\idioma\Configuration\
- Mac OS X: Macintosh HD:Aplicativos:Adobe Flash CS5:Configuration

Nota: Se você tiver que alterar um arquivo de origem de componente, a Adobe recomenda a cópia do arquivo de origem para outro local e a inclusão desse local no caminho de classe.

Para obter mais informações sobre o local dos arquivos de origem do componente, consulte “[Onde os arquivos de origem do componente são armazenados](#)” na página 21.

Definição de parâmetros e propriedades

Cada componente tem parâmetros que você pode definir para alterar sua aparência e comportamento. Um parâmetro é uma propriedade da classe do componente; ele aparece no Inspetor de propriedades e no Inspetor de componentes. As propriedades mais usadas aparecem como parâmetros de criação; as outras devem ser definidas com o ActionScript. Todos os parâmetros que podem ser definidos durante a criação também podem ser definidos com o ActionScript. A definição de um parâmetro com o ActionScript substitui qualquer valor definido durante a criação.

A maioria dos componentes de interface de usuário do ActionScript 3.0 herda as propriedades e os métodos da classe `UIComponent`, bem como de uma classe base. Por exemplo, as classes `Button` e `CheckBox` herdam as propriedades das classes `UIComponent` e `BaseButton`. As propriedades herdadas de um componente, bem como suas propriedades de classe, estarão disponíveis para que você possa acessá-las. Por exemplo, o componente `ProgressBar` herda a propriedade `ProgressBar.enabled` da classe `UIComponent`, mas também tem sua propriedade `ProgressBar.percentComplete`. Você pode acessar essas duas propriedades para interagir com uma ocorrência do componente `ProgressBar`. Para obter mais informações sobre as propriedades de um componente, consulte sua entrada de classe na [Referência do ActionScript 3.0](#).

Você pode definir parâmetros para uma ocorrência de componente usando o Inspetor de propriedades ou o Inspetor de componentes.

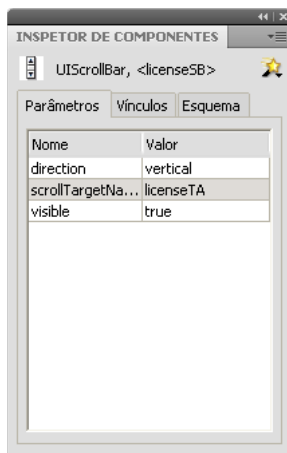
Especificação de um nome de ocorrência para um componente no Inspetor de propriedades:

- 1 Selecione Janela > Propriedades > Propriedades.
- 2 Selecione uma ocorrência de um componente no Palco.
- 3 Digite um nome para a ocorrência de componente na caixa <Nome da ocorrência>, localizada abaixo da lista suspensa Clipe de Filme. Uma outra alternativa é clicar na aba Parâmetros e digitar o nome na caixa abaixo da palavra *Componente*. Digite valores para qualquer parâmetro a ser definido.

Recomenda-se adicionar um sufixo ao nome da ocorrência para indicar o tipo do componente. Isso facilitará a leitura do código do ActionScript. Por exemplo, o nome de ocorrência **licenseSb** identifica que um componente é uma barra de rolagem que percorre um contrato de licença na área de texto **licenseTa**.

Especificação de parâmetros para uma ocorrência de componente no Inspetor de componentes:

- 1 Selecione Janela > Inspetor de componentes.
- 2 Selecione uma ocorrência de um componente no Palco.
- 3 Clique na guia Parâmetros e insira os valores de qualquer um dos parâmetros listados.



Parâmetros de componente no Inspetor de componentes

Definição das propriedades de componente no ActionScript

No ActionScript, você usa um operador de ponto (.) (sintaxe de pontos) para acessar propriedades ou métodos que pertencem a um objeto ou ocorrência no Palco. Uma expressão de sintaxe de pontos começa com o nome da ocorrência, seguido por um ponto, e termina com o elemento que você deseja especificar. Por exemplo, o código do ActionScript a seguir define a propriedade `width` da ocorrência de `Checkbox` `aCh` para deixá-la com a largura de 50 pixels:

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

A instrução `if` a seguir verifica se o usuário marcou a caixa de seleção:

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

A biblioteca

Ao acrescentar primeiro um componente a um documento, o Flash o importa como clipe de filme para o painel Biblioteca. Você também pode arrastar um componente do painel Componente diretamente para o painel Biblioteca e depois acrescentar uma instância dele ao Palco. De qualquer forma, você deve adicionar um componente à biblioteca para poder acessar seus elementos de classe.

Se você adicionar um componente à biblioteca e criar uma ocorrência desse componente usando o ActionScript, importe primeiro sua classe com a instrução `import`. Na instrução `import`, especifique o nome do pacote do componente e seu nome de classe. Por exemplo, esta instrução importa a classe `Button`:

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

Quando você coloca um componente na biblioteca, o Flash também importa uma pasta de seus ativos, que contém as capas de seus diferentes estados. As *capas* de um componente compõem a coleção de símbolos que formam sua exibição gráfica no aplicativo. Uma capa é a representação gráfica, ou o clipe de filme, que indica um estado específico do componente.

O conteúdo da pasta de ativos de componente permite que você altere as capas do componente caso deseje fazer isso. Para obter mais informações, consulte [“Personalização dos componentes de UI”](#) na página 101.

Depois que um componente estiver na biblioteca, será possível adicionar mais ocorrências desse componente ao documento, arrastando seu ícone do painel Componentes ou do painel Biblioteca para o Palco.

Dimensionamento de componentes

Use a ferramenta Transformação livre ou o método `setSize()` para redimensionar ocorrências de componente. Você pode chamar o método `setSize()` em qualquer ocorrência de componente (consulte `UIComponent.setSize()`) para redimensioná-lo. O código a seguir redimensiona uma ocorrência do componente `List` para 200 pixels de largura por 300 pixels de altura:

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

Um componente não é redimensionado automaticamente para caber em seu rótulo. Se uma ocorrência de componente adicionada a um documento não for grande o suficiente para ser exibida em seu rótulo, o texto do rótulo será recortado. Redimensione o componente de modo que caiba em seu rótulo.

Para obter mais informações sobre o dimensionamento de componentes, consulte suas entradas individuais na [Referência do ActionScript 3.0 para Flash Professional](#).

Visualização ao vivo

O recurso Visualização ao vivo, ativado por padrão, permite que você exiba componentes no Palco à medida que eles aparecerem no conteúdo Flash publicado. Os componentes são exibidos em seu tamanho aproximado.

Para ativar ou desativar a Visualização ao vivo:

- Selecione Controlar > Ativar visualização ao vivo. Uma marca de seleção ao lado da opção indica que ela está ativada.

A visualização ao vivo reflete diferentes parâmetros para diferentes componentes. Para obter informações sobre quais parâmetros do componente se refletem na visualização ao vivo, consulte a entrada de cada componente na [Referência do ActionScript 3.0 para Flash Professional](#).



Um componente Button com o recurso Visualização ao vivo ativado



Um componente Button com o recurso Visualização ao vivo desativado

Os componentes na Visualização ao vivo não são funcionais. Para testar a funcionalidade, use o comando Controlar > Testar filme.

Manipulação de eventos

Cada componente transmite eventos quando um usuário interage com ele. Quando um usuário clica em um botão, por exemplo, ele despacha um evento `MouseEvent.CLICK` e, quando um usuário seleciona um item em uma lista, a lista despacha um evento `Event.CHANGE`. Um evento também pode ocorrer quando algo significativo acontece em um componente; por exemplo, quando um conteúdo conclui o carregamento de uma ocorrência de `UILoader`, gerando um evento `Event.COMPLETE`. Para manipular um evento, você grava o código do ActionScript que é executado quando o evento ocorre.

Os eventos de um componente incluem os eventos de qualquer classe da qual o componente os herda. Isso significa que todos os componentes de interface de usuário do ActionScript 3.0 herda eventos da classe `UIComponent` porque ela é a classe base desses componentes de UI. Para ver a lista dos eventos transmitidos por um componente, consulte a seção Eventos da entrada de classe do componente na [Referência do ActionScript 3.0 para Flash Professional](#).

Para obter uma explicação completa sobre a manipulação de eventos no ActionScript 3.0, consulte *Programação do ActionScript 3.0*.

Sobre os ouvintes de eventos

As pontas-chave a seguir se aplicam à manipulação de eventos dos componentes do ActionScript 3.0:

- Todos os eventos são transmitidos por uma ocorrência de uma classe de componente. A ocorrência de componente é o *transmissor*.

- Você registra um *ouvinte* de eventos chamando o método `addEventListener()` da ocorrência de componente. Por exemplo, a linha de código a seguir adiciona um ouvinte para o evento `MouseEvent.CLICK` à ocorrência de `Button` `aButton`:

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

O segundo parâmetro do método `addEventListener()` registra o nome da função, `clickHandler`, a ser chamado quando o evento ocorrer. Essa função também é conhecida como *função de retorno de chamada*.

- Você pode registrar vários ouvintes para uma única ocorrência de componente.

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

- Você pode registrar um ouvinte para várias ocorrências de componente.

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

- A função de manipulador de eventos é passada para um objeto de evento que contém informações sobre o tipo de evento e a ocorrência que transmite o evento. Para obter mais informações, consulte “[Sobre o objeto de evento](#)” na página 26.

- O ouvinte permanecerá ativo até que o aplicativo seja encerrado ou você o remova explicitamente usando o método `removeEventListener()`. Por exemplo, a linha de código a seguir remove o ouvinte do evento `MouseEvent.CLICK` em `aButton`:

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

Sobre o objeto de evento

O objeto de evento herda os eventos da classe de objeto `Event` e tem propriedades que contém informações sobre o evento ocorrido, incluindo as propriedades `target` e `type`, que fornecem informações essenciais sobre o evento:

| Propriedade | Descrição |
|---------------------|---|
| <code>type</code> | Uma string que indica o tipo do evento. |
| <code>target</code> | Uma referência à ocorrência de componente que está transmitindo o evento. |

Quando um evento tem propriedades adicionais, elas são listadas na descrição de classe do evento, na [Referência do ActionScript 3.0 para Flash Professional](#).

O objeto de evento é gerado e passado automaticamente para a função de manipulador de eventos quando um evento ocorre.

Você pode usar o objeto de evento dentro da função para acessar o nome do evento transmitido ou o nome da ocorrência do componente que transmitiu o evento. No nome de ocorrência, você pode acessar outras propriedades de componente. Por exemplo, o código a seguir usa a propriedade `target` do objeto de evento `evtObj` para acessar a propriedade `label` de `aButton` e exibi-la no painel Saída:

```
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");
}
```

Trabalho com a lista de exibição

Todos os componentes do ActionScript 3.0 herdam da classe `DisplayObject` e, portanto, têm acesso a seus métodos e propriedades para interagir com a lista de exibição. A *lista de exibição* é a hierarquia dos objetos e elementos visuais exibidos de um aplicativo. Essa hierarquia inclui os seguintes elementos:

- O Palco, que é o contêiner de nível superior
- Os objetos de exibição, que incluem formas, `MovieClip` e campos de texto, entre outros
- Os contêineres de objeto de exibição, que são tipos especiais de objetos de exibição que podem conter objetos de exibição-filho.

A ordem dos objetos na lista de exibição determina sua profundidade no contêiner-pai. A profundidade de um objeto se refere à sua posição de cima para baixo ou de frente para trás no Palco ou em seu contêiner de exibição. A ordem de profundidade fica aparente quando os objetos se sobrepõem, mas ela existe mesmo quando isso não acontece. Cada objeto da lista de exibição tem uma profundidade correspondente no Palco. Se você deseja alterar a profundidade de um objeto colocando-o na frente de outros objetos ou movendo-o para trás de outros objetos, é necessário alterar sua posição na lista de exibição. A ordem padrão dos objetos na lista de exibição é a ordem em que eles são posicionados no Palco. A posição 0 na lista de exibição é o objeto na parte inferior da ordem de profundidade.

Adicionar um componente à lista de exibição

Você pode adicionar um objeto a um objeto `DisplayObjectContainer` chamando o método `addChild()` ou `addChildAt()` do contêiner. No caso do Palco, você também pode adicionar um objeto à sua lista de exibição durante a criação ou, no caso dos componentes, arrastando-o do painel Componentes para o Palco. Para adicionar um objeto a um contêiner com o ActionScript, primeiro crie uma ocorrência dele chamando seu construtor com o novo operador e, em seguida, chame o método `addChild()` ou `addChildAt()` para colocá-lo no Palco e na lista de exibição. O método `addChild()` coloca o objeto na próxima posição na lista de exibição, enquanto `addChildAt()` especifica a posição em que o objeto será adicionado. Se você especificar uma posição que já está ocupada, o objeto que estiver nessa posição e os que estiverem nas posições superiores serão movidos 1 posição para cima. A propriedade `numChildren` de um objeto `DisplayObjectContainer` contém o número de objetos de exibição que ele contém. Você pode recuperar um objeto da lista de exibição chamando o método `getChildAt()` e especificando a posição ou, se você souber o nome do objeto, chamando o método `getChildByName()`.

Nota: Quando você adicionar um componente com o ActionScript, atribua um nome à sua propriedade `name`, caso deseje acessá-lo pelo nome na lista de exibição.

O exemplo a seguir exibe os nomes e as posições dos três componentes na lista de exibição. Primeiro, arraste um NumericStepper, um Button e um ComboBox para o Palco, a fim de que eles se sobreponham uns aos outros e atribua a eles os nomes de ocorrência **aNs**, **aButton** e **aCb**. Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo:

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

Você verá as seguintes linhas no painel Saída:

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

Movimentação de um componente na lista de exibição

Você pode alterar a posição e a profundidade de exibição de um objeto na lista de exibição, chamando o método `addChildAt()` e fornecendo o nome de um objeto e a posição em que você deseja colocá-lo como parâmetros do método. Por exemplo, adicione o código a seguir ao exemplo anterior para colocar o NumericStepper na parte superior e repita o loop para exibir as novas posições dos componentes na lista de exibição:

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

Você verá o seguinte no painel Saída:

```
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
```

O NumericStepper também aparecerá na frente dos outros componentes na tela.

Observe que `numChildren` representa o número de objetos (de 1 a n) na lista de exibição, enquanto a primeira posição da lista é 0. Portanto, se houver três objetos na lista, a posição de índice do terceiro objeto será 2. Isso significa que você pode fazer referência à última posição na lista de exibição, ou ao objeto superior em termos de profundidade de exibição, como `numChildren - 1`.

Remoção de um componente da lista de exibição

Você pode remover um componente de um contêiner de objeto de exibição e de sua lista de exibição com os métodos `removeChild()` e `removeChildAt()`. O exemplo a seguir coloca os três componentes Button na frente um do outro no Palco e adiciona um ouvinte de eventos para cada um deles. Quando você clica em cada botão, o manipulador de eventos o remove da lista de exibição e do Palco.

- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste um botão do painel Componentes para o painel Biblioteca.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e adicione o seguinte código:


```
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);
}
```

Para obter uma explicação completa da lista de exibição, consulte “Exibir programação” em *Programação do ActionScript 3.0*.

Trabalho com o FocusManager

Quando um usuário pressiona a tecla Tab para navegar em um aplicativo Flash ou clica em um aplicativo, a classe FocusManager determina qual componente receberá o foco de entrada. Você não precisa adicionar uma ocorrência de FocusManager a um aplicativo ou gravar qualquer código para ativar o FocusManager, a não ser que esteja criando um componente.

Se um objeto RadioButton receber o foco, o FocusManager examinará esse objeto e todos os objetos com o mesmo valor groupName, e definirá o foco no objeto com a propriedade selected definida como true.

Cada componente modal Window contém uma ocorrência do FocusManager, a fim de que os controles dessa janela se tornem seu próprio conjunto de tabulações. Isso impedirá que um usuário navegue inadvertidamente para os componentes em outras janelas por meio da tecla Tab.

O FocusManager usa o nível de profundidade (ou a ordem z) dos elementos no contêiner como esquema de navegação padrão ou *loop de tabulação*. Geralmente, um usuário navega no loop de tabulação usando a tecla Tab; o foco se movimenta do primeiro componente que tem o foco para o último e, depois, novamente para o primeiro. Os níveis de profundidade são configurados basicamente pela ordem em que os componentes são arrastados para o Palco. No entanto, você também pode usar os comandos Modificar > Organizar > Trazer para o Primeiro plano/Enviar para Plano de fundo para determinar a ordem z final. Para obter mais informações sobre os níveis de profundidade, consulte “[Trabalho com a lista de exibição](#)” na página 27.

Você pode chamar o método setFocus() para atribuir o foco a uma ocorrência de componente em um aplicativo. O exemplo a seguir cria uma ocorrência do FocusManager para o contêiner atual (this) e atribui o foco à ocorrência de Button aButton.

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

Você pode determinar qual componente tem o foco chamando o método `getFocus()` e qual componente do loop de tabulação receberá o foco em seguida, chamando o método `getNextFocusManagerComponent()`. No exemplo a seguir, uma `CheckBox`, um `RadioButton` e um `Button` estão no Palco e cada componente tem ouvintes para os eventos `MouseEvent.CLICK` e `FocusEvent.MOUSE_FOCUS_CHANGE`. Quando o evento `MouseEvent.CLICK` ocorre, porque o usuário clicou no componente, a função `showFocus()` chama o método `getNextFocusManagerComponent()` para determinar qual componente do loop de tabulação receberá o foco em seguida. Em seguida, ela chama o método `setFocus()` para atribuir o foco a esse componente. Quando o evento `FocusEvent.MOUSE_FOCUS_CHANGE` ocorre, a função `fc()` exibe o nome do componente em que o ocorreu o evento. Esse evento é acionado quando o usuário clica em um componente que não seja o próximo no loop de tabulação.

```
// 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);
}
```

Para criar um botão que recebe o foco quando um usuário pressiona Enter (Windows) ou Return (Macintosh), defina a propriedade `FocusManager.defaultButton` com a ocorrência de `Button` que será o botão padrão, conforme no código a seguir:

```
import fl.managers.FocusManager;

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

A classe `FocusManager` substitui o retângulo de foco padrão do Flash Player e desenha um retângulo de foco personalizado com cantos arredondados.

Para obter mais informações sobre a criação de um esquema de foco em um aplicativo Flash, consulte a `FocusManager` class na [Referência do ActionScript 3.0 para Adobe Flash Platform](#). Para criar um gerenciador de foco personalizado, crie uma classe que implemente a interface do `IFocusManager`. Para obter mais informações, consulte `IFocusManager` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

Trabalho com componentes baseados em lista

Os componentes List, DataGrid e TileList herdam da classe base SelectableList. Por esse motivo, esses componentes são considerados componentes baseados em lista. Uma ComboBox consiste em uma caixa de texto e uma lista; por isso, ela também é um componente baseado em lista.

Uma lista é composta de linhas. Uma DataGrid e uma TileList são compostas por linhas que podem ser divididas em várias colunas. A interseção de uma linha e uma coluna é uma célula. Em uma lista, que é uma única coluna de linhas, cada linha é uma célula. Uma célula tem dois aspectos importantes:

- Os valores de dados que a célula retém chamam-se itens. Um *item* é um objeto do ActionScript usado para armazenar as unidades de informações em uma lista. Uma lista pode ser considerada uma matriz em que cada espaço indexado é um item. Em uma lista, um item é um objeto que geralmente tem uma propriedade `label` exibida e uma propriedade `data` usada para armazenar dados. Um *provedor de dados* é um modelo de dados dos itens em uma lista. Um provedor de dados permite que você preencha um componente baseado em lista simplesmente atribuindo-o à propriedade `dataProvider` do componente.
- Uma célula pode reter diferentes tipos de dados que variam de texto a imagens, MovieClip ou qualquer classe criada. Por esse motivo, uma célula deve ser desenhada ou renderizada de modo apropriado para seu conteúdo. Conseqüentemente, os componentes baseados em lista têm um *renderizador de célula* para renderizar suas células. No caso da DataGrid, cada coluna é um objeto `DataGridColumn`, que também tem uma propriedade `cellRenderer`, a fim de que cada coluna possa ser renderizada apropriadamente para seu conteúdo.

Todos os componentes baseados em lista têm as propriedades `cellRenderer` e `dataProvider` que você pode definir para carregar e renderizar as células desses componentes. Para obter informações sobre o uso dessas propriedades e o trabalho com componentes baseados em lista, consulte [“Trabalho com um DataProvider”](#) na página 31 e [“Trabalho com um CellRenderer”](#) na página 39.

Trabalho com um DataProvider

Um DataProvider é uma origem de dados que você pode usar para fornecer dados aos componentes ComboBox, DataGrid, List e TileList. Cada uma dessas classes de componente tem uma propriedade `dataProvider` à qual você pode atribuir um objeto DataProvider para preencher as células do componente com dados. Geralmente, um provedor de dados é uma coleção de dados, como um objeto Array ou XML.

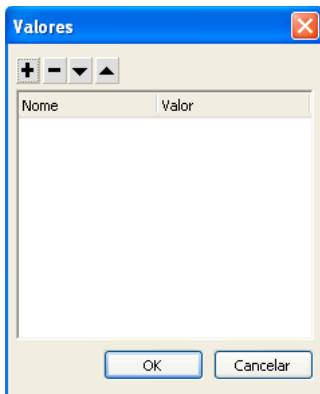
Criar um DataProvider

Para os componentes ComboBox, List e TileList, você pode criar um DataProvider usando o parâmetro `dataProvider` no ambiente de criação. O componente DataGrid não tem um parâmetro `dataProvider` no Inspetor de propriedades porque ele tem várias colunas e, conseqüentemente, seu provedor de dados é mais complexo. Você também pode usar o ActionScript para criar um DataProvider para esses componentes, bem como para a DataGrid.

Uso do parâmetro dataProvider

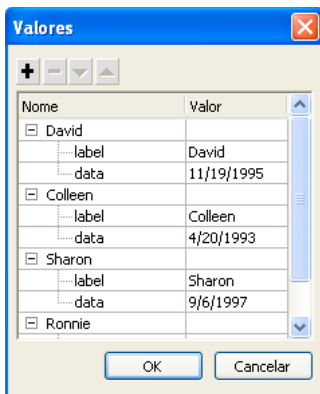
Você pode criar um provedor de dados simples para os componentes ComboBox, List e TileList clicando no parâmetro `dataProvider` na aba Parâmetros do Inspetor de propriedades ou do Inspetor de componentes.

Se você clicar duas vezes na célula de valor, que inicialmente mostra uma matriz vazia, a caixa de diálogo Valores será aberta e você poderá inserir diversos valores de rótulo e dados para criar o provedor de dados.



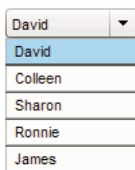
Caixa de diálogo Valores de dataProvider

Clique no sinal de adição para adicionar um item ao parâmetro dataProvider. Clique no sinal de subtração para excluir um item. Clique na seta para cima a fim de mover um item selecionado para cima na lista ou clique na seta para baixo a fim de mover um item selecionado para baixo na lista. A ilustração a seguir mostra a caixa de diálogo Valores que cria uma lista de nomes de crianças e suas datas de aniversário.



Caixa de diálogo Valores com dados

A matriz criada consiste em pares de campos de rótulo e valor. Os campos de rótulo são `label` e `data`, e os campos de valor são os nomes das crianças e suas datas de aniversário. O campo de rótulo identifica o conteúdo que aparece na lista, que, neste caso, é representado pelos nomes das crianças. A ComboBox resultante terá esta aparência:



A ComboBox preenchida pelo DataProvider

Quando você terminar de adicionar os dados, clique em OK para fechar a caixa de diálogo. Agora, a matriz no parâmetro dataProvider é preenchida com os itens criados.

Trabalho com componentes

| Nome | Valor |
|-------------------|---|
| allowMultipleS... | False |
| dataProvider | [{label:David,data:11/19/1995},{label:Colleen,data:4/20/1993},{label:Sharon,data:9/6/1997},...] |
| enabled | true |
| horizontalLine... | 4 |
| horizontalPag... | 0 |
| horizontalScro... | auto |

parâmetro `dataProvider` com dados

Você pode acessar os valores de rótulo e dados criados usando o ActionScript para acessar a propriedade `dataProvider` do componente.

Criar um DataProvider por meio do ActionScript

Você pode criar um `DataProvider` criando os dados em um objeto `Array` ou `XML` e fornecendo o objeto como parâmetro `value` para o construtor `DataProvider`.

Nota: No ActionScript 3.0, você não pode atribuir um objeto `Array` ou `XML` diretamente a uma propriedade `dataProvider` porque a propriedade é definida como um objeto `DataProvider` e só pode receber um objeto do tipo `DataProvider`.

O exemplo a seguir preenche um componente `List`, que é uma coluna de linhas com os nomes de várias crianças e suas respectivas datas de aniversário. O exemplo define a lista na matriz `items` e a define como parâmetro quando cria a ocorrência de `DataProvider` (`new DataProvider(items)`) e a atribui à propriedade `dataProvider` do componente `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);
```

A matriz consiste em pares de campos de rótulo e valor. Os campos de rótulo são `label` e `data`, e os campos de valor são os nomes das crianças e suas datas de aniversário. O campo de rótulo identifica o conteúdo que aparece na lista, que, neste caso, é representado pelos nomes das crianças. A lista resultante terá esta aparência:

| |
|---------|
| David |
| Colleen |
| Sharon |
| Ronnie |
| James |

Uma lista preenchida por um `DataProvider`

O valor do campo de dados estará disponível quando o usuário selecionar um item na lista clicando nele e provocando um evento `change`. O exemplo a seguir adiciona uma `TextArea` (`aTa`) e um manipulador de eventos (`changeHandler`) ao exemplo anterior para exibir a data de aniversário da criança quando um usuário selecionar um nome na lista.

```
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);

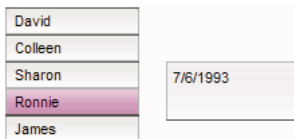
addChild(aList);
addChild(aTa);

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

aList.addEventListener(Event.CHANGE, changeHandler);

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

Agora, quando um usuário selecionar o nome de uma criança na lista, a data de aniversário da criança será exibida na TextArea, conforme mostra a ilustração a seguir. Isso é feito pela função `changeHandler()` quando ela define a propriedade `text` da TextArea (`aTa.text`) para o valor do campo de dados no item selecionado (`event.target.selectedItem.data`). A propriedade `event.target` é o objeto que acionou o evento, que, neste caso, é a lista.



Exibição do campo de dados no DataProvider de uma lista

Você pode incluir dados que não sejam texto em um DataProvider. O exemplo a seguir inclui MovieClips em um DataProvider que fornece dados para uma TileList. Ele cria o DataProvider chamando o método `addItem()` para adicionar cada item depois que ele cria o MovieClip, uma caixa colorida.

```
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();
}
```

Você também pode usar os dados XML (em vez de uma matriz) para preencher um objeto DataProvider. Por exemplo, o código a seguir armazena dados em um objeto XML chamado `employeesXML` e passa esse objeto como parâmetro de valor da função do construtor `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;
```

Você pode fornecer dados como atributos dos dados XML, como no código anterior, ou como propriedades dos dados XML, como no código a seguir:

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

O DataProvider também tem um conjunto de métodos e propriedades que permitem a você acessá-los e manipulá-los. Você pode usar a API do DataProvider para adicionar, remover, substituir, classificar e mesclar itens em um DataProvider.

Manipular um DataProvider

Você pode adicionar itens a um DataProvider com os métodos `addItem()` e `addItemAt()`. O exemplo a seguir adiciona itens que um usuário digita no campo de texto de uma ComboBox editável. Ele assume que uma ComboBox foi arrastada para o Palco e recebeu o nome de ocorrência `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);
}
```

Você também pode substituir e remover itens em um componente por meio de seu DataProvider. O exemplo a seguir implementa dois componentes List separados, `listA` e `listB`, e fornece um Button com o rótulo Sincronizar. Quando um usuário clica em Button, o exemplo usa o método `replaceItemAt()` para substituir os itens de `listB` pelos itens de `listA`. Se `listA` for maior que `listB`, o exemplo chamará o método `addItem()` para adicionar os itens extras a `listB`. Se `listB` for maior que `listA`, o exemplo chamará o método `removeItemAt()` para remover os itens extras de `listB`.


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

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

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

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

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

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

syncButton.addEventListener(MouseEvent.CLICK, syncHandler);

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

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

Você também pode mesclar e classificar um DataProvider usando os métodos `merge()`, `sort()` e `sortOn()`. O exemplo a seguir preenche duas ocorrências de DataGrid (`aDg` e `bDg`) com listagens parciais de dois times de beisebol. Ele adiciona um Button com o rótulo Mesclar e, quando o usuário clica nele, o manipulador de eventos (`mrgHandler`) mescla a lista de `bDg` com a lista de `aDg` e classifica a DataGrid resultante na coluna Nome.

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

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

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

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

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

Para obter mais informações, consulte a classe `DataProvider` na [Referência do ActionScript 3.0](#).

Trabalho com um CellRenderer

`CellRenderer` é uma classe que os componentes baseados em lista, como `List`, `DataGrid`, `TileList` e `ComboBox`, usam para manipular e exibir o conteúdo de célula personalizado de suas linhas. Uma célula personalizada pode conter texto, um componente predefinido, como uma `CheckBox`, ou qualquer classe de objeto de exibição criada. Para renderizar os dados usando o `CellRenderer` personalizado, você pode estender a classe `CellRenderer` ou implementar a interface do `ICellRenderer` para criar sua classe `CellRenderer` personalizada.

As classes `List`, `DataGrid`, `TileList` e `ComboBox` são subclasses da classe `SelectableList`. A classe `SelectableList` inclui o estilo `cellRenderer`. Esse estilo define o objeto de exibição que o componente usa para renderizar células.

Você pode ajustar a formatação dos estilos usados por `CellRenderer` chamando o método `setRendererStyle()` do objeto `List` (consulte [“Formatação de células”](#) na página 39). Uma outra alternativa é definir uma classe personalizada a ser usada como `CellRenderer` (consulte [“Definição de uma classe CellRenderer personalizada”](#) na página 40).

Formatação de células

A classe `CellRenderer` inclui vários estilos que permitem controlar o formato da célula.

Os estilos a seguir permitem que você defina as capas usadas nos diferentes estados da célula (`disabled`, `down`, `over` e `up`):

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

Os estilos a seguir se aplicam à formatação de texto:

- `disabledTextFormat`
- `textFormat`

- `textPadding`

Você pode definir esses estilos chamando o método `setRendererStyle()` do objeto `List` ou chamando o método `setStyle()` do objeto `CellRenderer`. Você pode obter esses estilos chamando o método `getRendererStyle()` do objeto `List` ou chamando o método `getStyle()` do objeto `CellRenderer`. Você também pode acessar um objeto que defina todos os estilos de renderizador (como propriedades nomeadas do objeto) por meio da propriedade `rendererStyles` do objeto `List` ou do método `getStyleDefinition()` do objeto `CellRenderer`.

É possível chamar o método `clearRendererStyle()` para redefinir um estilo para seu valor padrão.

Para obter ou definir a altura das linhas na lista, use a propriedade `rowHeight` do objeto `List`.

Definição de uma classe `CellRenderer` personalizada

Criar uma classe que estende a classe `CellRenderer` para definir um `CellRenderer` personalizado

Por exemplo, o código a seguir inclui duas classes. A classe `ListSample` instancia um componente `List` e usa a outra classe, `CustomRenderer`, para definir o renderizador de célula a ser usado no componente `List`. A classe `CustomRenderer` estende a classe `CellRenderer`.

- 1 Selecione Arquivo > Novo.
- 2 Na caixa de diálogo Novo documento exibida, selecione Arquivo Flash (ActionScript 3.0) e clique em OK.
- 3 Selecione Janela > Componentes para exibir o painel Componentes.
- 4 No painel Componentes, arraste um componente `List` para o Palco.
- 5 Se o Flash não estiver exibindo o Inspetor de propriedades, selecione Janela > Propriedades > Propriedades.
- 6 Com o componente `List` selecionado, defina as propriedades no Inspetor de propriedades:
 - Nome da ocorrência: `myList`
 - L (largura): 200
 - A (altura): 300
 - X: 20
 - Y: 20
- 7 Selecione Quadro 1 da Camada 1 na Linha do tempo e, em seguida, selecione Janela > Ações.
- 8 Digite o script a seguir no painel Ações:

```
myList.setStyle("cellRenderer", CustomCellRenderer);
myList.addItem({label:"Burger -- $5.95"});
myList.addItem({label:"Fries -- $1.95"});
```
- 9 Selecione Arquivo > Salvar. Atribua um nome ao arquivo e clique no botão OK.
- 10 Selecione Arquivo > Novo.
- 11 Na caixa de diálogo Novo documento exibida, selecione Arquivo ActionScript e clique no botão OK.
- 12 Na janela de script, digite o código a seguir para definir a classe `CustomCellRenderer`:

```

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

```

13 Selecione Arquivo > Salvar. Nomeie o arquivo como CustomCellRenderer.as, coloque-o no mesmo diretório do arquivo FLA e clique no botão OK.

14 Selecione Controlar > Testar filme.

Uso de uma classe que implemente a interface do ICellRenderer para definir um CellRenderer personalizado

Você também pode definir um CellRenderer usando qualquer classe que herde a classe DisplayObject e implemente a interface do ICellRenderer. Por exemplo, o código a seguir define duas classes. A classe ListSample2 adiciona um objeto List à lista de exibição e define seu CellRenderer para usar a classe CustomRenderer. A classe CustomRenderer estende a classe CheckBox (que, por sua vez, estende a classe DisplayObject) e implementa a interface do ICellRenderer. Observe que a classe CustomRenderer define os métodos getter e setter para as propriedades data e listData, definidas na interface do ICellRenderer. Outras propriedades e métodos definidos na interface do ICellRenderer (a propriedade selected e o método setSize()) já são definidas na classe CheckBox:

- 1 Selecione Arquivo > Novo.
- 2 Na caixa de diálogo Novo documento exibida, selecione Arquivo Flash (ActionScript 3.0) e clique em OK.
- 3 Selecione Janela > Componentes para exibir o painel Componentes.
- 4 No painel Componentes, arraste um componente List para o Palco.
- 5 Se o Flash não estiver exibindo o Inspetor de propriedades, selecione Janela > Propriedades > Propriedades.
- 6 Com o componente List selecionado, defina as propriedades no Inspetor de propriedades:

- Nome da ocorrência: myList
- L (largura): 100
- A (altura): 300
- X: 20
- Y: 20

7 Selecione Quadro 1 da Camada 1 na Linha do tempo e, em seguida, selecione Janela > Ações.

8 Digite o script a seguir no painel Ações:

```

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

```

9 Selecione Arquivo > Salvar. Atribua um nome ao arquivo e clique no botão OK.

10 Selecione Arquivo > Novo.

11 Na caixa de diálogo Novo documento exibida, selecione Arquivo ActionScript e clique no botão OK.

12 Na janela de script, digite o código a seguir para definir a classe 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 Selecione Arquivo > Salvar. Nomeie o arquivo como CustomCellRenderer.as, coloque-o no mesmo diretório do arquivo FLA e clique no botão OK.

14 Selecione Controlar > Testar filme.

Uso de um símbolo para definir um CellRenderer

Você também pode usar um símbolo da biblioteca para definir um CellRenderer. O símbolo do ActionScript deve ser exportado, ao passo que o nome de classe do símbolo de biblioteca deve ter um arquivo de classe associado que implemente a interface do ICellRenderer ou estenda a classe CellRenderer (ou uma de suas subclasses).

O exemplo a seguir define um CellRenderer personalizado usando um símbolo de biblioteca.

- 1 Selecione Arquivo > Novo.
- 2 Na caixa de diálogo Novo documento exibida, selecione Arquivo Flash (ActionScript 3.0) e clique em OK.
- 3 Selecione Janela > Componentes para exibir o painel Componentes.
- 4 No painel Componentes, arraste um componente List para o Palco.
- 5 Se o Flash não estiver exibindo o Inspetor de propriedades, selecione Janela > Propriedades > Propriedades.
- 6 Com o componente List selecionado, defina as propriedades no Inspetor de propriedades:
 - Nome da ocorrência: myList
 - L (largura): 100
 - A (altura): 400
 - X: 20
 - Y: 20

- 7 Clique no painel Parâmetros e clique duas vezes na segunda coluna da linha dataProvider.
- 8 Na caixa de diálogo Valores exibida, clique no sinal de adição duas vezes para adicionar dois elementos de dados (com rótulos definidos para label0 e label1) e clique no botão OK.
- 9 Com a ferramenta Texto, desenhe um campo de texto no Palco.
- 10 Com o campo de texto selecionado, defina as propriedades no Inspetor de propriedades:
 - Tipo de texto: Texto dinâmico
 - Nome da ocorrência: textField
 - L (largura): 100
 - Tamanho da fonte: 24
 - X: 0
 - Y: 0
- 11 Com o campo de texto selecionado, selecione Modificar > Converter em símbolo.
- 12 Na caixa de diálogo Converter em símbolo, defina as seguintes configurações e clique em OK.
 - Nome: MyCellRenderer
 - Tipo: MovieClip
 - Exportar para ActionScript: Selecionado
 - Exportar no primeiro quadro: Selecionado
 - Classe: MyCellRenderer
 - Classe base: flash.display.MovieClipSe o Flash exibir um aviso de classe do ActionScript, clique no botão OK na caixa de aviso.
- 13 Exclua a ocorrência do novo símbolo de clipe de filme do Palco.
- 14 Selecione Quadro 1 da Camada 1 na Linha do tempo e, em seguida, selecione Janela > Ações.
- 15 Digite o script a seguir no painel Ações:

```
myList.setStyle("cellRenderer", MyCellRenderer);
```
- 16 Selecione Arquivo > Salvar. Atribua um nome ao arquivo e clique no botão OK.
- 17 Selecione Arquivo > Novo.
- 18 Na caixa de diálogo Novo documento exibida, selecione Arquivo ActionScript e clique no botão OK.
- 19 Na janela de script, digite o código a seguir para definir a classe 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 Selecione Arquivo > Salvar. Nomeie o arquivo como MyCellRenderer.as, coloque-o no mesmo diretório do arquivo FLA e clique no botão OK.

21 Selecione Controlar > Testar filme.

propriedades CellRenderer

A propriedade `data` é um objeto que contém todas as propriedades definidas para o `CellRenderer`. Por exemplo, na classe a seguir, que define um `CellRenderer` personalizado que estende a classe `CheckBox`, observe que a função `setter` da propriedade `data` passa o valor `data.label` para a propriedade `label` que herda da classe `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;
    }
}
}
```

A propriedade `selected` define se uma célula será ou não selecionada na lista.

Aplicar um CellRenderer a uma coluna de um objeto DataGrid

Um objeto `DataGrid` pode ter várias colunas, e você pode especificar diferentes renderizadores de célula para cada coluna. Cada coluna de uma `DataGrid` é representada por um objeto `DataGridColumn`, e a classe `DataGridColumn` inclui uma propriedade `cellRenderer`, para a qual você pode definir o `CellRenderer` da coluna.

Definição de um CellRenderer para um célula editável

A classe `DataGridCellEditor` define um renderizador usado para células editáveis em um objeto `DataGrid`. Ela se torna o renderizador de uma célula quando a propriedade `editable` do objeto `DataGrid` é definido como `true` e o usuário clica na célula a ser editada. Para definir um `CellRenderer` para a célula editável, defina a propriedade `itemEditor` de cada elemento da matriz `columns` do objeto `DataGrid`.

Uso de uma imagem, arquivo SWF ou clipe de filme como um CellRenderer

A classe `ImageCell`, uma subclasse de `CellRenderer`, define um objeto usado para renderizar células em que o conteúdo principal da célula é uma imagem, arquivo SWF ou clipe de filme. A classe `ImageCell` inclui os seguintes estilos para definir a aparência da célula:

- `imagePadding` — O preenchimento em pixels que separa a borda da célula da borda da imagem
- `selectedSkin` — A capa usada para indicar o estado selecionado
- `textOverlayAlpha` — A opacidade da sobreposição por trás do rótulo da célula
- `textPadding` — O preenchimento em pixels que separa a borda da célula da borda do texto

A classe `ImageCell` é o `CellRenderer` padrão da classe `TileList`.

Tornar os componentes acessíveis

Você pode tornar o conteúdo visual nos aplicativos Flash acessíveis a usuários com deficiências visuais por meio de um leitor de tela, que fornece uma descrição em áudio do conteúdo da tela. Para obter informações sobre como tornar o aplicativo Flash acessível a um leitor de tela, consulte o Capítulo 18, “Criação de conteúdo acessível”, em *Uso do Flash*.

Para tornar um componente do ActionScript 3.0 acessível a um leitor de tela, você deve importar também sua classe de acessibilidade e chamar o método `enableAccessibility()` dessa classe. É possível tornar os seguintes componentes do ActionScript 3.0 acessíveis a um leitor de tela:

| Componente | Classe de acessibilidade |
|-------------|--------------------------|
| Button | ButtonAccImpl |
| CheckBox | CheckBoxAccImpl |
| ComboBox | ComboBoxAccImpl |
| List | ListAccImpl |
| RadioButton | RadioButtonAccImpl |
| TileList | TileListAccImpl |

As classes de acessibilidade do componente estão no pacote `fl.accessibility`. Para tornar uma `CheckBox` acessível a um leitor de tela, por exemplo, adicione as seguintes instruções ao aplicativo:

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

A acessibilidade de um componente é ativada somente uma vez, independentemente de quantas ocorrências são criadas.

Nota: A ativação da acessibilidade aumenta o tamanho do arquivo, já que as classes necessárias são incluídas durante a compilação.

A maioria dos componentes também podem ser acessados por meio do teclado. Para obter mais informações sobre a ativação dos componentes de acessibilidade e a navegação por meio do teclado, consulte as seções *Interação do usuário* de “[Uso dos componentes de UI](#)” na página 47 e as classes de acessibilidade na [Referência do ActionScript 3.0 para Flash Professional](#).

Capítulo 4: Uso dos componentes de UI

Este capítulo explica como usar os componentes de interface de usuário do ActionScript 3.0 incluídos no Flash.

Uso do componente Button

O componente Button é um botão retangular redimensionável que o usuário pode pressionar com o mouse ou a barra de espaço para iniciar uma ação no aplicativo. Você pode adicionar um ícone personalizado a um botão. Também é possível alterar o comportamento de um botão de modo que ele deixe de ser um botão de ação para se tornar um botão de alternância. O botão de alternância permanece pressionado quando é clicado e retorna ao seu estado ativo quando é clicado novamente.

O botão é a parte fundamental de vários formulários e aplicativos da Web. Você poderá usar os botões sempre que um usuário tiver que iniciar um evento. Por exemplo, a maioria dos formulários têm um botão Enviar. Você também poderá adicionar os botões Anterior e Próximo a uma apresentação.

Interação do usuário com o componente Button

É possível ativar ou desativar um botão em um aplicativo. No estado desativado, um botão não recebe entradas de mouse nem de teclado. Um botão ativado receberá o foco se você clicar nele ou acessá-lo por meio da tecla tab. Quando uma ocorrência de Button estiver com o foco, você poderá usar as seguintes teclas para controlá-la:

| Tecla | Descrição |
|-----------------|--|
| Shift+Tab | Move o foco para o objeto anterior. |
| Barra de espaço | Pressiona ou solta o botão e aciona o evento <code>click</code> . |
| Tabulação | Move o foco para o próximo objeto. |
| Enter/Return | Move o foco para o objeto seguinte se um botão for definido como o botão padrão do FocusManager. |

Para obter mais informações sobre o controle do foco, consulte a interface IFocusManager e a classe FocusManager na [Referência do ActionScript 3.0 para Flash Professional](#) e “Trabalho com o FocusManager” na página 29.

Uma visualização ao vivo de cada ocorrência de Button reflete as alterações feitas nos parâmetros no Inspetor de propriedades ou no Inspetor de componentes durante a criação.

Nota: Se um ícone for maior do que o botão, ele ultrapassará as bordas do botão.

Para designar um botão como botão de ação padrão em um aplicativo (o botão que recebe o evento click quando um usuário pressiona Enter), defina `FocusManager.defaultButton`. Por exemplo, o código a seguir define o botão padrão para ser uma ocorrência de Button chamada `submitButton`.

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

Quando você adicionar o componente Button a um aplicativo, poderá torná-lo acessível a um leitor de tela adicionando as seguintes linhas do código do ActionScript:

```
import fl.accessibility.ButtonAccImpl;

ButtonAccImpl.enableAccessibility();
```

A acessibilidade de um componente é ativada somente uma vez, independentemente de quantas ocorrências são criadas.

Parâmetros do componente Button

Você pode definir os seguintes padrões de autoria no Inspetor de propriedades (Janela > Propriedades > Propriedades) ou no Inspetor de componentes (Janela > Inspetor de componentes) de cada instância de Button: *emphasized*, *label*, *labelPlacement*, *selected* e *toggle*. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Ao atribuir um valor a esses parâmetros, você está definindo o estado inicial da propriedade no aplicativo. A definição da propriedade no ActionScript substitui o valor definido no parâmetro. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe Button na [Referência do ActionScript 3.0 para Flash Professional](#).

Criar um aplicativo com o componente Button

O procedimento a seguir explica como adicionar um componente Button a um aplicativo durante a criação. Neste exemplo, Button altera o estado de um componente ColorPicker quando você clica nele.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente Button do painel Componentes para o Palco e digite os seguintes valores no Inspetor de propriedades:
 - Digite o nome de ocorrência **aButton**.
 - Digite **Show** como parâmetro label.
- 3 Adicione um ColorPicker ao Palco e atribua a ele o nome de ocorrência **aCp**.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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;
    }
}
```

A segunda linha do código registra a função `clickHandler()` como função de manipulador do evento `MouseEvent.CLICK`. O evento ocorre quando um usuário clica no botão, fazendo com que a função `clickHandler()` adote uma das seguintes ações, dependendo do valor do botão:

- Mostrar faz com que o `ColorPicker` fique visível e altera o rótulo do botão para Desativar.
- Desativar desativa o `ColorPicker` e altera o rótulo do botão para Ativar.
- Ativar ativa o `ColorPicker` e altera o rótulo do botão para Ocultar.
- Ocultar faz com que o `ColorPicker` fique invisível e altera o rótulo do botão para Mostrar.

5 Selecione Controlar > Testar filme para executar o aplicativo.

Criar um aplicativo com o componente Button

O procedimento a seguir cria um botão de alternância por meio do `ActionScript` e exibe o tipo de evento no painel Saída quando você clica no botão. O exemplo cria a ocorrência de `Button` chamando o construtor da classe e a adiciona ao Palco chamando o método `addChild`.

1 Crie um novo documento Flash (`ActionScript 3.0`).

2 Arraste o componente `Button` do painel Componentes para o painel Biblioteca do documento atual.

Esse procedimento adicionará o componente à biblioteca, mas não o tornará visível no aplicativo.

3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código para criar uma ocorrência de `Button`:

```
import fl.controls.Button;

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

O método `move()` posiciona o botão no local 50 (coordenada x), 50 (coordenada y) no Palco.

4 Agora, adicione o seguinte `ActionScript` para criar um ouvinte de evento e uma função de manipulador de evento:

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

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

5 Selecione Controlar > Testar filme.

Quando você clica no botão, o Flash exibe a mensagem “Tipo de evento: click” no painel Saída.

Uso do componente CheckBox

A `CheckBox` é uma caixa quadrada que pode ser marcada ou desmarcada. Quando ele é marcado, uma caixa de seleção aparece na caixa. Você pode adicionar um rótulo de texto a uma `CheckBox` e colocá-lo à esquerda, à direita, acima ou abaixo da `CheckBox`.

Você pode usar as `CheckBoxes` para reunir um conjunto de valores `verdadeiro` ou `falso` que não sejam mutuamente exclusivos. Por exemplo, um aplicativo que reúne informações sobre o tipo de carro a ser comprado pode usar `CheckBoxes` para permitir que você selecione recursos.

Interação do usuário com a CheckBox

É possível ativar ou desativar uma CheckBox em um aplicativo. Se a CheckBox estiver ativada e o usuário clicar nela ou em seu rótulo, ela receberá o foco de entrada e será exibida com a aparência pressionada. Se um usuário mover o ponteiro para fora da área delimitadora de uma CheckBox ou de seu rótulo enquanto pressiona o botão do mouse, a aparência do componente retornará ao seu estado original e reterá o foco de entrada. O estado de uma CheckBox muda apenas depois que o mouse é solto sobre o componente. Além disso, a CheckBox tem dois estados desativados, marcado e desmarcado, que usam `selectedDisabledSkin` e `disabledSkin`, respectivamente, que não permitem a interação do mouse ou do teclado.

Se uma CheckBox for desativada, ela será exibida como desativada, independentemente da interação do usuário. No estado desativado, uma CheckBox não recebe a entrada do mouse ou do teclado.

Uma ocorrência de CheckBox receberá o foco se um usuário clicar nela ou acessá-la por meio da tecla tab. Quando uma ocorrência de CheckBox estiver com o foco, você poderá usar as seguintes teclas para controlá-la:

| Tecla | Descrição |
|-----------------|--|
| Shift+Tab | Move o foco para o elemento anterior. |
| Barra de espaço | Marca ou desmarca o componente e aciona o evento <code>change</code> . |
| Tabulação | Move o foco para o elemento seguinte. |

Para obter mais informações sobre o controle do foco, consulte “Trabalho com o FocusManager” na página 29 e a classe FocusManager na [Referência do ActionScript 3.0 para Flash Professional](#).

Uma visualização ao vivo de cada ocorrência de CheckBox reflete as alterações feitas nos parâmetros no Inspetor de propriedades ou no Inspetor de componentes durante a criação.

Quando você adicionar o componente CheckBox a um aplicativo, poderá torná-lo acessível a um leitor de tela adicionando as seguintes linhas do código do ActionScript:

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

A acessibilidade de um componente é ativada somente uma vez, independentemente de quantas ocorrências do componente existem.

Parâmetros do componente CheckBox

Você pode definir os seguintes parâmetros de autoria no Inspetor de propriedades ou no Inspetor de componentes para cada instância do componente CheckBox: `label`, `labelPlacement` e `selected`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe CheckBox na [Referência do ActionScript 3.0 para Flash Professional](#).

Criar um aplicativo por meio da CheckBox

O procedimento a seguir explica como adicionar um componente CheckBox a um aplicativo durante a criação, usando um trecho de um formulário de aplicativo de empréstimo. O formulário pergunta se o candidato é um proprietário de imóvel e fornece uma CheckBox para que o usuário responda “sim”. Em caso afirmativo, o formulário apresentará dois botões de opção para o usuário indicar o valor relativo do imóvel.

Criar um aplicativo por meio do componente Checkbox

- 1 Crie um novo documento Flash (ActionScript 3.0).

- 2 Arraste um componente CheckBox do painel Componentes para o Palco.
- 3 No Inspetor de propriedades, faça o seguinte:
 - Digite **homeCh** como nome de ocorrência.
 - Digite **140** como valor da largura (L).
 - Digite “**Quer ter uma casa própria?**” como parâmetro label.
- 4 Arraste dois componentes RadioButton do painel Componentes para o Palco e coloque-os abaixo e à direita da CheckBox. Digite os valores a seguir para eles no Inspetor de propriedades:
 - Digite **underRb** e **overRb** como nomes de ocorrência.
 - Digite **120** como parâmetro de largura (L) dos dois RadioButtons.
 - Digite **Abaixo de US\$ 500.000?** como parâmetro label **underRb**.
 - Digite **Acima de US\$ 500.000?** como parâmetro label **overRb**.
 - Digite **valueGrp** como parâmetro groupName dos dois RadioButtons.
- 5 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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;
}
```

Esse código cria um manipulador para o evento `CLICK` que ativará os RadioButtons `underRb` e `overRb` se a CheckBox `homeCh` for marcada e os desativará se `homeCh` não for marcada. Para obter mais informações, consulte a classe `MouseEvent` na [Referência do ActionScript 3.0 para Flash Professional](#).

- 6 Selecione Controlar > Testar filme.

O exemplo a seguir duplica o aplicativo anterior, mas cria a CheckBox e os RadioButtons com o ActionScript.

Criar uma CheckBox por meio do ActionScript

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente CheckBox e o componente RadioButton do painel Componentes para o painel Biblioteca do documento atual. Se o painel Biblioteca não estiver aberto, pressione `Ctrl+L` ou selecione Janela > Biblioteca para abri-lo.

Isso disponibiliza os componentes para os seus aplicativos, mas não os coloca no Palco.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código para criar e posicionar ocorrências de componentes:

```
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?";
```

Esse código usa os construtores `CheckBox()` e `RadioButton()` para criar os componentes e o método `addChild()` para colocá-los no Palco. Ele usa o método `move()` para posicionar os componentes no Palco.

- 4 Agora, adicione o seguinte ActionScript para criar um ouvinte de evento e uma função de manipulador de evento:

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

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

Esse código cria um manipulador para o evento `CLICK` que ativará os botões de opção `underRb` e `overRb` se a `CheckBox` `homeCh` for marcada e os desativará se `homeCh` não for marcada. Para obter mais informações, consulte a classe `MouseEvent` na [Referência do ActionScript 3.0 para Flash Professional](#).

- 5 Selecione Controlar > Testar filme.

Uso do componente ColorPicker

O componente `ColorPicker` permite que o usuário selecione uma cor em uma lista de amostras. O modo padrão do `ColorPicker` mostra uma única cor em um botão quadrado. Quando um usuário clica no botão, a lista de cores disponíveis aparece em um painel de amostras juntamente com um campo de texto que exibe o valor hexadecimal da seleção de cor atual.

Você pode definir as cores que aparecem no `ColorPicker` definindo sua propriedade `colors` com os valores de cor a serem exibidos.

Interação do usuário com o componente ColorPicker

Um ColorPicker permite que um usuário selecione uma cor e a aplique a outro objeto no aplicativo. Por exemplo, se você deseja permitir que o usuário personalize elementos do aplicativo, como uma cor de fundo ou a cor do texto, inclua um ColorPicker e aplique a cor selecionada pelo usuário.

O usuário escolhe uma cor clicando em sua amostra no painel ou digitando seu valor hexadecimal no campo de texto. Depois que o usuário selecionar uma cor, use a propriedade `selectedColor` do ColorPicker para aplicar a cor ao texto ou a outro objeto no aplicativo.

Uma ocorrência de ColorPicker receberá o foco se um usuário mover o ponteiro sobre ela ou acessá-la por meio da tecla `tab`. Quando o painel de amostras de um ColorPicker for aberto, você poderá usar as seguintes teclas para controlá-lo:

| Tecla | Descrição |
|--------------------|---|
| Início | Mova a seleção para a primeira cor no painel de amostras. |
| Seta para cima | Mova a seleção uma linha acima no painel de amostras. |
| Seta para baixo | Mova a seleção uma linha abaixo no painel de amostras. |
| Seta para direita | Mova a seleção uma cor para a direita no painel de amostras. |
| Seta para esquerda | Mova a seleção uma cor para a esquerda no painel de amostras. |
| End | Mova a seleção para a última cor no painel de amostras. |

Parâmetros do componente ColorPicker

Você pode definir os seguintes parâmetros de criação no Inspetor de propriedades ou no Inspetor de componentes para cada ocorrência de ColorPicker: `selectedColor` e `showTextField`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe ColorPicker na [Referência do ActionScript 3.0 para Flash Professional](#).

Criar um aplicativo com o componente ColorPicker

O exemplo a seguir adiciona um componente ColorPicker a um aplicativo durante a criação. Neste exemplo, cada vez que você altera a cor no ColorPicker, a função `changeHandler()` chama a função `drawBox()` para desenhar um nova caixa com a cor selecionada no ColorPicker.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um ColorPicker do painel Componentes para o centro do Palco e atribua a ele o nome de ocorrência **aCp**.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme.
- 5 Clique no ColorPicker e selecione uma cor para colorir a caixa.

Criar um ColorPicker por meio do ActionScript

Este exemplo usa o construtor `ColorPicker()` e o método `addChild()` para criar um `ColorPicker` no Palco. Ele define a propriedade `colors` para os valores de cor vermelho (0xFF0000), verde (0x00FF00) e azul (0x0000FF) a fim de especificar as cores que o `ColorPicker` exibirá. Ele também cria uma `TextArea` e, cada vez que você selecionar uma cor diferente no `ColorPicker` o exemplo alterará a cor do texto na `TextArea` para que seja correspondente.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente `ColorPicker` do painel Componentes para o painel Biblioteca.
- 3 Arraste o componente `TextArea` do painel Componentes para o painel Biblioteca.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do ActionScript:

Uso dos componentes de UI

```
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 Selezione Controllar > Testar filme.

Uso do componente ComboBox

O componente ComboBox permite que o usuário faça uma única seleção em uma lista suspensa. Uma ComboBox pode ser estática ou editável. Uma ComboBox editável permite que um usuário digite o texto diretamente no campo de texto, na parte superior da lista. Se a lista suspensa chegar ao final do documento, ela abrirá para cima, e não para baixo. O ComboBox é constituído por três subcomponentes: os componentes BaseButton, TextInput e List.

Em uma ComboBox editável, apenas o botão é a área de pressionamento, e não a caixa de texto. Em uma ComboBox estática, o botão e a caixa de texto constituem a área de pressionamento. Essa área responde quando a lista suspensa é aberta ou fechada.

Quando o usuário faz uma seleção na lista, com o mouse ou através do teclado, o rótulo da seleção é copiado para o campo de texto na parte superior da ComboBox.

Interação do usuário com o componente ComboBox

Você pode usar um componente ComboBox em qualquer formulário ou aplicativo que requeira a seleção de um item em uma lista. Por exemplo, é possível fornecer uma lista suspensa de estados em um formulário de endereço de cliente. Você pode usar uma ComboBox editável para obter cenários mais complexos. Por exemplo, em um aplicativo que forneça orientações de direção, você pode usar uma ComboBox editável para permitir que um usuário especifique seus endereços de origem e destino. A lista suspensa conterá os endereços especificados anteriormente.

Se a ComboBox for editável, significando que a propriedade `editable` está definida como `true`, as teclas a seguir removerão o foco da caixa de entrada de texto e deixarão o valor anterior. A exceção é a tecla Enter, que aplicará o novo valor primeiro se o usuário tiver digitado algum texto.

| Tecla | Descrição |
|-----------------|---|
| Shift + Tab | Move o foco para o item anterior. Se um novo item for selecionado, o evento <code>change</code> será despachado. |
| Tabulação | Move o foco para o item seguinte. Se um novo item for selecionado, o evento <code>change</code> será despachado. |
| Seta para baixo | Move a seleção um item para baixo. |
| End | Move a seleção para o final da lista. |
| Escape | Fecha a lista suspensa e retorna o foco para a ComboBox. |
| Enter | Fecha a lista suspensa e retorna o foco para a ComboBox. Quando a ComboBox é editável e o usuário digita texto, a tecla Enter define o valor para o texto digitado. |
| Início | Move a seleção para o início da lista. |
| Page Up | Move a seleção uma página para cima. |
| Page Down | Move a seleção uma página para baixo. |

Quando você adicionar o componente ComboBox a um aplicativo, poderá torná-lo acessível a um leitor de tela adicionando as seguintes linhas do código do ActionScript:

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

A acessibilidade de um componente é ativada somente uma vez, independentemente de quantas ocorrências do componente existem.

Parâmetros do componente ComboBox

Você pode definir os seguintes parâmetros de criação no Inspetor de propriedades ou no Inspetor de componentes para cada ocorrência de ComboBox: `dataProvider`, `editable`, `prompt` e `rowCount`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe ComboBox na [Referência do ActionScript 3.0 para Flash Professional](#). Para obter informações sobre o uso do parâmetro `dataProvider`, consulte “Uso do parâmetro `dataProvider`” na página 31.

Criar um aplicativo com o componente ComboBox

O procedimento a seguir descreve como adicionar um componente ComboBox a um aplicativo durante a criação. A ComboBox é editável e se você digitar **Add** no campo de texto, o exemplo adicionará um item à lista suspensa.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste uma ComboBox ao Palco e atribua a ela o nome de ocorrência **aCb**. Na aba Parâmetros, defina o parâmetro `editable` como `true`.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código:

```
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 Selecione Controlar > Testar filme.

Criar uma ComboBox por meio do ActionScript

O exemplo a seguir cria uma ComboBox com o ActionScript e a preenche com uma lista das universidades da área de São Francisco, Califórnia. Ele define a propriedade `width` da ComboBox para acomodar a largura do texto do prompt e define a propriedade `dropdownWidth` de modo que seja um pouco mais larga para acomodar o maior nome de universidade.

O exemplo cria a lista de universidades em uma ocorrência de matriz, usando a propriedade `label` para armazenar os nomes de escola e a propriedade `data` para armazenar as URLs do site de cada escola. Ele atribui a matriz à ComboBox definindo sua propriedade `dataProvider`.

Quando um usuário seleciona uma universidade na lista, esse exemplo aciona um evento `Event.CHANGE` e uma chamada para a função `changeHandler()`, que carrega a propriedade `data` em uma solicitação de URL para acessar o site da escola.

Observe que a última linha define a propriedade `selectedIndex` da ocorrência de ComboBox como `-1` para reexibir o prompt quando a lista é fechada. Do contrário, o prompt será substituído pelo nome da escola selecionada.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente ComboBox do painel Componentes para o painel Biblioteca.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme.

Você pode implementar e executar esse exemplo no ambiente de criação do Flash, mas receberá mensagens de aviso se tentar acessar os sites da universidade clicando nos itens da ComboBox. Para acessar a ComboBox funcional na Internet, visite:

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

Uso do componente DataGrid

O componente DataGrid permite exibir os dados em linhas e colunas, extraindo os dados de um array ou de um arquivo XML externo que pode ser inserido em uma array do DataProvider. O componente DataGrid inclui rolagem vertical e horizontal, suporte a eventos (incluindo suporte a células editáveis) e recursos de classificação.

Você pode redimensionar e personalizar características como a fonte, a cor e as bordas das colunas em uma grade. Use um clipe de filme personalizado como um renderizador de célula para qualquer coluna em uma grade. (Um renderizador de célula exibe o conteúdo de uma célula.) É possível desativar as barras de rolagem e usar os métodos DataGrid para criar uma exibição de estilo de visualização de página. Para obter mais informações sobre personalização, consulte a classe DataGridColumn na *Referência do ActionScript 3.0 para Flash Professional*.

Mais tópicos da Ajuda

[Criando, preenchendo e redimensionando o componente DataGrid](#)

[Personalizando e classificando o componente DataGrid](#)

[Filtrando e formatando dados no componente DataGrid](#)

Interação do usuário com o componente DataGrid

Você pode usar o mouse e o teclado para interagir com um componente DataGrid.

Se a propriedade `sortableColumns` e a propriedade `sortable` da coluna forem `true`, e você clicar em um cabeçalho, os dados serão classificados com base nos valores da coluna. Você pode desativar a classificação de uma coluna individual definindo sua propriedade `sortable` como `false`.

Se a propriedade `resizableColumns` for `true`, você poderá redimensionar as colunas arrastando os divisores de coluna na linha do cabeçalho.

Se você clicar em uma célula editável, o foco ficará nessa célula; se você clicar em uma célula não-editável, o foco não será afetado. Uma célula é editável quando as propriedades `DataGrid.editable` e `DataGridColumn.editable` da célula são `true`.

Para obter mais informações, consulte as classes `DataGrid` e `DataGridColumn` na [Referência do ActionScript® 3.0 para Adobe Flash® Professional CS5](#).

Quando uma ocorrência de `DataGrid` receber o foco após um clique ou o pressionamento da tecla `tab`, você poderá usar as seguintes teclas para controlá-la:

| Tecla | Descrição |
|-------------------------|--|
| Seta para baixo | Quando uma célula está sendo editada, o ponto de inserção é deslocado para o final do texto da célula. Se uma célula não for editável, a tecla de seta para baixo manipulará a seleção da mesma maneira que o componente <code>List</code> . |
| Seta para cima | Quando uma célula está sendo editada, o ponto de inserção é deslocado para o início do texto da célula. Se uma célula não for editável, a tecla de seta para cima manipulará a seleção da mesma maneira que o componente <code>List</code> . |
| Shift+Up/Seta para cima | Se a <code>DataGrid</code> não for editável e <code>allowMultipleSelection</code> for <code>true</code> , selecionará linhas contíguas. A reversão da direção com a seta oposta desmarca as linhas selecionadas até que você passe para a linha inicial, na qual as linhas dessa direção são selecionadas. |
| Shift+Click | Se <code>allowMultipleSelection</code> for <code>true</code> , selecionará todas as linhas entre a linhas selecionada e a posição atual do circunflexo (célula realçada). |
| Ctrl+Click | Se <code>allowMultipleSelection</code> for <code>true</code> , selecionará linhas adicionais, que não precisam ser contíguas. |
| Seta para direita | Quando uma célula está sendo editada, o ponto de inserção é deslocado um caractere para a direita. Se uma célula não for editável, a tecla de seta para direita não terá efeito nenhum. |
| Seta para esquerda | Quando uma célula está sendo editada, o ponto de inserção é deslocado um caractere para a esquerda. Se uma célula não for editável, a tecla de seta para esquerda não terá efeito nenhum. |
| Início | Seleciona a primeira linha da <code>DataGrid</code> . |
| End | Seleciona a última linha da <code>DataGrid</code> . |
| PageUp | Seleciona a primeira linha em uma página da <code>DataGrid</code> . Uma página consiste no número de linhas que a <code>DataGrid</code> pode exibir sem rolagem. |

| Tecla | Descrição |
|--------------------------|---|
| PageDown | Seleciona a última linha em uma página da DataGrid. Uma página consiste no número de linhas que a DataGrid pode exibir sem rolagem. |
| Return/Enter/Shift+Enter | Quando uma célula é editável, a alteração é confirmada e o ponto de inserção é movido para a célula na mesma coluna, na linha seguinte (para cima ou para baixo, dependendo da alternância do deslocamento). |
| Shift+Tab/Tab | Se a DataGrid for editável, o foco será movido para o item anterior/seguinte até que o final da coluna seja atingido, e depois para a linha anterior/seguinte até que a primeira ou última célula seja atingida. Se a primeira célula for selecionada, Shift+Tab moverá o foco para o controle anterior. Se a última célula for selecionada, a tecla Tab moverá o foco para o controle seguinte. Se a DataGrid não for editável, o foco será movido para o controle anterior/seguinte. |

Você pode usar o componente DataGrid como base para diversos tipos de aplicativos orientados a dados. É possível exibir facilmente uma visualização tabular dos dados, mas também é possível usar os recursos de renderizador de célula para criar partes mais sofisticadas e editáveis da interface do usuário. Estas são formas práticas de usar o componente DataGrid:

- Um cliente de webmail
- Páginas de resultados de pesquisa
- Aplicativos de planilha, como calculadoras de empréstimo e aplicativos de formulário de imposto

Quando você criar um aplicativo com o componente DataGrid, ele será útil para a compreensão do design do componente List, pois a classe DataGrid estende a classe SelectableList. Para obter mais informações sobre a classe SelectableList e o componente List, consulte as classes SelectableList e List na [Referência do ActionScript® 3.0 para Adobe Flash® Professional CS5](#).

Quando você adicionar um componente DataGrid a um aplicativo, poderá torná-lo acessível a um leitor de tela adicionando as seguintes linhas do código do ActionScript:

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

A acessibilidade de um componente é ativada somente uma vez, independentemente de quantas ocorrências o componente tem. Para obter mais informações, consulte o Capítulo 18, “Criação de conteúdo acessível”, em *Uso do Flash*.

Parâmetros do componente DataGrid

Você pode definir os seguintes parâmetros de criação no Inspetor de propriedades ou no Inspetor de componentes para cada ocorrência de DataGrid: `allowMultipleSelection`, `editable`, `headerHeight`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `resizableColumns`, `rowHeight`, `showHeaders`, `verticalLineScrollSize`, `verticalPageScrollSize` e `verticalScrollPolicy`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe DataGrid na [Referência do ActionScript 3.0 para Flash Professional](#).

Criar um aplicativo com o componente DataGrid

Para criar um aplicativo com o componente DataGrid, você deve primeiro descobrir o local de origem dos dados. Geralmente, os dados vêm de uma matriz, que você pode puxar para a grade definindo a propriedade `dataProvider`. Você também pode usar os métodos das classes DataGrid e DataGridColumn para adicionar dados à grade.

Use um provedor de dados local com um componente DataGrid:

Este exemplo cria a DataGrid para exibir a lista de uma equipe de beisebol. Ele define a lista em uma matriz (`aRoster`) e a atribui à propriedade `dataProvider` da DataGrid.

- 1 No Flash, selecione Arquivo > Novo e selecione Arquivo Flash (ActionScript 3.0).
- 2 Arraste o componente DataGrid do painel Componentes para o Palco.
- 3 No Inspetor de propriedades, digite o nome de ocorrência **aDg**.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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);
};
```

A função `bldRosterGrid()` define o tamanho da DataGrid e define a ordem das colunas e seus respectivos tamanhos.

- 5 Selecione Controlar > Testar filme.

Especificação de colunas e adição de classificação para um componente DataGrid em um aplicativo

Observe que você pode clicar em qualquer cabeçalho de coluna para classificar o conteúdo da DataGrid em ordem decrescente pelos valores dessa coluna.

O exemplo a seguir usa o método `addColumn()` para adicionar instâncias de `DataGridColumn` a `DataGrid`. As colunas representam nomes de jogador e suas pontuações. O exemplo também define a propriedade `sortOptions` para especificar as opções de classificação de cada coluna: `Array.CASEINSENSITIVE` para a coluna Nome e `Array.NUMERIC` para a coluna Pontuação. Ele dimensiona a `DataGrid` corretamente definindo o tamanho para o número de linhas e a largura para 200.

- 1 No Flash, selecione Arquivo > Novo e selecione Arquivo Flash (ActionScript 3.0).
- 2 Arraste o componente `DataGrid` do painel Componentes para o Palco.
- 3 No Inspetor de propriedades, digite o nome de ocorrência **aDg**.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme.

Criar uma ocorrência de componente `DataGrid` usando o ActionScript

Este exemplo cria uma `DataGrid` usando o ActionScript e a preenche com uma matriz de nomes de jogador e pontuações.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente `DataGrid` do painel Componentes para o painel Biblioteca do documento atual.
Esse procedimento adicionará o componente à biblioteca, mas ele não ficará visível no aplicativo.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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);
```

Esse código cria a ocorrência de `DataGrid` e, em seguida, dimensiona e posiciona a grade.

- 4 Crie uma matriz, adicione dados a ela e identifique a matriz como provedor de dados da `DataGrid`:

```
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 Selecione Controlar > Testar filme.

Carregar uma DataGrid com um arquivo XML

O exemplo a seguir usa a classe DataGridColumn para criar as colunas da DataGrid. Ele preenche a DataGrid passando um objeto XML como parâmetro `value` do construtor `DataProvider()`.

- 1 Utilize um editor de texto para criar um arquivo XML com os dados a seguir e salve-o com o nome `team.xml` na mesma pasta onde você salvou o arquivo 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 Crie um novo documento Flash (ActionScript 3.0).
- 3 No painel Componentes, clique duas vezes no componente DataGrid para adicioná-lo ao Palco.
- 4 No Inspetor de propriedades, digite o nome de ocorrência **aDg**.
- 5 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme.

Uso do componente Label

O componente Label exibe uma única linha de texto, geralmente para identificar algum outro elemento ou atividade em uma página da Web. Você pode especificar que um rótulo seja formatado com HTML para tirar proveito de suas tags de formatação de texto. Também é possível controlar o alinhamento e o tamanho de um rótulo. Os componentes Label não têm bordas, não podem receber foco e não transmitem nenhum evento.

Uma visualização ao vivo de cada ocorrência de Label reflete as alterações feitas nos parâmetros no Inspetor de propriedades ou no Inspetor de componentes durante a criação. O rótulo não tem uma borda; portanto, a única maneira de obter sua visualização ao vivo é definindo seu parâmetro text.

Interação do usuário com o componente Label

Use um componente Label para criar um rótulo de texto para outro componente em um formulário, como um rótulo “Nome:” à esquerda de um campo TextInput que aceita o nome de um usuário. É recomendável usar um componente Label em vez de um campo de texto sem formatação, pois, dessa forma, você pode usar estilos para manter uma aparência consistente.

Para girar um componente Label, você deve incorporar as fontes; do contrário, elas não serão exibidas quando você testar o filme.

Parâmetros do componente Label

Você pode definir os seguintes parâmetros de autoria no Inspetor de propriedades ou no Inspetor de componentes para cada instância do componente Label: `autoSize`, `condenseWhite`, `selectable`, `text` e `wordWrap`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe Label na [Referência do ActionScript 3.0 para Flash Professional](#).

Criar um aplicativo com o componente Label

O procedimento a seguir explica como adicionar um componente Label a um aplicativo durante a criação. Neste exemplo, o rótulo simplesmente exibe o texto “Data de expiração”.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente Label do painel Componentes para o Palco e atribua a ele os seguintes valores no Inspetor de propriedades:
 - Digite **aLabel** como nome de ocorrência.
 - Digite **80** como valor L.
 - Digite **100** como valor X.
 - Digite **100** como valor Y.
 - Digite **Data de Expiração** como parâmetro `text`.
- 3 Arraste um componente TextArea para o Palco e atribua a ele os seguintes valores no Inspetor de propriedades:
 - Digite **aTa** como nome de ocorrência.
 - Digite **22** como valor A.
 - Digite **200** como valor X.
 - Digite **100** como valor Y.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme.

Criar um ocorrência de componente Label usando o ActionScript

O exemplo a seguir cria um parâmetro Label usando o ActionScript. Ele usa um rótulo para identificar a função de um componente ColorPicker e usa a propriedade `htmlText` para aplicar formatação ao texto do rótulo.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente Label do painel Componentes para o painel Biblioteca do documento atual.
- 3 Arraste o componente ColorPicker do painel Componentes para o painel Biblioteca do documento atual.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme.

Uso do componente List

O componente List é uma caixa de listagem rolável de seleção única ou múltipla. Uma lista também pode exibir gráficos, incluindo outros componentes. Adicione os itens exibidos na lista usando a caixa de diálogo Valores que aparece quando você clica nos rótulos ou campos de parâmetro de dados. Você também pode usar os métodos `List.addItem()` e `List.addItemAt()` para adicionar itens à lista.

O componente List usa um índice baseado em zero, no qual o item com o índice 0 é o item superior exibido. Ao adicionar, remover ou substituir itens de lista usando os métodos e propriedades da classe List, talvez seja necessário especificar o índice do item de lista.

Interação do usuário com o componente List

Você pode configurar uma lista para que os usuários possam fazer seleções únicas ou múltiplas. Por exemplo, um usuário que está visitando um site de comércio eletrônico precisa selecionar o item a ser comprado. Existe uma lista com 30 itens que o usuário pode rolar e clicar em algum para selecionar.

Você também pode criar uma lista que use cliques de filme personalizados como linhas, a fim de que possa exibir mais informações para o usuário. Por exemplo, em um aplicativo de email, cada caixa de correio pode ser um componente List e cada linha pode ter ícones para indicar a prioridade e o status.

A lista recebe o foco quando você clica nela ou a acessa por meio da tecla tab. Em seguida, você pode usar as seguintes teclas para controlá-la:

| Tecla | Descrição |
|----------------------|---|
| Teclas alfanuméricas | Salta para o próximo item que tem <code>Key.getAsci()</code> como primeiro caractere em seu rótulo. |
| Control | Alterna a tecla que permite fazer e remover várias seleções não contíguas. |
| Seta para baixo | A seleção é movida um item para baixo. |
| Início | A seleção é movida para o início da lista. |
| Page Down | A seleção é movida uma página para baixo. |
| Page Up | A seleção é movida uma página para cima. |
| Shift | Permite a seleção contígua. |
| Seta para cima | A seleção é movida um item para cima. |

Nota: Observe que os tamanhos de rolagem são indicado em pixels, não em linhas.

Nota: O tamanho da página usado pelas teclas *Page Up* e *Page Down* é um número menor que o número de itens que cabem na tela. Por exemplo, a paginação em uma lista suspensa de dez linhas mostrará os itens 0-9, 9-18, 18-27 e assim sucessivamente, com uma sobreposição de item por página.

Para obter mais informações sobre o controle do foco, consulte a interface `IFocusManager` e a classe `FocusManager` na [Referência do ActionScript 3.0 para Flash Professional](#) e “Trabalho com o `FocusManager`” na página 29.

Uma visualização ao vivo de cada ocorrência de `List` no Palco reflete as alterações feitas nos parâmetros no Inspetor de propriedades ou no Inspetor de componentes durante a criação.

Quando você adicionar o componente `List` a um aplicativo, poderá torná-lo acessível a um leitor de tela adicionando as seguintes linhas do código do ActionScript:

```
import fl.accessibility.ListAccImpl;

ListAccImpl.enableAccessibility();
```

A acessibilidade de um componente é ativada somente uma vez, independentemente de quantas ocorrências o componente tem. Para obter mais informações, consulte o Capítulo 18, “Criação de conteúdo acessível”, em *Uso do Flash*.

Parâmetros do componente List

Você pode definir os seguintes parâmetros no Inspetor de propriedades ou no Inspetor de componentes de cada instância do componente `List`: `allowMultipleSelection`, `dataProvider`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `multipleSelection`, `verticalLineScrollSize`, `verticalPageScrollSize` e `verticalScrollPolicy`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe `List` na [Referência do ActionScript 3.0 para Flash Professional](#). Para obter informações sobre o uso do parâmetro `dataProvider`, consulte “Uso do parâmetro `dataProvider`” na página 31.

Criar um aplicativo com o componente List

O procedimento a seguir descreve como adicionar um componente `List` a um aplicativo durante a criação.

Adição de um componente List simples a um aplicativo

Neste exemplo, a lista consiste em rótulos que identificam modelos de carro e campos de dados que contêm preços.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente List do painel Componentes para o Palco.
- 3 No Inspetor de propriedades, faça o seguinte:
 - Digite o nome de ocorrência **aList**.
 - Atribua o valor **200** à L (largura).
- 4 Use a ferramenta Texto para criar um campo de texto abaixo de `aList` e atribua a ele o nome de ocorrência **aTf**.
- 5 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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;
}
```

Esse código usa o método `addItem()` para preencher `aList` com três itens, atribuindo um valor `label` a cada um, que aparece na lista, e um valor `data`. Quando você seleciona um item na lista, o ouvinte de eventos chama a função `showData()`, que exibe o valor `data` para o item selecionado.

- 6 Selecione Controlar > Testar filme para compilar e executar esse aplicativo.

Preenchimento de uma ocorrência de List com um provedor de dados

Esse exemplo cria uma lista de modelos de carro e seus preços. No entanto, ele usa um provedor de dados para preencher a lista, em vez do método `addItem()`.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente List do painel Componentes para o Palco.
- 3 No Inspetor de propriedades, faça o seguinte:
 - Digite o nome de ocorrência **aList**.
 - Atribua o valor **200** à L (largura).
- 4 Use a ferramenta Texto para criar um campo de texto abaixo de `aList` e atribua a ele o nome de ocorrência **aTf**.
- 5 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme para ver a lista com seus itens.

Uso de um componente List para controlar uma ocorrência de MovieClip

O exemplo a seguir cria uma lista de nomes de cor e, quando você seleciona um, ele aplica a cor a um MovieClip.

- 1 Crie um documento Flash (ActionScript 3.0).
- 2 Arraste o componente List do painel Componentes para o Palco e atribua a ele os seguintes valores no Inspetor de propriedades:
 - Digite **aList** como nome de ocorrência.
 - Digite **60** como valor A.
 - Digite **100** como valor X.
 - Digite **150** como valor Y.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do ActionScript:

```
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 Selecione Controlar > Testar filme para executar o aplicativo.
- 5 Clique nas cores na lista para vê-las exibidas em um MovieClip.

Criar uma ocorrência de componente List usando o ActionScript

Este exemplo cria uma lista simples usando o ActionScript e a preenche usando o método `addItem()`.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente List do painel Componentes para o painel Biblioteca.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme para executar o aplicativo.

Uso do componente NumericStepper

O componente `NumericStepper` permite que um usuário percorra um conjunto ordenado de números. O componente consiste em um número em uma caixa de texto exibida ao lado de pequenos botões de seta para cima e seta para baixo. Quando um usuário pressiona os botões, o número é aumentado ou diminuído incrementalmente de acordo com a unidade especificada no parâmetro `stepSize`, até que usuário libere os botões ou até que o valor máximo ou mínimo seja atingido. O texto na caixa de texto do componente `NumericStepper` também é editável.

Uma visualização ao vivo de cada ocorrência de `NumericStepper` reflete a configuração do parâmetro `value` no Inspetor de propriedades ou no Inspetor de componentes. No entanto, não há interação de mouse ou teclado com os botões de seta do `NumericStepper` na visualização ao vivo.

Interação do usuário com o componente NumericStepper

Você pode usar o componente `NumericStepper` sempre que um usuário precisar selecionar um valor numérico. Por exemplo, você pode usar um componente `NumericStepper` em um formulário para definir o mês, o dia e o ano da data de expiração de um cartão de crédito. É possível também usar um componente `NumericStepper` para permitir que um usuário aumente ou diminua um tamanho de fonte.

O componente `NumericStepper` manipula somente dados numéricos. Além disso, você deve redimensionar o escalonador durante a criação para exibir mais de duas casas decimais (por exemplo, os números 5246 ou 1,34).

É possível ativar ou desativar um `NumericStepper` em um aplicativo. No estado desativado, um `NumericStepper` não recebe entradas de mouse ou de teclado. Quando estiver ativado, o `NumericStepper` receberá o foco se você clicar nele ou se acessá-lo por meio da tecla `tab`, e seu foco interno será definido para a caixa de texto. Quando uma ocorrência de `NumericStepper` estiver com o foco, você poderá usar as seguintes teclas para controlá-la:

| Tecla | Descrição |
|--------------------|--|
| Seta para baixo | O valor é alterado em uma unidade. |
| Seta para esquerda | Move o ponto de inserção para a esquerda dentro da caixa de texto. |
| Seta para direita | Move o ponto de inserção para a direita dentro da caixa de texto. |
| Shift+Tab | Move o foco para o objeto anterior. |
| Tabulação | Move o foco para o próximo objeto. |
| Seta para cima | O valor é alterado em uma unidade. |

Para obter mais informações sobre o controle do foco, consulte a classe `FocusManager` na [Referência do ActionScript 3.0 para Flash Professional](#) e “[Trabalho com o FocusManager](#)” na página 29.

Parâmetros do componente `NumericStepper`

Você pode definir os seguintes parâmetros no Inspetor de propriedades ou no Inspetor de componentes de cada instância de `NumericStepper`: `maximum`, `minimum`, `stepSize` e `value`. Cada um desses parâmetros tem uma propriedade `ActionScript` correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe `NumericStepper` na [Referência do ActionScript 3.0 para Flash Professional](#).

Criar um aplicativo com o `NumericStepper`

O procedimento a seguir explica como adicionar um componente `NumericStepper` a um aplicativo durante a criação. O exemplo coloca um componente `NumericStepper` e um componente `Label` no Palco, e cria um ouvinte para um evento `Event.CHANGE` na ocorrência de `NumericStepper`. Quando o valor no escalonador numérico for alterado, o exemplo exibirá o novo valor na propriedade `text` da ocorrência de `Label`.

- 1 Arraste um componente `NumericStepper` do painel Componentes para o Palco.
- 2 No Inspetor de propriedades, digite o nome de ocorrência **aNs**.
- 3 Arraste um componente `Label` do painel Componentes para o Palco.
- 4 No Inspetor de propriedades, digite o nome de ocorrência **aLabel**.
- 5 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do `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;
};
```

Este exemplo define a propriedade `text` do rótulo para o valor do `NumericStepper`. A função `changeHandler()` atualiza a propriedade `text` do rótulo sempre que o valor na ocorrência de `NumericStepper` é alterado.

- 6 Selecione Controlar > Testar filme.

Criar um NumericStepper por meio do ActionScript

Este exemplo cria três NumericStepper com o código do ActionScript, para criar o mês, o dia e o ano da data de nascimento do usuário. Ele também adiciona rótulos para um prompt e para os identificadores de cada um dos NumericStepper.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um rótulo para o painel Biblioteca.
- 3 Arraste um componente NumericStepper para o painel Biblioteca.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme para executar o aplicativo.

Uso do componente ProgressBar

O componente ProgressBar exibe o progresso do carregamento do conteúdo, o que tranquiliza o usuário quando o conteúdo é grande e pode atrasar a execução do aplicativo. O componente ProgressBar é útil para exibir o progresso do carregamento de imagens e de partes de um aplicativo. O processo de carregamento pode ser determinado ou indeterminado. Uma barra de andamento *determinada* é uma representação linear do progresso de uma tarefa ao longo do tempo, e é utilizada quando o volume de conteúdo a ser carregado é conhecido. Uma barra de andamento *indeterminada* é usada quando o volume do conteúdo a ser carregado é desconhecido. Você também pode adicionar um componente Label para exibir o progresso do carregamento como percentual.

O componente ProgressBar usa um dimensionamento de 9 fatias, e tem uma capa de barra, capa de faixa e uma capa indeterminada.

Interação do usuário com o componente ProgressBar

Existem três modos em que o componente ProgressBar pode ser usado. Os modos mais utilizados são o modo de evento e o modo em poll. Esses modos especificam um processo de carregamento que emite os eventos `progress` e `complete` (modo de evento e em poll) ou expõe as propriedades `bytesLoaded` e `bytesTotal` (modo em poll). É possível também usar o componente ProgressBar no modo manual definindo as propriedades `maximum`, `minimum` e `value` juntamente com chamadas para o método `ProgressBar.setProgress()`. Você pode definir a propriedade `indeterminate` para indicar se ProgressBar terá um preenchimento em listras e uma origem de tamanho desconhecido (`true`) ou um preenchimento sólido e uma origem de tamanho conhecido (`false`).

Defina o modo de ProgressBar especificando sua propriedade `mode` por meio do parâmetro `mode` no Inspetor de propriedades ou no Inspetor de componentes, ou usando o ActionScript.

Se você usar ProgressBar para mostrar o status do processamento, por exemplo, 100.000 itens, caso ela esteja em um loop de quadro único, não haverá nenhuma atualização visível para ProgressBar porque não há redesenhos da tela.

Parâmetros do componente ProgressBar

Você pode definir os seguintes parâmetros no Inspetor de propriedades ou no Inspetor de componentes de cada instância de ProgressBar: `direction`, `mode` e `source`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome.

Você pode escrever o ActionScript de modo a controlar essas e outras opções do componente ProgressBar usando suas propriedades, métodos e eventos. Para obter mais informações, consulte a classe ProgressBar na [Referência do ActionScript 3.0 para Flash Professional](#).

Criar um aplicativo com ProgressBar

O procedimento a seguir mostra como adicionar um componente ProgressBar a um aplicativo durante a criação. Neste exemplo, ProgressBar usa o modo de evento. Nesse modo, o conteúdo de carregamento emite os eventos `progress` e `complete` que ProgressBar despacha para indicar o progresso. Quando o evento `progress` ocorre, o exemplo atualiza um rótulo para mostrar o percentual do conteúdo carregado. Quando o evento `complete` ocorre, o exemplo exibe “Carregamento concluído” e o valor da propriedade `bytesTotal`, que é o tamanho do arquivo.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente ProgressBar do painel Componentes para o Palco.
 - No Inspetor de propriedades, digite o nome de ocorrência **aPb**.
 - Na seção Parâmetros, digite **200** como valor X.
 - Digite **260** como valor Y.
 - Selecione `event` como parâmetro `mode`.
- 3 Arraste o componente Button do painel Componentes para o Palco.
 - No Inspetor de propriedades, digite **loadButton** como nome de ocorrência.
 - Digite **220** como parâmetro X.
 - Digite **290** como parâmetro Y.
 - Digite **Carregar som** como parâmetro `label`.
- 4 Arraste o componente Label para o Palco e atribua a ele o nome de ocorrência **progLabel**.
 - Digite **150** como valor L.
 - Digite **200** como parâmetro X.
 - Digite **230** como parâmetro Y.
 - Na seção Parâmetros, limpe o valor do parâmetro `text`.
- 5 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do ActionScript, que carrega um arquivo de áudio 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 Selecione Controlar > Testar filme.

Criar um aplicativo com o componente ProgressBar no modo em poll

O exemplo a seguir define ProgressBar no modo em poll. No modo em poll, o progresso é determinado através da escuta dos eventos `progress` no conteúdo que está sendo carregado e através do uso das propriedades `bytesLoaded` e `bytesTotal` para calcular o progresso. Este exemplo carrega um objeto `Sound`, escuta os eventos `progress` e calcula o percentual carregado usando suas propriedades `bytesLoaded` e `bytesTotal`. Ele exibe o percentual carregado em um rótulo e no painel Saída.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente ProgressBar do painel Componentes para o Palco e digite os seguintes valores no Inspetor de propriedades:
 - Digite **aPb** como nome de ocorrência.
 - Digite **185** como valor X.
 - Digite **225** como valor Y.
- 3 Arraste o componente Label para o Palco e digite os seguintes valores no Inspetor de propriedades:
 - Digite **progLabel** como nome de ocorrência.
 - Digite **180** como valor X.

- Digite **180** como valor Y.
- Na seção Parâmetros, limpe o valor do parâmetro text.

4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do ActionScript, que cria um objeto Sound (aSound) e chama loadSound() para carregar um som no objeto 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 Selecione Controlar > Testar filme para executar o aplicativo.

Criar um aplicativo com o componente ProgressBar no modo manual

O exemplo a seguir define ProgressBar no modo manual. Nesse modo, você deve definir o progresso manualmente chamando o método setProgress() e atribuir a ele os valores atual e máximo para determinar a extensão do progresso. Não defina a propriedade source no modo manual. O exemplo usa um componente NumericStepper, com um valor máximo de 250, para incrementar o componente ProgressBar. Quando o valor no NumericStepper for alterado e acionar um evento CHANGE, o manipulador de eventos (nsChangeHandler) chamará o método setProgress() para avançar o ProgressBar. Esse exemplo também exibe a porcentagem concluída de progresso, com base no valor máximo.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente ProgressBar do painel Componentes para o Palco e atribua a ele os seguintes valores no Inspetor de propriedades:
 - Digite **aPb** como nome de ocorrência.
 - Digite **180** como valor X.
 - Digite **175** como valor Y.
- 3 Arraste o componente NumericStepper para o Palco e digite os seguintes valores no Inspetor de propriedades:
 - Digite **aNs** como nome de ocorrência.
 - Digite **220** como valor X.
 - Digite **215** como valor Y.
 - Na seção Parâmetros, digite **250** como parâmetro máximo, **0** como valor mínimo, **1** como parâmetro stepSize e **0** como parâmetro value.

4 Arraste o componente Label para o Palco e digite os seguintes valores no Inspetor de propriedades:

- Digite **progLabel** como nome de ocorrência.
- Digite **150** como valor L.
- Digite **180** como valor X.
- Digite **120** como valor Y.
- Na aba Parâmetros, limpe o valor Label do parâmetro text.

5 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código:

```
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 Selecione Controlar > Testar filme para executar o aplicativo.

7 Clique na seta para cima no NumericStepper para avançar a ProgressBar.

Criar uma ProgressBar por meio do ActionScript

Este exemplo cria uma ProgressBar por meio do ActionScript. Além disso, ele duplica a funcionalidade do exemplo anterior, que cria uma ProgressBar no modo manual.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente ProgressBar para o painel Biblioteca.
- 3 Arraste o componente NumericStepper para o painel Biblioteca.
- 4 Arraste o componente Label para o painel Biblioteca.
- 5 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código:

```
import fl.controls.ProgressBar;  
import fl.controls.NumericStepper;  
import fl.controls.Label;  
import fl.controls.ProgressBarDirection;  
import fl.controls.ProgressBarMode;  
import flash.events.Event;  
  
var aPb:ProgressBar = new ProgressBar();  
var aNs:NumericStepper = new NumericStepper();  
var progLabel:Label = new Label();  
  
addChild(aPb);  
addChild(aNs);  
addChild(progLabel);  
  
aPb.move(180,175);  
aPb.direction = ProgressBarDirection.RIGHT;  
aPb.mode = ProgressBarMode.MANUAL;  
  
progLabel.setSize(150, 22);  
progLabel.move(180, 150);  
progLabel.text = "";  
  
aNs.move(220, 215);  
aNs.maximum = 250;  
aNs.minimum = 0;  
aNs.stepSize = 1;  
aNs.value = 0;  
  
aNs.addEventListener(Event.CHANGE, nsChangeHandler);  
  
function nsChangeHandler(event:Event):void {  
    aPb.setProgress(aNs.value, aNs.maximum);  
    progLabel.text = "Percent of progress = " + int(aPb.percentComplete) + "%";  
}
```

- 6 Selecione Controlar > Testar filme para executar o aplicativo.
- 7 Clique na seta para cima no NumericStepper para avançar a ProgressBar.

Uso do componente RadioButton

O componente RadioButton permite que o usuário seja forçado a fazer uma única escolha em um conjunto de opções. Esse componente deve ser usado em um grupo de, pelo menos duas ocorrências de RadioButton. Apenas um membro do grupo pode ser selecionado, em qualquer momento determinado. Selecionar um botão de rádio em um grupo cancela a seleção do botão de rádio selecionado atualmente no grupo. Defina o parâmetro `groupName` para indicar o grupo ao qual o botão de opção pertence.

Um botão de opção é parte fundamental de vários aplicativos de formulário na Web. Você poderá usar os botões de opção sempre que quiser que um usuário faça uma única escolha em um grupo de opções. Por exemplo, use os botões de opção em um formulário para perguntar qual cartão de crédito um usuário deseja usar.

Interação do usuário com o componente RadioButton

Um botão de opção pode ser ativado ou desativado. Um botão de opção desativado não recebe entradas de mouse nem de teclado. Quando o usuário clica em um grupo do componente RadioButton ou o acessa por meio da tecla tab, apenas o botão de opção selecionado recebe o foco. Em seguida, o usuário poderá utilizar as seguintes teclas para controlá-lo:

| Tecla | Descrição |
|-----------------------------------|--|
| Seta para cima/seta para esquerda | A seleção é movida para o botão de opção anterior no grupo de botões de opção. |
| Seta para baixo/seta para direita | A seleção é movida para o próximo botão de opção no grupo de botões de opção. |
| Tabulação | Move o foco do grupo de botões de opção para o próximo componente. |

Para obter mais informações sobre o controle do foco, consulte a interface IFocusManager e a classe FocusManager na [Referência do ActionScript 3.0 para Flash Professional](#) e “[Trabalho com o FocusManager](#)” na página 29.

Uma visualização ao vivo de cada ocorrência de RadioButton no Palco reflete as alterações feitas nos parâmetros no Inspetor de propriedades ou no Inspetor de componentes durante a criação. No entanto, a exclusão mútua da seleção não é exibida na visualização ao vivo. Se você definir o parâmetro selecionado como `true` para dois botões de opção do mesmo grupo, eles aparecerão selecionados mesmo que somente a última ocorrência criada apareça selecionada em tempo de execução. Para obter mais informações, consulte “[Parâmetros do componente RadioButton](#)” na página 79.

Quando você adicionar o componente RadioButton a um aplicativo, poderá torná-lo acessível a um leitor de tela adicionando as seguintes linhas de código:

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

A acessibilidade de um componente é ativada somente uma vez, independentemente de quantas ocorrências do componente existem. Para obter mais informações, consulte o Capítulo 18, “Criação de conteúdo acessível”, em *Uso do Flash*.

Parâmetros do componente RadioButton

Você pode definir os seguintes parâmetros de autoria no Inspetor de propriedades ou no Inspetor de componentes de cada instância do componente RadioButton: `groupName`, `label`, `LabelPlacement`, `selected` e `value`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe RadioButton na [Referência do ActionScript 3.0 para Flash Professional](#).

Você pode escrever o ActionScript para definir as opções adicionais das ocorrências de RadioButton usando os métodos, as propriedades e os eventos da classe RadioButton.

Criar um aplicativo com o componente RadioButton

O procedimento a seguir explica como adicionar um componente RadioButton a um aplicativo durante a criação. Neste exemplo, os RadioButtons são usados para apresentar uma pergunta sim-ou-não. Os dados do RadioButton são exibidos em uma TextArea.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste dois componentes RadioButton do painel Componentes para o Palco.
- 3 Selecione o primeiro botão de opção. No Inspetor de propriedades, atribua a ele o nome de ocorrência **yesRb** e o nome de grupo **rbGroup**.

- 4 Selecione o segundo botão de opção. No Inspetor de propriedades, atribua a ele o nome de ocorrência **noRb** e o nome de grupo **rbGroup**.
- 5 Arraste um componente TextArea do painel Componentes para o Palco e atribua a ele o nome de ocorrência **aTa**.
- 6 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme para executar o aplicativo.

Criar um RadioButton por meio do ActionScript

Este exemplo usa o ActionScript para criar três RadioButtons para as cores vermelho, azul e verde, e desenha uma caixa cinza. A propriedade `value` de cada RadioButton especifica o valor hexadecimal da cor associada ao botão. Quando um usuário clica em um dos RadioButtons, a função `clickHandler()` chama `drawBox()`, passando a cor da propriedade `value` do RadioButton para colorir a caixa.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente RadioButton para o painel Biblioteca.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme para executar o aplicativo.

Para obter mais informações, consulte a classe `RadioButton` na [Referência do ActionScript 3.0 para Flash Professional](#).

Uso do componente ScrollPane

Você pode usar o componente `ScrollPane` para exibir um conteúdo maior do que a área no qual ele foi carregado. Por exemplo, se você tiver uma imagem grande e se houver somente um pequeno espaço para ela em um aplicativo, será possível carregá-la em um `ScrollPane`. O `ScrollPane` pode aceitar clipes de filme, arquivos JPEG, PNG, GIF e SWF.

Componentes como o ScrollPane e o UILoader têm eventos `complete` que permitem determinar quando o carregamento do conteúdo foi concluído. Se você deseja definir as propriedades no conteúdo de um componente ScrollPane ou UILoader, escute o evento `complete` e defina a propriedade no manipulador de eventos. Por exemplo, o código a seguir cria um ouvinte para o evento `Event.COMPLETE` e um manipulador de eventos que define a propriedade `alpha` do conteúdo do ScrollPane para `.5`:

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

Se você especificar um local ao carregar o conteúdo no ScrollPane, deverá especificar o local (coordenadas X e Y) como 0, 0. Por exemplo, o código a seguir carrega o ScrollPane corretamente porque a caixa é desenhada no local 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
```

Para obter mais informações, consulte a classe ScrollPane na [Referência do ActionScript 3.0 para Flash Professional](#).

Interação do usuário com o componente ScrollPane

Um ScrollPane pode ser ativado ou desativado. Um ScrollPane desativado não recebe entradas de mouse nem de teclado. Um usuário pode utilizar as seguintes teclas para controlar um ScrollPane quando ele tiver o foco:

| Tecla | Descrição |
|--------------------|--|
| Seta para baixo | O conteúdo é movido uma rolagem de linha vertical para cima. |
| Seta para cima | O conteúdo é movido uma rolagem de linha vertical para baixo. |
| End | O conteúdo é movido para o final do ScrollPane. |
| Seta para esquerda | O conteúdo é movido uma rolagem de linha horizontal para a direita. |
| Seta para direita | O conteúdo é movido uma rolagem de linha horizontal para a esquerda. |
| Início | O conteúdo é movido para o início do ScrollPane. |
| End | O conteúdo é movido para o final do ScrollPane. |
| PageDown | O conteúdo é movido uma página de rolagem vertical para cima. |
| PageUp | O conteúdo é movido uma página de rolagem vertical para baixo. |

Um usuário pode utilizar o mouse para interagir com o ScrollPane em seu conteúdo e nas barras de rolagem vertical e horizontal. O usuário poderá arrastar o conteúdo utilizando o mouse quando a propriedade `scrollDrag` estiver definida como `true`. A exibição de um ponteiro de mão no conteúdo indica que o usuário pode arrastar o conteúdo. Diferente da maioria dos outros controles, as ações ocorrem quando o botão do mouse é pressionado e continua até que ele seja solto. Se o conteúdo tiver pargas de tabulação válidas, defina `scrollDrag` como `false`. Do contrário, todos os cliques de mouse no conteúdo ativarão o arrastamento da rolagem.

Parâmetros do componente ScrollPane

Você pode definir os parâmetros a seguir para cada instância de ScrollPane no Inspetor de propriedades ou no Inspetor de componentes: `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `scrollDrag`, `source`, `verticalLineScrollSize`, `verticalPageScrollSize` e `verticalScrollPolicy`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe ScrollPane na [Referência do ActionScript 3.0 para Flash Professional](#).

Você pode escrever o ActionScript de modo a controlar essas e outras opções do componente ScrollPane usando suas propriedades, métodos e eventos.

Criar um aplicativo com o componente ScrollPane

O procedimento a seguir explica como adicionar um componente ScrollPane a um aplicativo durante a criação. Neste exemplo, o ScrollPane carrega uma imagem de um caminho especificado pela propriedade `source`.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente ScrollPane do painel Componentes para o Palco e atribua a ele o nome de ocorrência **aSp**.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme para executar o aplicativo.

Criar uma ocorrência de ScrollPane por meio do ActionScript

O exemplo cria um ScrollPane, define seu tamanho e carrega uma imagem nele usando a propriedade `source`. Ele também cria dois ouvintes. O primeiro escuta um evento `scroll` e exibe a posição da imagem quando o usuário rola a tela verticalmente ou horizontalmente. O segundo escuta um evento `complete` e exibe uma mensagem no painel Saída que informa que o carregamento da imagem foi concluído.

Este exemplo cria um ScrollPane por meio de um ActionScript e coloca um MovieClip (uma caixa vermelha) nele com 150 pixels de largura por 300 pixels de altura.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente ScrollPane do painel Componentes para o painel Biblioteca.
- 3 Arraste o componente DataGrid do painel Componentes para o painel Biblioteca.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme para executar o aplicativo.

Uso do componente Slider

O componente Slider permite que o usuário selecione um valor deslizando uma *miniatura* gráfica entre os pontos de extremidade de uma trilha que corresponde a uma faixa de valores. Você pode usar um controle deslizante para permitir que um usuário escolha um valor como um número ou percentual, por exemplo. Também é possível usar o ActionScript para fazer com que o valor do controle deslizante influencie o comportamento de um segundo objeto. Por exemplo, você pode associar o controle deslizante a uma imagem e reduzi-la ou ampliá-la com base na posição, ou valor, relativo da miniatura do controle deslizante.

O valor atual do controle deslizante é determinado pelo local relativo da miniatura entre os pontos de extremidade da trilha ou os valores mínimo e máximo do controle deslizante.

O controle deslizante permite uma faixa contínua de valores entre seus valores mínimo e máximo, mas você também pode definir o parâmetro `snapInterval` para especificar intervalos entre os valores mínimo e máximo. Um controle deslizante pode mostrar marques de tique, que são independentes dos valores atribuídos do controle deslizante, em intervalos específicos ao longo da trilha.

O controle deslizante tem uma orientação horizontal por padrão, mas você pode atribuir a ele uma orientação vertical, definindo o valor do parâmetro `direction` como vertical. A trilha do controle deslizante é alongada de uma extremidade a outra e as marcas de tique são colocadas da esquerda para a direita imediatamente acima da trilha.

Interação do usuário com o componente Slider

Quando uma ocorrência de Slider estiver com o foco, você poderá usar as seguintes teclas para controlá-la:

| Tecla | Descrição |
|--------------------|---|
| Seta para direita | Aumenta o valor associado de um controle deslizante horizontal. |
| Seta para cima | Aumenta o valor associado de um controle deslizante vertical. |
| Seta para esquerda | Diminui o valor associado de um controle deslizante horizontal. |
| Seta para baixo | Diminui o valor associado de um controle deslizante vertical. |
| Shift+Tab | Move o foco para o objeto anterior. |
| Tabulação | Move o foco para o próximo objeto. |

Para obter mais informações sobre o controle do foco, consulte a interface `IFocusManager` e a classe `FocusManager` na [Referência do ActionScript 3.0 para Flash Professional](#) e “Trabalho com o `FocusManager`” na página 29.

Uma visualização ao vivo de cada ocorrência de `Slider` reflete as alterações feitas nos parâmetros no Inspetor de propriedades ou no Inspetor de componentes durante a criação.

Parâmetros do componente `Slider`

Você pode definir os parâmetros de autoria a seguir no Inspetor de propriedades ou no Inspetor de componentes de cada instância de `Slider`: `direction`, `liveDragging`, `maximum`, `minimum`, `snapInterval`, `tickInterval` e `value`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe `Slider` na [Referência do ActionScript 3.0 para Flash Professional](#).

Criar um aplicativo com o controle deslizante

O exemplo a seguir cria uma ocorrência de `Slider` para permitir que o usuário expresse seu nível de satisfação com algum evento hipotético. O usuário move o controle deslizante para a direita ou esquerda a fim de indicar o nível de satisfação mais alto ou mais baixo.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente `Label` do painel Componentes para o centro do Palco.
 - Atribua a ele o nome de ocorrência **valueLabel**.
 - Atribua o valor **0percent** ao parâmetro `text`.
- 3 Arraste um componente `Slider` do painel Componentes e centralize-o abaixo de `value_lbl`.
 - Atribua a ele o nome de ocorrência **aSlider**.
 - Atribua a ele a largura (L:) **200**.
 - Atribua a ele a altura (L:) **10**.
 - Atribua o valor **100** ao parâmetro `maximum`.
 - Atribua o valor **10** aos parâmetros `snapInterval` e `tickInterval`.
- 4 Arraste outra ocorrência de `Label` do painel Biblioteca e centralize-a abaixo de `aSlider`.
 - Atribua a ele o nome de ocorrência **promptLabel**.
 - Atribua a ele a largura (L:) 250.
 - Atribua a ele a altura (L:) 22.
 - Digite **Informe seu nível de satisfação** no parâmetro `text`.

- 5 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme.

Neste exemplo, quando você move a miniatura do controle deslizante de um intervalo para outro, um ouvinte do evento `SliderEvent.CHANGE` atualiza a propriedade `text` de `valueLabel` para exibir o percentual que corresponde à posição da miniatura.

Criar um aplicativo com o componente Slider por meio do ActionScript

O exemplo a seguir cria um controle deslizante usando o ActionScript. O exemplo baixa uma imagem de uma flor e usa o controle deslizante para permitir que o usuário esmaeaça ou aplique brilho à imagem, alterando sua propriedade `alpha` para corresponder ao valor do controle deslizante.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste os componentes Label e Slider do painel Componentes para o painel Biblioteca do documento atual.
Esse procedimento adicionará os componentes à biblioteca, mas não o tornará visível no aplicativo.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código para criar e posicionar ocorrências de componentes:

```
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 Selecione Controlar > Testar filme para executar o aplicativo.
- 5 Mova a miniatura do controle deslizante para a esquerda para esmaecer a imagem e para a direita para aplicar brilho a ela.

Uso do componente TextArea

O componente TextArea é um invólucro do objeto ActionScript nativo TextField. Você pode usar o componente TextArea para exibir texto e para editar e receber entradas de texto, caso a propriedade `editable` seja `true`. O componente pode exibir ou receber várias linhas de texto e quebrar automaticamente longas linhas de texto caso a propriedade `wordWrap` seja definida como `true`. A propriedade `restrict` permite restringir os caracteres que um usuário pode digitar, enquanto a propriedade `maxChars` permite especificar o número máximo de caracteres a serem digitados. Se o texto ultrapassar os limites horizontal ou vertical da área de texto, as barras de rolagem horizontal e vertical aparecerão automaticamente, a menos que suas propriedades associadas, `horizontalScrollPolicy` e `verticalScrollPolicy`, sejam definidas como `off`.

Você pode usar um componente `TextArea` sempre que precisar de um campo de texto de várias linhas. Por exemplo, você pode usar um componente `TextArea` como um campo de comentário em um formulário. É possível configurar um ouvinte que verifique se o campo está vazio quando um usuário acessa o campo usando a tecla `tab`. Esse ouvinte pode exibir uma mensagem de erro informando que um comentário deve ser digitado no campo.

Se você precisar de um campo de texto com uma única linha, use o componente `TextInput`.

Você pode definir o estilo `textFormat` usando o método `setStyle()` para alterar o estilo do texto que aparece em uma ocorrência de `TextArea`. Também é possível formatar um componente `TextArea` com HTML usando a propriedade `htmlText` no `ActionScript` e definir a propriedade `displayAsPassword` como `true` para mascarar o texto com asteriscos. Se você definir a propriedade `condenseWhite` como `true`, o Flash removerá o espaço branco extra do novo texto resultante de espaços, quebras de linha etc. Isso não terá nenhum efeito sobre o texto que já está no controle.

Interação do usuário com o componente `TextArea`

Um componente `TextArea` pode estar ativado ou desativado em um aplicativo. No estado desativado, ele não pode receber entradas de mouse nem de teclado. Quando ativado, ele segue as mesmas regras de foco, seleção e navegação que o objeto `TextField` do `ActionScript`. Quando uma ocorrência de `TextArea` estiver com o foco, você poderá controlá-la usando as seguintes teclas:

| Tecla | Descrição |
|----------------|--|
| Teclas de seta | Movê o ponto de inserção para cima, para baixo, para esquerda ou para direita no texto, caso este seja editável. |
| Page Down | Movê o ponto de inserção para o final do texto, caso este seja editável. |
| Page Up | Movê o ponto de inserção para o início do texto, caso este seja editável. |
| Shift+Tab | Movê o foco para o objeto anterior no loop de tabulação. |
| Tabulação | Movê o foco para o objeto seguinte no loop de tabulação. |

Para obter mais informações sobre o controle do foco, consulte a classe `FocusManager` na [Referência do ActionScript 3.0 para Flash Professional](#) e “Trabalho com o `FocusManager`” na página 29.

Parâmetros do componente `TextArea`

Você pode definir os parâmetros de autoria a seguir para cada instância do componente `TextArea` no Inspetor de propriedades ou no Inspetor de componentes: `condenseWhite`, `editable`, `horizontalScrollPolicy`, `maxChars`, `restrict`, `text`, `verticalScrollPolicy` e `wordWrap`. Cada um desses parâmetros tem uma propriedade `ActionScript` correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe `TextArea` na [Referência do ActionScript 3.0 para Flash Professional](#).

Uma visualização ao vivo de cada ocorrência de `TextArea` reflete as alterações feitas nos parâmetros no Inspetor de propriedades ou no Inspetor de componentes durante a criação. Se uma barra de rolagem for necessária, ela aparecerá na visualização ao vivo, mas não funcionará. O texto não pode ser selecionado na visualização ao vivo, e não é possível digitar texto na ocorrência do componente no Palco.

Você pode escrever o `ActionScript` de modo a controlar essas e outras opções do componente `TextArea` usando suas propriedades, métodos e eventos. Para obter mais informações, consulte a classe `TextArea` na [Referência do ActionScript 3.0 para Flash Professional](#).

Criar um aplicativo com o componente TextArea

O procedimento a seguir explica como adicionar um componente TextArea a um aplicativo durante a criação. O exemplo configura um manipulador de eventos `focusOut` na ocorrência de TextArea que verifica se o usuário digitou algo na área de texto antes de atribuir o foco a uma parte diferente da interface.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente TextArea do painel Componentes para o Palco e atribua a ele o nome de ocorrência **aTa**. Deixe seus parâmetros definidos com as configurações padrão.
- 3 Arraste um segundo componente TextArea do painel Componentes para o Palco, coloque-o abaixo do primeiro e atribua a ele o nome de ocorrência **bTa**. Deixe seus parâmetros definidos com as configurações padrão.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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();
}
```

Este exemplo restringe os caracteres digitados na área de texto `aTa` a caracteres minúsculos, vírgula, apóstrofo e espaços. Ele também configura os manipuladores dos eventos `change`, `KEY_FOCUS_CHANGE` e `MOUSE_FOCUS_CHANGE` na área de texto `aTa`. A função `changeHandler()` faz com que o texto digitado na área de texto `aTa` apareça automaticamente na área de texto `bTa`, atribuindo `aTa.text` a `bTa.text` em cada evento `change`. A função `k_m_fHandler()` dos eventos `KEY_FOCUS_CHANGE` e `MOUSE_FOCUS_CHANGE` impede que você pressione a tecla `Tab` para mover-se para o campo seguinte sem digitar nenhum texto. Ele faz isso impedindo o comportamento padrão.

- 5 Selecione Controlar > Testar filme.

Se você pressionar a tecla `Tab` para mover o foco para a segunda área de texto sem digitar nenhum texto, uma mensagem de erro será exibida e o foco deverá retornar para a primeira área de texto. Quando você digitar texto na primeira área de texto, o verá duplicado na segunda área de texto.

Criação de uma ocorrência de TextArea por meio do ActionScript

O exemplo a seguir cria um componente TextArea com o ActionScript. Ele define a propriedade `condenseWhite` como `true` para condensar o espaço em branco e atribui o texto à propriedade `htmlText` para tirar proveito dos atributos de formatação de texto HTML.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente TextArea para o painel Biblioteca.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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);
```

Este exemplo usa a propriedade `htmlText` para aplicar atributos HTML negrito e sublinhado a um bloco de texto e exibi-lo na área de texto `a_ta`. O exemplo também define a propriedade `condenseWhite` como `true` para condensar o espaço em branco no bloco de texto. O método `setSize()` define a altura e largura da área de texto, enquanto o método `move()` define sua posição. O método `addChild()` adiciona a ocorrência de `TextArea` ao Palco.

4 Selezione Controlar > Testar filme.

Uso do componente TextInput

`TextInput` é um componente de texto de única linha que representa um invólucro do objeto nativo `TextField` do ActionScript. Se você precisar de um campo de texto de várias linhas, use o componente `TextArea`. Por exemplo, você pode usar um componente `TextInput` como um campo de senha em um formulário. É possível também configurar um ouvinte que verifique se o campo tem caracteres suficientes quando um usuário acessa o campo usando a tecla `tab`. Esse ouvinte pode exibir uma mensagem de erro indicando que o número correto de caracteres deve ser digitado.

Você pode definir o estilo `textFormat` usando o método `setStyle()` para alterar o estilo do texto que aparece em uma ocorrência de `TextInput`. Um componente `TextInput` também pode ser formatado com HTML ou como um campo de senha que mascara o texto.

Interação do usuário com o componente TextInput

Um componente `TextInput` pode estar ativado ou desativado em um aplicativo. No estado desativado, ele não recebe entradas de mouse nem de teclado. Quando ativado, ele segue as mesmas regras de foco, seleção e navegação que o objeto `TextField` do ActionScript. Quando uma ocorrência de `TextInput` estiver com o foco, você poderá usar as seguintes teclas para controlá-la:

| Tecla | Descrição |
|----------------|---|
| Teclas de seta | Movem o ponto de inserção um caractere para a esquerda ou para a direita. |
| Shift+Tab | Move o foco para o objeto anterior. |
| Tabulação | Move o foco para o próximo objeto. |

Para obter mais informações sobre o controle do foco, consulte a classe `FocusManager` na [Referência do ActionScript 3.0 para Flash Professional](#) e “Trabalho com o `FocusManager`” na página 29.

Uma visualização ao vivo de cada ocorrência de `TextInput` reflete as alterações feitas nos parâmetros no Inspetor de propriedades ou no Inspetor de componentes durante a criação. O texto não pode ser selecionado na visualização ao vivo, e não é possível digitar texto na ocorrência do componente no Palco.

Quando você adiciona o componente TextInput a um aplicativo, pode usar o painel Acessibilidade para torná-lo acessível aos leitores de tela.

Parâmetros do componente TextInput

Você pode definir os parâmetros de autoria a seguir para cada instância do componente TextInput no Inspetor de propriedades ou no Inspetor e componentes: `editable`, `displayAsPassword`, `maxChars`, `restrict` e `text`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe TextInput na [Referência do ActionScript 3.0 para Flash Professional](#).

Você pode programar o ActionScript de modo a controlar essas e outras opções do componente TextInput usando suas propriedades, métodos e eventos. Para obter mais informações, consulte a classe TextInput na [Referência do ActionScript 3.0 para Flash Professional](#).

Criação de um aplicativo com o componente TextInput

O procedimento a seguir explica como adicionar um componente TextInput a um aplicativo. O exemplo usa dois campos de TextInput para receber e confirmar uma senha. Ele usa um ouvinte de eventos para verificar se um mínimo de oito caracteres foram digitados e se o texto dos dois campos são correspondentes.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente Label do painel Componentes para o Palco e atribua a ele os seguintes valores no Inspetor de propriedades:
 - Digite o nome de ocorrência **pwdLabel**.
 - Digite um valor L igual a **100**.
 - Digite um valor X igual a **50**.
 - Digite um valor Y igual a **150**.
 - Na seção Parâmetros, digite o valor **Password:** para o parâmetro `text`.
- 3 Arraste um segundo componente Label do painel Componentes para o Palco e atribua a ele os seguintes valores:
 - Digite o nome de ocorrência **confirmLabel**.
 - Digite um valor L igual a **100**.
 - Digite um valor X igual a **50**.
 - Digite um valor Y igual a **200**.
 - Na seção Parâmetros, digite o valor **ConfirmPassword:** para o parâmetro `text`.
- 4 Arraste um componente TextInput do painel Componentes para o Palco e atribua a ele os seguintes valores:
 - Digite o nome de ocorrência **pwdTi**.
 - Digite um valor L igual a **150**.
 - Digite um valor X igual a **190**.
 - Digite um valor Y igual a **150**.
 - Na seção Parâmetros, clique duas vezes no valor do parâmetro `displayAsPassword` e selecione **true**. Isso faz com que o valor digitado no campo de texto seja mascarado com asteriscos.

5 Arraste um segundo componente TextInput do painel Componentes para o Palco e atribua a ele os seguintes valores:

- Digite o nome de ocorrência **confirmTi**.
- Digite um valor L igual a **150**.
- Digite um valor X igual a **190**.
- Digite um valor Y igual a **200**.
- Na seção Parâmetros, clique duas vezes no valor do parâmetro `displayAsPassword` e selecione **true**. Isso faz com que o valor digitado no campo de texto seja mascarado com asteriscos.

6 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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);
```

Esse código configura um manipulador de eventos `enter` na ocorrência de TextInput chamada `confirmTi`. Se as duas senhas não forem correspondentes ou o usuário digitar menos de oito caracteres, o exemplo exibirá a mensagem: “A senha está incorreta. Digite-a novamente.” Se as senhas tiverem oito caracteres ou mais e forem correspondentes, o exemplo exibirá o valor digitado no painel Saída.

7 Selecione Controlar > Testar filme.

Criar uma ocorrência de TextInput por meio do ActionScript

O exemplo a seguir cria um componente TextInput com o ActionScript. Ele também cria um rótulo para solicitar que um usuário digite seu nome. O exemplo define a propriedade `restrict` do componente para permitir somente letras maiúsculas e minúsculas, ponto e espaço. Ele também cria um objeto `TextFormat` usado para formatar o texto nos componentes Label e TextInput.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente TextInput do painel Componentes para o painel Biblioteca.
- 3 Arraste o componente Label do painel Componentes para o painel Biblioteca.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme para executar o aplicativo.

Uso do componente TileList

O componente `TileList` consiste em uma lista que é composta por linhas e colunas de dados fornecidos por um provedor de dados. Um *item* representa uma unidade de dados armazenada em uma célula na `TileList`. Um item, que se origina no provedor de dados, geralmente tem uma propriedade `label` e uma propriedade `source`. A propriedade `label` identifica o conteúdo a ser exibido em uma célula e a propriedade `source` fornece um valor para ela.

Você pode criar uma ocorrência de `Array` ou recuperar uma em um servidor. O componente `TileList` tem métodos que agem como proxy em seu provedor de dados; por exemplo, os métodos `addItem()` e `removeItem()`. Se nenhum provedor de dados externos for fornecido à lista, esses métodos criam automaticamente uma ocorrência de provedor de dados, que é exposta por meio de `List.dataProvider`.

Interação do usuário com o componente TileList

Uma `TileList` renderiza cada célula usando uma entidade gráfica que implementa a interface do `ICellRenderer`. Você pode especificar esse renderizador com a propriedade `cellRenderer` da `TileList`. O `CellRenderer` padrão do componente `TileList` é a `ImageCell`, que exibe uma imagem (classe, bitmap, ocorrência ou URL) e um rótulo opcional. O rótulo é uma única linha que sempre está alinhada à parte inferior da célula. Você pode rolar uma `TileList` apenas em uma direção.

Quando uma ocorrência de `TileList` estiver com o foco, você também poderá usar as seguintes teclas para acessar os itens dentro dela:

| Tecla | Descrição |
|--|--|
| Seta para cima e seta para baixo | Permite a movimentação para cima e para baixo em uma coluna. Se a propriedade <code>allowMultipleSelection</code> for <code>true</code> , você poderá usar essas teclas em combinação com a tecla Shift para selecionar várias células. |
| Seta para esquerda e seta para direita | Permite a movimentação para esquerda ou para direita em uma linha. Se a propriedade <code>allowMultipleSelection</code> for <code>true</code> , você poderá usar essas teclas em combinação com a tecla Shift para selecionar várias células. |
| Início | Seleciona a primeira célula em uma <code>TileList</code> . Se a propriedade <code>allowMultipleSelection</code> for <code>true</code> , e você mantiver pressionada a tecla Shift enquanto pressiona Home, todas as células da seleção atual até a primeira célula serão selecionadas. |
| End | Seleciona a última célula em uma <code>TileList</code> . Se a propriedade <code>allowMultipleSelection</code> for <code>true</code> , e você mantiver pressionada a tecla Shift enquanto pressiona End, todas as células da seleção atual até a última célula serão selecionadas. |
| Ctrl | Se a propriedade <code>allowMultipleSelection</code> for definida como <code>true</code> , você poderá selecionar várias células, em uma ordem não específica. |

Quando você adicionar o componente `TileList` a um aplicativo, poderá torná-lo acessível a um leitor de tela adicionando as seguintes linhas do código do ActionScript:

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

A acessibilidade de um componente é ativada somente uma vez, independentemente de quantas ocorrências o componente tem. Para obter mais informações, consulte o Capítulo 18, “Criação de conteúdo acessível”, em *Uso do Flash*.

Parâmetros do componente `TileList`

Você pode definir os parâmetros de autoria a seguir no Inspetor de propriedades ou no Inspetor de componentes de cada instância do componente `TileList`: `allowMultipleSelection`, `columnCount`, `columnWidth`, `dataProvider`, `direction`, `horizontalScrollLineSize`, `horizontalScrollPageSize`, `labels`, `rowCount`, `rowHeight`, `scrollPolicy`, `verticalScrollLineSize` e `verticalScrollPageSize`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Para obter informações sobre o uso do parâmetro `dataProvider`, consulte “Uso do parâmetro `dataProvider`” na página 31.

Você pode escrever o ActionScript de modo a definir opções adicionais para as ocorrências de `TileList` usando seus métodos, propriedades e eventos. Para obter mais informações, consulte a classe `TileList` na [Referência do ActionScript 3.0 para Flash Professional](#).

Criar um aplicativo com o componente `TileList`

Este exemplo usa `MovieClip` para preencher uma `TileList` com uma matriz de cores de pintura.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente `TileList` para o Palco e atribua a ela o nome de ocorrência **aTL**.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme para testar o aplicativo.

Criação de um componente TileList por meio do ActionScript

Este exemplo cria, de forma dinâmica, uma instância de TileList e adiciona instâncias dos componentes ColorPicker, ComboBox, NumericStepper e CheckBox a ela. Ele cria uma matriz que contém rótulos e os nomes do componente a serem exibidos, e atribui a matriz (dp) à propriedade dataProvider da TileList. Este exemplo usa as propriedades columnWidth e rowHeight e o método setSize() para organizar o layout da TileList, o método move() para posicioná-la no Palco, o estilo contentPadding para colocar espaço entre as bordas da ocorrência de TileList e seu conteúdo, e o método sortItemsOn() para classificar o conteúdo com base em seus rótulos.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste os seguintes componentes do painel Componentes para o painel Biblioteca: ColorPicker, ComboBox, NumericStepper, CheckBox e TileList.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do ActionScript:

```
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 Selecione Testar > Controlar filme para testar o aplicativo.

Uso do componente UILoader

O componente UILoader é um contêiner que pode exibir arquivos SWF, JPEG, JPEG progressivo, PNG e GIF. Você pode usar um carregador de UI sempre que precisar recuperar conteúdo de um local remoto e inseri-lo em um aplicativo Flash. Por exemplo, você pode usar um carregador de UI para adicionar o logotipo de uma empresa (arquivo JPEG) a um formulário. Você também pode usar o componente UILoader em um aplicativo que exiba fotos. Use o método `load()` para carregar o conteúdo, a propriedade `percentLoaded` para determinar o volume de conteúdo que já foi carregado e o evento `complete` para determinar quando o carregamento será concluído.

Você pode dimensionar o conteúdo do carregador de UI ou redimensionar o próprio carregador de UI para acomodar o tamanho do conteúdo. Por padrão, o conteúdo é ajustado ao carregador de UI. Você também pode carregar o conteúdo em tempo de execução e monitorar o progresso do carregamento (embora o conteúdo seja armazenado em cache após seu carregamento, o progresso salta rapidamente para 100%). Se você especificar um local ao carregar o conteúdo no carregador de UI, deverá especificar o local (coordenadas X e Y) como 0, 0.

Interação do usuário com o componente UILoader

Um componente UILoader pode receber o foco. No entanto, o conteúdo carregado no componente UILoader pode aceitar o foco e ter suas próprias interações de foco. Para obter mais informações sobre o controle do foco, consulte a classe `FocusManager` na [Referência do ActionScript 3.0 para Flash Professional](#) e “Trabalho com o `FocusManager`” na página 29.

Parâmetros do componente UILoader

Você pode definir os seguintes parâmetros de autoria no Inspetor de propriedades ou no Inspetor de componentes de cada instância do componente UILoader: `autoLoad`, `maintainAspectRatio`, `source` e `scaleContent`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome.

Uma visualização ao vivo de cada ocorrência de UILoader reflete as alterações feitas nos parâmetros no Inspetor de propriedades ou no Inspetor de componentes durante a criação.

Você pode escrever o ActionScript de modo a definir opções adicionais para as ocorrências de UILoader usando seus métodos, propriedades e eventos. Para obter mais informações, consulte a classe UILoader na [Referência do ActionScript 3.0 para Flash Professional](#).

Criação de um aplicativo com o componente UILoader

O procedimento a seguir explica como adicionar um componente UILoader a um aplicativo durante a criação. Neste exemplo, o carregador obtém uma imagem GIF de um logotipo.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente UILoader do painel Componentes para o Palco.
- 3 No Inspetor de propriedades, digite o nome de ocorrência **aUI**.
- 4 Selecione o carregador no Palco e no Inspetor de componentes, e digite **`http://www.helpexamples.com/images/logo.gif`** como parâmetro `source`.

Criação de uma ocorrência do componente UILoader por meio do ActionScript

Este exemplo cria um componente UILoader com o ActionScript e carrega uma imagem de flor JPEG. Quando o evento `complete` ocorre, ele exibe o número de bytes carregados no painel Saída.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente UILoader do painel Componentes para o painel Biblioteca.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme.

Uso do componente UIScrollBar

O componente UIScrollBar permite adicionar uma barra de rolagem a um campo de texto. Você pode adicionar uma barra de rolagem a um campo de texto durante a criação ou em tempo de execução com o ActionScript. Para usar o componente UIScrollBar, crie um campo de texto no Palco e arraste o componente UIScrollBar do painel Componentes para qualquer quadrante da caixa delimitadora do campo de texto.

Se a barra de rolagem for menor que o tamanho combinado de suas setas de rolagem, ela não será exibida corretamente. Um dos botões de seta ficará oculto atrás do outro. O Flash não fornece verificação de erro para isso. Nesse caso, é recomendável ocultar a barra de rolagem com o ActionScript. Se a barra de rolagem for dimensionada para que haja espaço suficiente para a caixa de rolagem (miniatura), o Flash tornará a caixa de rolagem invisível.

O componente UIScrollBar funciona como qualquer outra barra de rolagem. Ele contém botões de seta em ambas as extremidades, e uma trilha de rolagem e uma caixa de rolagem (miniatura) entre eles. O componente UIScrollBar pode ser anexado a qualquer borda de um campo de texto e usado verticalmente e horizontalmente.

Para obter informações sobre TextField, consulte a classe TextField na [Referência do ActionScript 3.0 para Flash Professional](#).

Interação do usuário com o componente UIScrollBar

Diferente de muitos outros componentes, o componente UIScrollBar pode receber uma entrada de mouse contínua, quando, por exemplo, o usuário mantém pressionado o botão do mouse, em vez de exigir repetidos cliques.

Não há interação do teclado com o componente UIScrollBar.

Parâmetros do componente UIScrollBar

Você pode definir os seguintes parâmetros de autoria no Inspetor de propriedades ou no Inspetor de componentes de cada instância do componente UIScrollBar: `direction` e `scrollTargetName`. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome.

Você pode escrever o ActionScript de modo a definir opções adicionais para as ocorrências de UIScrollBar usando seus métodos, propriedades e eventos. Para obter mais informações, consulte a classe UIScrollBar na [Referência do ActionScript 3.0 para Flash Professional](#).

Criação de um aplicativo com o componente UIScrollBar

O procedimento a seguir descreve como adicionar um componente UIScrollBar a um aplicativo durante a criação.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Crie um campo de texto dinâmico que tenha altura suficiente para reter uma ou duas linhas de texto, e atribua a ele o nome de ocorrência `myText` no Inspetor de propriedades.
- 3 No Inspetor de propriedades, defina o tipo de linha do campo de entrada de texto para Multilinha ou Multilinha sem quebra caso você pretenda usar a barra de rolagem horizontalmente.
- 4 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do ActionScript para preencher a propriedade `text`, a fim de que um usuário precise rolá-lo para vê-lo na íntegra:

```
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."
```

***Nota:** Verifique se o campo de texto no Palco é pequeno demais a ponto de ser necessária a rolagem para que ele seja exibido completamente. Se não for, a barra de rolagem não aparecerá ou poderá aparecer simplesmente como duas linhas sem miniatura (a parte que você arrasta para rolar o conteúdo).*

- 5 Verifique se o encaixe do objeto está ativado (Exibir > Encaixe > Encaixar nos objetos).
- 6 Arraste uma ocorrência de UIScrollBar do painel Componentes para o campo de entrada de texto próximo ao lado ao qual você deseja anexá-lo. O componente deve ser sobreposto ao campo de texto quando você soltar o mouse a fim de que seja corretamente ligado ao campo. Atribua a ele o nome de ocorrência **mySb**.

A propriedade `scrollTargetName` do componente é preenchida automaticamente com o nome de ocorrência do campo de texto no Inspetor de propriedades e no Inspetor de componentes. Se ela não aparecer na aba Parâmetros, é provável que a ocorrência de UIScrollBar não tenha sido suficientemente sobreposta.
- 7 Selecione Controlar > Testar filme.

Criar um componente UIScrollBar por meio do ActionScript

Você pode criar uma ocorrência de UIScrollBar com o ActionScript e associá-la a um campo de texto em tempo de execução. O exemplo a seguir cria uma ocorrência de UIScrollBar orientada horizontalmente e a anexa à parte inferior de uma ocorrência de campo de texto chamada **myTxt**, que é carregada com o texto a partir de uma URL. O exemplo também define o tamanho da barra de rolagem para que corresponda ao tamanho do campo de texto:

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente ScrollBar para o painel Biblioteca.
- 3 Abra o painel Ações, selecione Quadro 1 na Linha do tempo principal e digite o seguinte código do 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 Selecione Controlar > Testar filme.

Capítulo 5: Personalização dos componentes de UI

Sobre a personalização dos componentes de UI

Você pode personalizar a aparência dos componentes em seus aplicativos modificando um ou ambos os elementos a seguir:

Estilos Cada componente tem um conjunto de estilos que você pode definir para especificar os valores usados pelo Flash ao processar a aparência do componente. Os estilos geralmente especificam as capas e os ícones a serem usados em um componente em seus diferentes estados, e também os valores de formatação de texto e preenchimento.

Capas Uma *capa* consiste na coleção de símbolos que compõem a aparência gráfica do componente em determinado estado. Enquanto um estilo especifica a capa a ser utilizada, uma capa é um elemento gráfico usado pelo Flash para desenhar o componente. A *atribuição de capa* é o processo de alteração da aparência de um componente por meio da modificação ou substituição de seus gráficos.

Nota: A aparência padrão dos componentes do ActionScript 3.0 poderia ser considerada como um tema (Aeon Halo), mas essas capas são incorporadas aos componentes. Os componentes do ActionScript 3.0 não oferecem suporte aos arquivos de tema externos que tinham suporte no ActionScript 2.0.

Definição de estilos

Os estilos de um componente geralmente especificam os valores de suas capas, ícones, formatação de texto e preenchimento quando o Flash desenha o componente em seus diversos estados. Por exemplo, o Flash desenha um Button com uma capa diferente para mostrar seu estado pressionado, que ocorre quando você clica com o mouse sobre ele, em relação à capa usada para mostrar o estado ativo ou normal. Ele usa também uma capa diferente no estado desativado, causado pela definição da propriedade `enabled` como `false`.

Você pode definir estilos para componentes em nível de documento, classe e ocorrência. Além disso, algumas propriedades de estilo podem ser herdadas de um componente pai. Por exemplo, o componente List herda estilos de ScrollBar ao herdar de BaseScrollPane.

Você pode definir estilos para personalizar um componente das seguintes formas:

- Definir estilos em uma ocorrência de componente. Você pode alterar as propriedades de cor e de texto de uma única ocorrência de componente. Isso é eficaz em algumas situações, mas pode ser demorado se você precisar definir propriedades individuais para todos os componentes de um documento.
- Definir estilos para todos os componentes de um determinado tipo em um documento. Se quiser aplicar uma aparência consistente a todos os componentes de determinado tipo — por exemplo, a todas as CheckBoxes ou a todos os Buttons em um documento — você pode definir estilos em nível de componente.

Os valores das propriedades de estilo definidas em contêineres são herdadas pelos componentes contidos.

O Flash não exibe as alterações feitas em propriedades de estilo quando você visualiza componentes no Palco usando o recurso de Visualização ao vivo.

Compreensão das configurações de estilo

Aqui estão alguns pontos fundamentais do uso de estilos:

Herança É definido que um componente filho herde um estilo do componente pai por design. Você não pode definir a herança para os estilos no ActionScript.

Precedência Se o estilo de um componente é definido de mais de uma forma, o Flash usa o primeiro estilo encontrado, de acordo com sua ordem de precedência. O Flash procura estilos na seguinte ordem, até que um valor seja encontrado:

- 1 O Flash procura uma propriedade de estilo na ocorrência do componente.
- 2 Se o estilo for um dos estilos herdados, o Flash percorrerá a hierarquia pai em busca de um valor herdado.
- 3 O Flash procura o estilo no componente.
- 4 O Flash procura uma configuração global no StyleManager.
- 5 Se, ainda assim, a propriedade não for definida, terá o valor `undefined`.

Acesso aos estilos padrão de um componente

Você pode acessar os estilos padrão de um componente usando o método `getStyleDefinition()` estático na classe do componente. Por exemplo, o código a seguir recupera os estilos padrão do componente `ComboBox` e exibe os valores padrão das propriedades `buttonWidth` e `downArrowDownSkin`:

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

Definição e obtenção de estilos em uma ocorrência de componente

Qualquer ocorrência de um componente de UI pode chamar diretamente os métodos `setStyle()` e `getStyle()` para definir ou recuperar um estilo. A sintaxe a seguir define um estilo e um valor para uma ocorrência de componente:

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

Esta sintaxe recupera um estilo para uma ocorrência de componente:

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

Observe que o método `getStyle()` retorna o tipo `Object`, pois pode retornar vários estilos, por ter diferentes tipos de dados. Por exemplo, o código a seguir define o estilo de fonte para uma ocorrência de `TextArea` (`aTa`), e depois a recupera usando o método `getStyle()`. O exemplo projeta o valor retornado em um objeto `TextFormat` para atribuí-lo a uma variável `TextFormat`. Sem a projeção, o compilador emitiria um erro devido à tentativa de forçar a conversão de uma variável `Object` em uma variável `TextFormat`.

```
import flash.text.TextFormat;

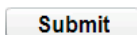
var tf:TextFormat = new TextFormat();
tf.font = "Georgia";
aTa.setStyle("textFormat", tf);
aTa.text = "Hello World!";
var aStyle:TextFormat = aTa.getStyle("textFormat") as TextFormat;
trace(aStyle.font);
```

Usar TextFormat para definir propriedades de texto

Use o objeto TextFormat para formatar texto em uma ocorrência de componente. O objeto TextFormat tem propriedades que permitem especificar características de texto como `bold`, `bullet`, `color`, `font`, `italic`, `size` e muitas outras. Você pode definir essas propriedades no objeto TextFormat, e depois chamar o método `setStyle()` para aplicá-las a uma ocorrência de componente. Por exemplo, o código a seguir define as propriedades `font`, `size` e `bold` de um objeto TextFormat e as aplica a uma ocorrência de 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);
```

A ilustração a seguir mostra o efeito dessas configurações em um botão com o rótulo Submit:



As propriedades de estilo definidas em uma ocorrência de componente com `setStyle()` têm a maior prioridade e substituem todas as demais configurações de estilo. No entanto, quanto mais propriedades você definir usando `setStyle()` em uma única ocorrência de componente, mais lento será o processamento do componente em tempo de execução.

Definição de um estilo para todas as ocorrências de um componente

Você pode definir um estilo para todas as ocorrências de uma classe de componentes usando o método `setComponentStyle()` estático da classe StyleManager. Por exemplo, você pode definir a cor do texto como vermelha em todos os Buttons, arrastando um Button para o Palco e adicionando o seguinte código ActionScript ao painel Ações, no Quadro 1 da Linha do tempo:

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

Todos os Buttons que você adicionar em seguida ao Palco terão rótulos vermelhos.

Definir um estilo para todos os componentes

Você pode definir um estilo para todos os componentes usando o método `setStyle()` estático da classe StyleManager.

- 1 Arraste um componente List para o Palco e atribua a ele o nome de ocorrência de **aList**.
- 2 Arraste um componente Button para o Palco e atribua a ele o nome de ocorrência de **aButton**.
- 3 Pressione **F9** ou selecione Ações, no menu Janela, para abrir o painel Ações, se ainda não estiver aberto. Digite o seguinte código no Quadro 1 da Linha do tempo para definir a cor do texto como vermelha para todos os componentes:

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

- 4 Adicione o código a seguir ao painel Ações para preencher List com texto.

```
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 Selecione Controlar > Testar filme ou pressione Ctrl+Enter para compilar o código e testar seu conteúdo. O texto no rótulo do botão e na lista deverá estar vermelho.

Sobre capas

A aparência de um componente é constituída por elementos gráficos como um contorno, uma cor de preenchimento, ícones e até mesmo outros componentes. Uma ComboBox, por exemplo, contém um componente List, e um componente List contém uma ScrollBar. Juntos, os elementos gráficos compõem a aparência da ComboBox. No entanto, a aparência de um componente muda com base em seu estado atual. Por exemplo, uma CheckBox, sem seu rótulo, tem uma aparência como esta quando aparece em seu aplicativo:



Uma CheckBox em seu estado ativo normal

Se você clicar e mantiver o botão do mouse pressionado sobre a CheckBox, sua aparência será alterada para isto:



Uma CheckBox em seu estado pressionado

Quando você solta o botão do mouse, a CheckBox retorna à sua aparência original, mas agora tem uma marca de seleção para mostrar que foi selecionada.



Uma CheckBox em seu estado selecionado

Os ícones que representam o componente em seus diversos estados são chamados, coletivamente, de *capas*. Você pode alterar a aparência de um componente em qualquer um ou em todos os seus estados editando suas capas no Flash, assim como faria com qualquer outro símbolo do Flash. Você pode acessar as capas de um componente de duas formas. A forma mais simples consiste em arrastar o componente para o Palco e clicar nele duas vezes. Isso abre uma paleta com as capas do componente, que tem a aparência a seguir no caso de uma CheckBox.



As capas de uma CheckBox

Você também pode acessar as capas de um componente individualmente, no painel Biblioteca. Quando você arrasta um componente para o Palco, também o copia para a biblioteca, juntamente com uma pasta com seus ativos e quaisquer outros componentes que ele contenha. Por exemplo, se você arrastar uma ComboBox para o Palco, o painel Biblioteca também irá conter os componentes List, ScrollBar e TextInput, que são incorporados à ComboBox, além de uma pasta de capas para cada um desses componentes e uma pasta Ativos compartilhados, contendo os elementos compartilhados por esses componentes. Você pode editar as capas de qualquer um desses componentes abrindo sua pasta de capas (ComboBoxSkins, ListSkins, ScrollBarSkins ou TextInputSkins) e clicando duas vezes no ícone da capa que deseja editar. Por exemplo, clicar duas vezes em ComboBox_downSkin abre a capa no modo de edição de símbolo, conforme mostrado na ilustração a seguir:



A ComboBox_downSkin

Criar uma nova capa

Se quiser criar uma nova aparência para um componente no seu documento, edite as capas do componente para alterar sua aparência. Para acessar as capas de um componente, basta clicar nele duas vezes no Palco para abrir uma paleta com suas capas. Depois, clique duas vezes na capa que deseja editar para abri-la no modo de edição de símbolo. Por exemplo, clique duas vezes no componente TextArea no Palco para abrir seus ativos no modo de edição de símbolo. Defina o controle de zoom como 400%, ou mais se preferir, e edite o símbolo para alterar sua aparência. Quando você terminar, a alteração afetará todas as ocorrências do componente no documento. Como alternativa, você pode clicar duas vezes em uma capa específica no painel Biblioteca para abri-la no Palco no modo de edição de símbolo.

Você pode modificar capas de componentes das seguintes formas:

- Criar uma nova capa para todas as ocorrências
- Criar novas capas para algumas ocorrências

Criar uma capa para todas as ocorrências

Quando você edita a capa de um componente, por padrão, altera a aparência do componente em todas as suas ocorrências no documento. Se quiser criar aparências diferentes para o mesmo componente, você deverá duplicar as capas que deseja alterar e atribuir-lhes nomes diferentes, editá-las e depois definir os estilos adequados para aplicar a elas. Para obter mais informações, consulte “[Criar capas para algumas ocorrências](#)” na página 106.

Este capítulo descreve como alterar uma ou mais capas para cada um dos componentes de UI. Se você seguir um destes procedimentos para alterar uma ou mais capas de um componente de UI, irá alterar todas as ocorrências no documento.

Criar capas para algumas ocorrências

Você pode criar uma capa para algumas ocorrências de um componente usando o seguinte procedimento geral:

- Selecione a capa na pasta Ativos do componente, no painel Biblioteca.
- Duplique a capa e atribua a ela um nome de classe exclusivo.
- Edite a capa para conferir a ela a aparência desejada.
- Chame o método `setStyle()` da ocorrência do componente para atribuir a nova capa ao estilo de capas.

O procedimento a seguir cria uma nova `selectedDownSkin` para uma de duas ocorrências de `Button`.

- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste dois `Buttons` do painel Componentes para o Palco e atribua a eles os nomes de ocorrência de **aButton** e **bButton**.
- 3 Abra o painel Biblioteca, e depois as pastas `Component Assets` e `ButtonSkins` dentro dele.
- 4 Clique na capa `selectedDownSkin` para selecioná-la.
- 5 Clique com o botão direito para abrir o menu de contexto e selecione `Duplicar`.
- 6 Na caixa de diálogo `Duplicar símbolo`, atribua à nova capa um nome exclusivo, por exemplo **Button_mySelectedDownSkin**. Em seguida, clique em `OK`.
- 7 Na pasta `Biblioteca > Component Assets > ButtonSkins`, selecione `Button_mySelectedDownSkin` e clique com o botão direito para abrir o menu de contexto. Selecione `Ligação` para abrir a caixa de diálogo `Propriedades da ligação`.
- 8 Clique na caixa de seleção `Exportar para ActionScript`. Deixe a caixa de seleção `Exportar` no primeiro quadro selecionada e verifique se o nome da classe é exclusivo. Clique em `OK`, e depois clique em `OK` novamente em resposta ao aviso informando que não foi possível encontrar uma definição de classe e que uma será criada.
- 9 Clique duas vezes na capa `Button_mySelectedDownSkin`, no painel Biblioteca, para abri-la no modo de edição de símbolo.
- 10 Clique no preenchimento azul no centro da capa até que a cor apareça no Seletor de cor de preenchimento, no Inspetor de propriedades. Clique no seletor de cor e selecione a cor `#00CC00` para o preenchimento da capa.
- 11 Clique no botão `Voltar`, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 12 No Inspetor de propriedades, clique na guia `Parâmetros` de cada botão e defina o parâmetro de alternância como `true`.
- 13 Adicione o seguinte código ao painel `Ações` no Quadro 1 da Linha do tempo:

```
bButton.setStyle("selectedDownSkin", Button_mySelectedDownSkin);
bButton.setStyle("downSkin", Button_mySelectedDownSkin);
```
- 14 Selecione `Controlar > Testar filme`.

15 Clique em cada botão. Observe que a capa pressionada (selecionada e não selecionada) do objeto `bButton` usa o símbolo da nova capa.

Personalizar o componente Button

Você pode transformar um componente `Button` horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou qualquer propriedade aplicável da classe `Button`, como `height`, `width`, `scaleX` e `scaleY`.

Redimensionar o botão não altera o tamanho do ícone ou do rótulo. A caixa delimitadora de um `Button` corresponde à sua borda e também designa a área de pressionamento da ocorrência. Se você aumentar o tamanho da ocorrência, também aumentará o tamanho da área de pressionamento. Se a caixa delimitadora for muito pequena para ajustar-se ao rótulo, o rótulo será recortado para ajustar-se.

Se o `Button` tiver um ícone maior do que ele, o ícone se estenderá para além das bordas do `Button`.

Usar estilos com o componente Button

Os estilos de um `Button` geralmente especificam os valores de suas capas, ícones, formatação de texto e preenchimento quando o componente é desenhado em seus diversos estados.

O procedimento a seguir coloca dois `Buttons` no Palco e define a propriedade `emphasized` como `true` para os dois `Buttons` quando o usuário clica em um deles. Também define o estilo `emphasizedSkin` do segundo `Button` como o estilo `selectedOverSkin` quando o usuário clica nele, de forma que os dois `Buttons` exibam capas diferentes para o mesmo estado.

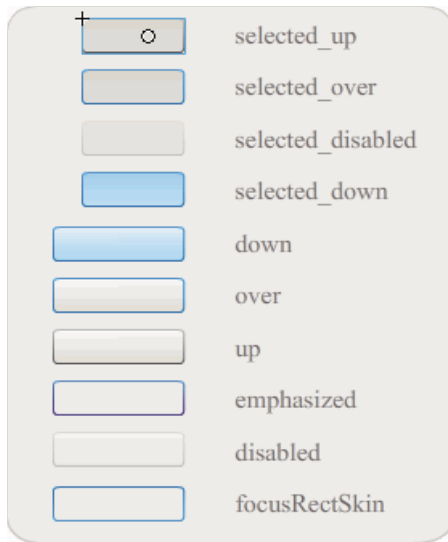
- 1 Crie um arquivo Flash (ActionScript 3.0).
- 2 Arraste dois `Buttons` para o Palco, um por vez, e atribua a eles os nomes de ocorrência de `aBtn` e `bBtn`. Na guia Parâmetros do Inspetor de propriedades, atribua a eles os rótulos `Button A` e `Button B`.
- 3 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo:

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

- 4 Selecione Controlar > Testar filme.
- 5 Clique em um dos botões para ver o efeito do estilo `emphasizedSkin` em cada um deles.

Usar capas com o componente Button

O componente `Button` usa as capas a seguir, que correspondem a seus diferentes estados. Para editar uma ou mais capas, a fim de alterar a aparência de `Button`, clique duas vezes na ocorrência de `Button` no Palco para abrir uma paleta com suas capas, conforme mostrado na ilustração a seguir:

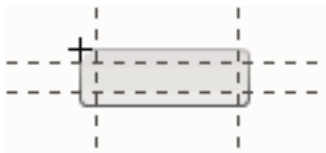


Capas de Button

Se um botão está ativado, exibe o estado de mouse acima quando o ponteiro se move sobre ele. O botão recebe foco de entrada e exibe seu estado pressionado quando é pressionado. O botão retorna ao seu estado de mouse acima quando o botão do mouse é solto. Se o ponteiro é movido para fora do botão enquanto o mouse é pressionado, o botão retorna ao estado original. Se o parâmetro de alternância estiver definido como `true`, o estado pressionado será mostrado com a `selectedDownSkin`, o estado ativo, com a `selectedUpSkin` e o estado de mouse acima, com a `selectedOverSkin`.

Se um Button está desativado, exibe seu estado desativado, independentemente da interação do usuário.

Para editar uma das capas, clique nela duas vezes para abri-la no modo de edição de símbolo, conforme mostrado na ilustração a seguir:



Button em modo de edição de símbolo

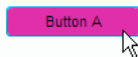
Neste ponto, você pode usar as ferramentas de autoria do Flash para editar a capa como desejar.

O procedimento a seguir altera a cor da capa `selected_over` de Button.

- 1 Crie um novo arquivo Flash (ActionScript 3.0).
- 2 Arraste um Button do painel Componentes para o Palco. Na guia Parâmetros, defina o parâmetro de alternância como `true`.
- 3 Clique duas vezes no Button para abrir a paleta com suas capas.
- 4 Clique duas vezes na capa `selected_over` para abri-la no modo de edição de símbolo.
- 5 Defina o controle de zoom como 400% para ampliar o ícone para edição.
- 6 Clique duas vezes no plano de fundo até que sua cor apareça no Seletor de cor de preenchimento, no Inspetor de propriedades.
- 7 Selecione a cor `#CC0099` no Seletor de cor de preenchimento para aplicá-la ao plano de fundo da capa `selected_over`.

- 8 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 9 Selecione Controlar > Testar filme.
- 10 Clique no botão para colocá-lo no estado selecionado.

Quando você mover o ponteiro do mouse sobre o Button, o estado `selected_over` deverá aparecer como na ilustração a seguir.



Button exibindo a capa `selected_over` com a cor modificada

Personalizar o componente CheckBox

Você pode transformar um componente CheckBox horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou as propriedades aplicáveis da classe CheckBox. Por exemplo, você pode alterar o tamanho de uma CheckBox definindo suas propriedades `height`, `width`, `scaleX` e `scaleY`. Redimensionar a CheckBox não altera o tamanho do rótulo ou do ícone da caixa de seleção, mas somente o tamanho da caixa delimitadora.

A caixa delimitadora de uma ocorrência de CheckBox é invisível e também designa a área de pressionamento da ocorrência. Se você aumentar o tamanho da ocorrência, também aumentará o tamanho da área de pressionamento. Se a caixa delimitadora for muito pequena para ajustar-se ao rótulo, o rótulo será recortado para ajustar-se.

Usar estilos com a CheckBox

Você pode definir propriedades de estilo para alterar a aparência de uma ocorrência de CheckBox. Por exemplo, o procedimento a seguir altera o tamanho e a cor do rótulo de uma CheckBox.

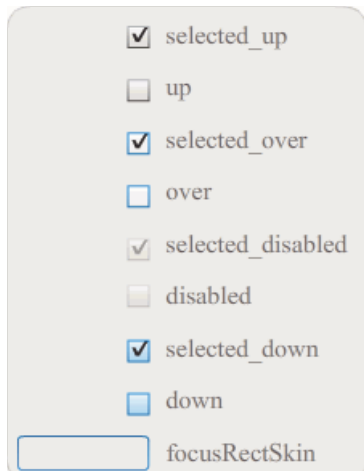
- 1 Arraste o componente CheckBox do painel Componentes para o Palco e atribua a ele um nome de ocorrência de **myCb**.
- 2 Clique na guia Parâmetros, no Inspetor de propriedades, e digite o seguinte valor para o parâmetro do rótulo: **Less than \$500?**
- 3 No Quadro 1 da Linha do tempo principal, digite o código a seguir no painel Ações:

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

Para obter mais informações, consulte “[Definição de estilos](#)” na página 101. Para obter informações sobre como definir propriedades de estilo para alterar os ícones e as capas do componente, consulte “[Criar uma nova capa](#)” na página 105 e “[Usar capas com a CheckBox](#)” na página 109.

Usar capas com a CheckBox

O componente CheckBox tem as seguintes capas, cuja aparência você pode alterar editando-as.



Capas de CheckBox

Este exemplo altera a cor do contorno e a cor do plano de fundo do componente em seus estados `up` e `selectedUp`. Seria preciso seguir etapas semelhantes para alterar as capas de outros estados.

- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste o componente CheckBox para o Palco, o que também o colocará na biblioteca com uma pasta contendo seus ativos.
- 3 Clique duas vezes no componente CheckBox no Palco para abrir seu painel de ícones de capa.
- 4 Clique duas vezes no ícone `selected_up` para abri-lo no modo de edição de símbolo.
- 5 Defina o controle de zoom como 800% para ampliar o ícone para edição.
- 6 Clique na borda da CheckBox para selecioná-la. Use o Seletor de cor de preenchimento, no Inspetor de propriedades, para selecionar a cor #0033FF e aplicá-la à borda.
- 7 Clique duas vezes no plano de fundo da CheckBox para selecioná-la e, mais uma vez, use o Seletor de cor de preenchimento para definir a cor do fundo como #00CCFF.
- 8 Repita as etapas de 4 a 8 para a capa ativa da CheckBox.
- 9 Selecione Controlar > Testar filme.

Personalizar o componente ColorPicker

O único redimensionamento possível em um `ColorPicker` é por meio de seus estilos: `swatchWidth`, `swatchHeight`, `backgroundPadding`, `textFieldWidth` e `textFieldHeight`. Se você tentar alterar o tamanho do `ColorPicker` com a ferramenta Transformar ou com o ActionScript usando o método `setSize()`, ou então as propriedades `width`, `height`, `scaleX` ou `scaleY`, esses valores serão ignorados quando você criar o arquivo SWF e o `ColorPicker` será exibido em seu tamanho padrão. O plano de fundo da paleta será redimensionado para ajustar-se ao número de colunas definido usando `setStyle()` para o estilo `columnCount`. O número de colunas padrão é 18. Você pode definir as cores personalizadas como 1024, e a paleta será redimensionada verticalmente para ajustar-se ao número de amostras.

Usar estilos com o componente ColorPicker

Você pode definir vários estilos para alterar a aparência do componente ColorPicker. O procedimento a seguir, por exemplo, altera o número de colunas (`columnCount`) no ColorPicker para 12, altera a altura (`swatchHeight`) e a largura (`swatchWidth`) das amostras de cor e altera o preenchimento do campo de texto (`textPadding`) e do plano de fundo (`backgroundPadding`).

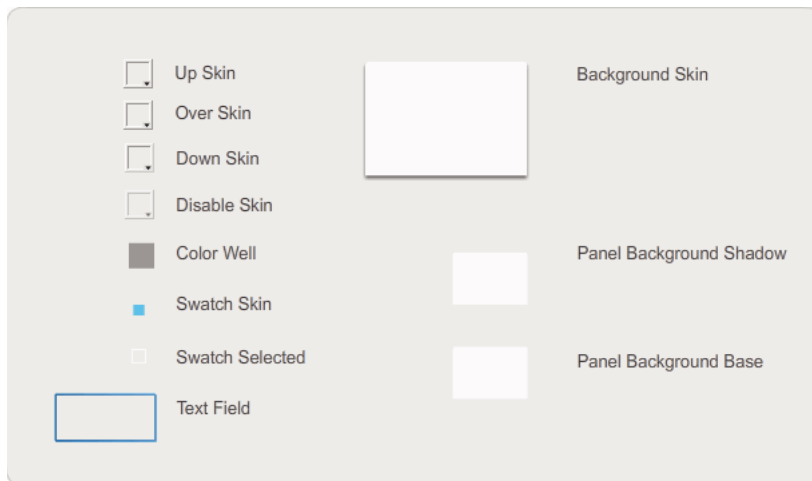
- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste o componente ColorPicker para o Palco e atribua a ele o nome de ocorrência de **aCp**.
- 3 Abra o painel Ações, selecione o Quadro 1 na Linha do tempo principal e digite o seguinte código:

```
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 Selecione Controlar > Testar filme.
- 5 Clique no ColorPicker para abri-lo e ver como as configurações alteraram sua aparência.

Usar capas com o componente ColorPicker

O componente ColorPicker usa as capas a seguir para representar seus estados visuais.



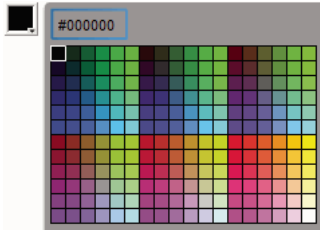
Capas de ColorPicker

Você pode alterar a cor da capa Background para alterar a cor do plano de fundo da paleta.

- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste o componente ColorPicker para o Palco.
- 3 Clique nele duas vezes para abrir sua paleta de capas.
- 4 Clique duas vezes na capa Background até que ela seja selecionada e que o Seletor de cor de preenchimento apareça no Inspetor de propriedades.
- 5 Selecione a cor #999999 usando o Seletor de cor de preenchimento para aplicá-la à capa Background.

- 6 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 7 Selecione Controlar > Testar filme.

Quando você clicar no ColorPicker, o plano de fundo da paleta deverá estar cinza, conforme mostrado na ilustração a seguir.



ColorPicker com capa Background em cinza escuro

Personalizar o componente ComboBox

Você pode transformar um componente ComboBox horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou as propriedades aplicáveis da classe `ComboBox`, como `height`, `width`, `scaleX` e `scaleY`.

A ComboBox será redimensionada para ajustar-se à largura e à altura especificadas. A lista será redimensionada para ajustar-se à largura do componente, a menos que a propriedade `dropdownWidth` tenha sido definida.

Se o texto for grande demais para ajustar-se à ComboBox, será recortado. Você precisa redimensionar a ComboBox e definir a propriedade `dropdownWidth` para ajustar-se ao texto.

Usar estilos com o componente ComboBox

Você pode definir propriedades de estilo para alterar a aparência de um componente ComboBox. Os estilos especificam valores para as capas, o renderizador de célula, o preenchimento e a largura de botão do componente. O exemplo a seguir define os estilos `buttonWidth` e `textPadding`. O estilo `buttonWidth` define a largura da área de pressionamento do botão e está em vigor quando a ComboBox é editável e você só pode pressionar o botão para abrir a lista suspensa. O estilo `textPadding` especifica a quantidade de espaço entre a borda externa do campo de texto e o texto. É útil para centralizar o texto verticalmente no campo de texto se você tornar a ComboBox mais alta. Caso contrário, pode parecer que o texto está na parte superior do campo de texto.

- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste o componente ComboBox para o Palco e atribua a ele o nome de ocorrência de **aCb**.
- 3 Abra o painel Ações, selecione o Quadro 1 na Linha do tempo principal e digite o seguinte código:

```
import fl.data.DataProvider;

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

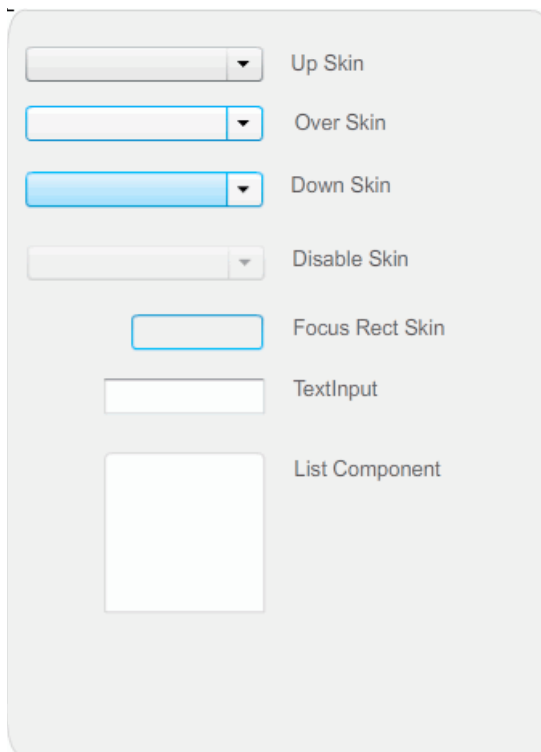
var items:Array = [
    {label:"San Francisco", data:"601 Townsend St."},
    {label:"San Jose", data:"345 Park Ave."},
    {label:"San Diego", data:"10590 West Ocean Air Drive, Suite 100"},
    {label:"Santa Rosa", data:"2235 Mercury Way, Suite 105"},
    {label:"San Luis Obispo", data:"3220 South Higuera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

4 Selecione Controlar > Testar filme.

Observe que a área do botão na qual você pode clicar para abrir a lista suspensa é apenas uma área estreita à direita. Observe também que o texto está centralizado verticalmente no campo de texto. Você pode tentar executar o exemplo sem as duas instruções `setStyle()` para ver o efeito.

Usar capas com a ComboBox

A ComboBox usa as capas a seguir para representar seus estados visuais:



Capas de ComboBox

Você pode alterar a cor da capa Up para alterar a cor do componente em seu estado inativo no Palco.

1 Crie um novo documento de arquivo Flash (ActionScript 3.0).

- 2 Arraste o componente ComboBox para o Palco.
- 3 Clique nele duas vezes para abrir sua paleta de capas.
- 4 Clique duas vezes na capa Up até que ela seja selecionada e aberta para edição.
- 5 Defina o controle de zoom como 400%.
- 6 Clique na área central da capa até que sua cor apareça no Seletor de cor de preenchimento, no Inspetor de propriedades.
- 7 Selecione a cor #33FF99 usando o Seletor de cor de preenchimento para aplicá-la à capa Up.
- 8 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 9 Selecione Controlar > Testar filme.

A ComboBox deverá aparecer no Palco conforme mostrada na ilustração a seguir.



ComboBox com cor personalizada para capa de plano de fundo

Personalizar o componente DataGrid

Você pode transformar um componente DataGrid horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou as propriedades aplicáveis, como `width`, `height`, `scaleX` e `scaleY`. Se não houver barra de rolagem horizontal, a largura das colunas será ajustada proporcionalmente. Se ocorrer o ajuste de colunas (e conseqüentemente de células), o texto nas células poderá ser recortado.

Usar estilos com o componente DataGrid

Você pode definir propriedades de estilo para alterar a aparência de um componente DataGrid. O componente DataGrid herda estilos do componente List (consulte “[Usar estilos com o componente List](#)” na página 120).

Definir estilos para uma coluna individual

Um objeto DataGrid pode ter várias colunas, e você pode especificar diferentes renderizadores de célula para cada coluna. Cada coluna de uma DataGrid é representada por um objeto DataGridColumn, e a classe DataGridColumn inclui uma propriedade `cellRenderer`, para a qual você pode definir o CellRenderer da coluna.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente DataGrid para o painel Biblioteca.
- 3 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo. Este código cria uma DataGrid com uma longa seqüência de caracteres de texto na terceira coluna. No final, define a propriedade `cellRenderer` da coluna com o nome de um renderizador de célula que processa uma célula de várias linhas.

```
/* 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 Salve o arquivo FLA como MultiLineGrid.fla.
- 5 Crie um novo arquivo ActionScript.
- 6 Copie o código ActionScript a seguir na janela Script:

```
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 Salve o arquivo ActionScript como MultiLineCell.as na mesma pasta onde você salvou MultiLineGrid.fla.

8 Volte ao aplicativo MultiLineGrid.fla e selecione Controlar > Testar filme.

A DataGrid deve ter esta aparência:

| firstName | lastName | note | item |
|-----------|----------|---|------|
| Winston | Elstad | An example of a cell renderer class that displays a multiple line TextField | 100 |
| Ric | Dietrich | An example of a cell renderer class that displays a multiple line TextField | 101 |
| Ewing | Canepa | An example of a cell renderer class that displays a multiple line TextField | 102 |
| Kevin | Wade | An example of a cell renderer class that displays a multiple line TextField | 103 |

DataGrid para o aplicativo MultiLineGrid.fla

Definir estilos de cabeçalho

Você pode definir o estilo de texto de uma linha de cabeçalho usando o estilo `headerTextFormat`. O exemplo a seguir usa o objeto `TextFormat` para definir que o estilo `headerTextFormat` use a fonte Arial, a cor vermelha e um tamanho de fonte 14, em itálico.

- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste o componente DataGrid para o Palco e atribua a ele o nome de ocorrência de **aDg**.
- 3 Abra o painel Ações, selecione o Quadro 1 na Linha do tempo principal e digite o seguinte código:


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

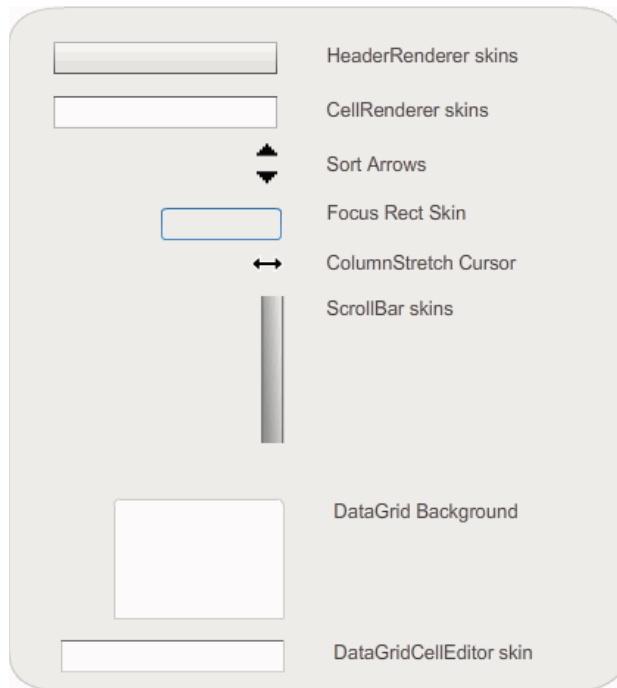
var myDP:Array = new Array();
myDP = [{FirstName:"Winston", LastName:"Elstad"},
        {FirstName:"Ric", LastName:"Dietrich"},
        {FirstName:"Ewing", LastName:"Canepa"},
        {FirstName:"Kevin", LastName:"Wade"},
        {FirstName:"Kimberly", LastName:"Dietrich"},
        {FirstName:"AJ", LastName:"Bilow"},
        {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 Seleccione Controlar > Testar filme para executar o aplicativo.

Usar capas com o componente DataGrid

O componente DataGrid usa as capas a seguir para representar seus estados visuais:



Capas de DataGrid

A capa CellRenderer é usada para as células do corpo de DataGrid, enquanto a capa HeaderRenderer é usada para a linha do cabeçalho. O procedimento a seguir altera a cor do plano de fundo da linha do cabeçalho, mas você pode usar o mesmo processo para alterar a cor do plano de fundo das células do corpo de DataGrid, editando a capa CellRenderer.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente DataGrid para o Palco e atribua a ele o nome de ocorrência de **adg**.
- 3 Clique duas vezes no componente para abrir sua paleta de capas.
- 4 Defina o controle de zoom como 400% para ampliar os ícones para edição.
- 5 Clique duas vezes na capa HeaderRenderer para abrir a paleta com as capas de HeaderRenderer.
- 6 Clique duas vezes na Up_Skin para abri-la no modo de edição de símbolo. Clique em seu plano de fundo até que seja selecionado e o Seletor de cor de preenchimento apareça no Inspetor de propriedades.
- 7 Selecione a cor #00CC00 usando o Seletor de cor de preenchimento para aplicá-la ao plano de fundo da capa Up_Skin de HeaderRenderer.
- 8 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 9 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo, para adicionar dados à DataGrid:

```
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 Selecione Controlar > Testar filme para testar o aplicativo.

A DataGrid deverá aparecer como na ilustração a seguir, com o plano de fundo da linha do cabeçalho em verde.

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

DataGrid com plano de fundo da linha do cabeçalho personalizado

Personalizar o componente Label

Você pode transformar um componente Label horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Você também pode definir o parâmetro de autoria `autoSize`; a definição desse parâmetro não altera a caixa delimitadora na visualização ao vivo, mas o Label é redimensionado. O Label é redimensionado de acordo com o parâmetro `wordwrap`. Se o parâmetro for `true`, o Label será redimensionado verticalmente para ajustar-se ao texto. Se o parâmetro for `false`, o Label será redimensionado horizontalmente. Em tempo de execução, use o método `setSize()`. Para obter mais informações consulte o método `Label.setSize()` e a propriedade `Label.autoSize` na *Referência do ActionScript 3.0 para o Flash Professional*. Consulte também “[Criar um aplicativo com o componente Label](#)” na página 65.

Usar estilos com o componente Label

Você pode definir propriedades de estilo para alterar a aparência de uma ocorrência de Label. Todo o texto em uma ocorrência do componente Label deve compartilhar o mesmo estilo. O componente Label tem um estilo `textFormat`, que tem os mesmos atributos do objeto `TextFormat` e permite definir as mesmas propriedades para o conteúdo `Label.text` que seriam definidas para um `TextField` regular do Flash. O exemplo a seguir define a cor do texto em um rótulo como vermelha.

1 Arraste o componente Label do painel Componentes para o Palco e atribua a ele um nome de ocorrência de `a_label`.

2 Clique na guia Parâmetros e substitua o valor da propriedade de texto pelo texto:

Color me red

3 Selecione o Quadro 1 na Linha do tempo principal, abra o painel Ações e digite o seguinte código:

```
/* 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 Selecione Controlar > Testar filme.

Para obter mais informações sobre os estilos de Label, consulte a classe Classe Label na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

As capas e o Label

O componente Label não tem elementos visuais de capa.

Personalizar o componente List

Você pode transformar um componente List horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` e as propriedades aplicáveis da classe `List`, como `height`, `width`, `scaleX` e `scaleY`.

Quando uma lista é redimensionada, suas linhas diminuem horizontalmente, recortando o texto contido nelas. Verticalmente, a lista adiciona ou remove linhas conforme necessário. As barras de rolagem são posicionadas automaticamente conforme necessário.

Usar estilos com o componente List

Você pode definir propriedades de estilo para alterar a aparência de um componente List. Os estilos especificam valores para as capas e o preenchimento do componente quando ele é desenhado.

Os diversos estilos de capas permitem especificar diferentes classes a serem usadas com a capa. Para obter mais informações sobre o uso de estilos de capa, consulte “[Sobre capas](#)” na página 104.

O procedimento a seguir define o valor do estilo `contentPadding` para o componente `List`. Observe que o valor dessa configuração é subtraído do tamanho da `List` para obter o preenchimento em torno do conteúdo. Então, talvez seja preciso aumentar o tamanho da `List` para impedir que o texto contido nela seja recortado.

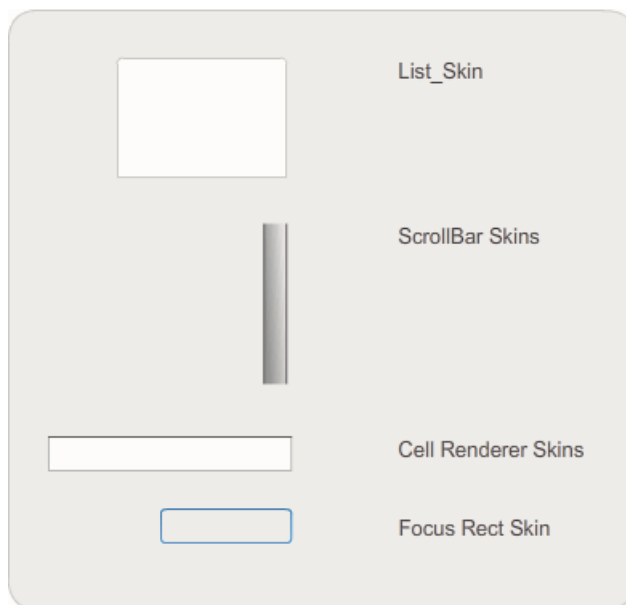
- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste o componente `List` do painel Componentes para o Palco e atribua a ele um nome de ocorrência de `aList`.
- 3 Selecione o Quadro 1 na Linha do tempo principal, abra o painel Ações e digite o código a seguir, que define o estilo `contentPadding` e adiciona dados à Lista:

```
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 Selecione Controlar > Testar filme.

Usar capas com o componente List

O componente `List` usa as capas a seguir para representar seus estados visuais:

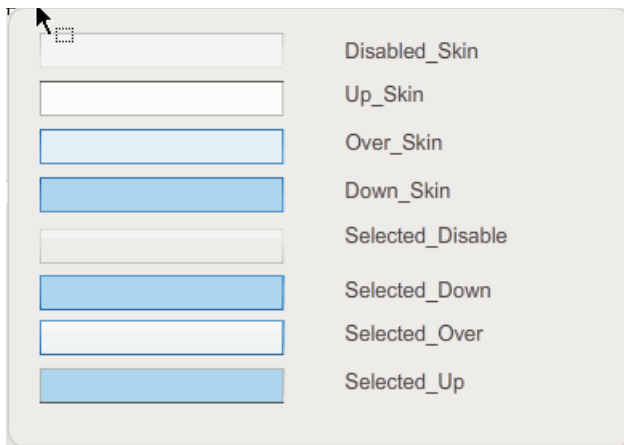


Capas de List

Para obter mais informações sobre a atribuição de capa à `ScrollBar`, consulte “[Personalizar o componente UI ScrollBar](#)” na página 136. Para obter mais informações sobre a atribuição da capa `Focus Rect`, consulte “[Personalizar o componente TextArea](#)” na página 130

Nota: Alterar a capa da `ScrollBar` em um componente causará a mesma alteração em todos os outros componentes que usam a `ScrollBar`.

Clique duas vezes na capa `Cell Renderer` para abrir uma segunda paleta de capas para os diferentes estados de uma célula de `List`.



Capas Cell Renderer de List

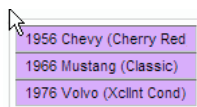
Você pode alterar a aparência das células de List editando essas capas. O procedimento a seguir altera a cor da capa Up para alterar a aparência de List em seu estado inativo normal.

- 1 Crie um novo documento de arquivo Flash (ActionScript 3.0).
- 2 Arraste o componente List do painel Componentes para o Palco e atribua a ele um nome de ocorrência de **aList**.
- 3 Clique duas vezes na List para abrir sua paleta de capas.
- 4 Clique duas vezes na capa Cell Renderer para abrir a paleta com as capas Cell Renderer.
- 5 Clique duas vezes na capa Up_Skin para abri-la para edição.
- 6 Clique na área de preenchimento da capa para selecioná-la. Um Seletor de cor de preenchimento aparecerá no Inspetor de propriedades com a atual cor de preenchimento da capa.
- 7 Selecione a cor #CC66FF usando o Seletor de cor de preenchimento para aplicá-la ao preenchimento da capa Up_Skin.
- 8 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 9 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo, para adicionar dados à 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 Selecione Controlar > Testar filme.

A List deverá ser exibida como na ilustração a seguir:



Células de List com a cor de Up_Skin personalizada

Os quadros resultam da definição do estilo `contentPadding`.

Personalizar o componente NumericStepper

Você pode transformar um componente NumericStepper horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou qualquer propriedade ou método aplicável da classe NumericStepper, como `width`, `height`, `scaleX` e `scaleY`.

Redimensionar o componente NumericStepper não altera a largura dos botões de seta para cima e para baixo. Se o seletor é redimensionado de forma a tornar-se maior do que a altura padrão, o comportamento padrão inclui os botões de seta nas partes superior e inferior do componente. Caso contrário, o dimensionamento de 9 fatias determina como os botões são desenhados. Os botões de seta sempre aparecem à direita da caixa de texto.

Os estilos e o componente NumericStepper

Você pode definir as propriedades de estilo do componente NumericStepper para alterar sua aparência. Os estilos especificam valores para as capas, o preenchimento e o formato de texto do componente quando ele é desenhado. O estilo `textFormat` permite alterar o tamanho e a aparência do valor do NumericStepper. Os diversos estilos de capas permitem especificar diferentes classes a serem usadas com a capa. Para obter mais informações sobre o uso de estilos de capa, consulte [“Sobre capas”](#) na página 104.

Este procedimento usa o estilo `textFormat` para alterar a aparência do valor exibido por NumericStepper.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente NumericStepper do painel de componentes para o Palco e atribua a ele um nome de ocorrência de `myNs`.
- 3 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo principal:

```
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 Selecione Controlar > Testar filme.

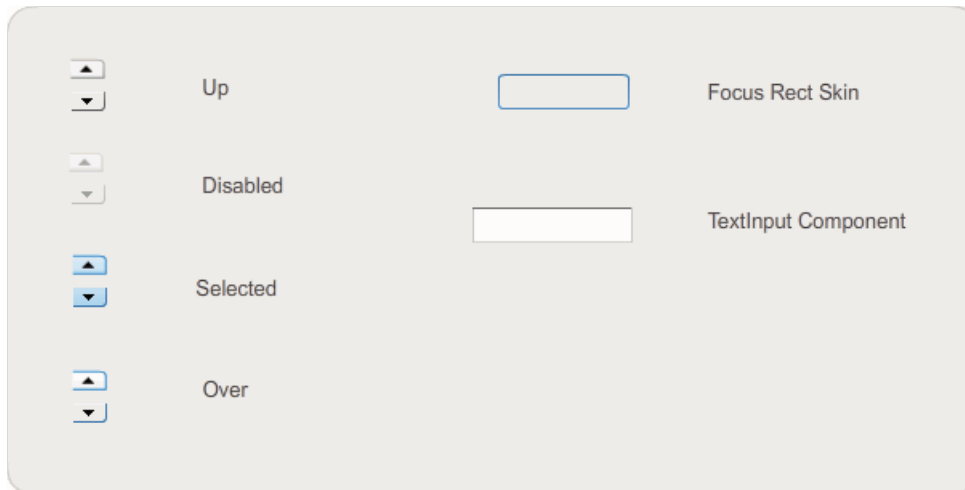
As capas e o componente NumericStepper

O componente NumericStepper tem capas que representam os estados ativo, pressionado, desativado e selecionado de seus botões.

Se um seletor está ativado, os botões para cima e para baixo exibem seus estados de mouse acima quando o ponteiro se move sobre eles. Os botões exibem seu estado pressionado quando são pressionados. Os botões retornam ao estado de mouse acima quando o botão do mouse é solto. Se o ponteiro é movido para fora dos botões enquanto o mouse é pressionado, os botões retornam ao estado original.

Se um seletor está desativado, exibe seu estado desativado, independentemente da interação do usuário.

Um componente NumericStepper tem as seguintes capas:



Capas de NumericStepper

- 1 Crie um novo arquivo FLA.
- 2 Arraste o componente NumericStepper para o Palco.
- 3 Defina o controle de zoom como 400% para ampliar a imagem para edição.
- 4 Clique duas vezes no plano de fundo da capa TextInput, no painel de capas, até que você alcance o nível de Grupo e a cor do plano de fundo apareça no Seletor de cor de preenchimento, no Inspetor de propriedades.
- 5 Usando o Seletor de cor de preenchimento, no Inspetor de propriedades, selecione a cor #9999FF para aplicá-la ao plano de fundo da capa TextInput.
- 6 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 7 Clique duas vezes no NumericStepper novamente para reabrir o painel de capas.
- 8 Clique duas vezes no plano de fundo do botão de seta para cima, no grupo Up, até que o plano de fundo seja selecionado e sua cor apareça no Seletor de cor de preenchimento, no Inspetor de propriedades.
- 9 Selecione a cor #9966FF para aplicá-la ao plano de fundo do botão de seta para cima.
- 10 Repita as etapas 8 e 9 para a seta para baixo no grupo Up.
- 11 Selecione Controlar > Testar filme.

A ocorrência de NumericStepper deverá aparecer como mostrada na ilustração a seguir:



Personalizar o componente ProgressBar

Você pode transformar um componente ProgressBar horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou as propriedades adequadas da classe `ProgressBar`, como `height`, `width`, `scaleX` e `scaleY`.

A ProgressBar tem três capas: uma capa de faixa, uma capa de barra e uma capa indeterminada. Ela usa o dimensionamento de 9 fatias para dimensionar os ativos.

Os estilos e o componente ProgressBar

Você pode definir propriedades de estilo para alterar a aparência de uma ocorrência de ProgressBar. Os estilos de ProgressBar especificam valores para sua capa e seu preenchimento quando o componente é desenhado. O exemplo a seguir aumenta o tamanho de uma ocorrência de ProgressBar e define seu estilo barPadding.

- 1 Crie um novo arquivo FLA.
- 2 Arraste o componente ProgressBar do painel Componentes para o Palco e atribua a ele um nome de ocorrência de **myPb**.
- 3 No Quadro 1 da Linha do tempo principal, digite o código a seguir no painel Ações:

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

- 4 Selecione Controlar > Testar filme.

Para obter informações sobre como definir estilos de capas, consulte “[Sobre capas](#)” na página 104.

As capas e o componente ProgressBar

O componente ProgressBar usa capas para representar a faixa da barra de progresso, a barra concluída e uma barra indeterminada, conforme mostrado na ilustração a seguir.



Capas de ProgressBar

A barra é inserida sobre a capa de faixa, usando o barPadding para determinar o posicionamento. Os ativos são dimensionados com uso do dimensionamento de 9 fatias.

A barra indeterminada é usada quando a propriedade `indeterminate` da ocorrência de ProgressBar é definida como `true`. A capa é redimensionada vertical e horizontalmente para ajustar-se ao tamanho da ProgressBar.

Você pode editar essas capas para alterar a aparência da ProgressBar. O exemplo a seguir altera a cor da barra indeterminada.

- 1 Crie um novo arquivo FLA.
- 2 Arraste um componente ProgressBar para o Palco e clique nele duas vezes para abrir seu painel de ícones de capa.
- 3 Clique duas vezes na capa da barra indeterminada.
- 4 Defina o controle de zoom como 400% para ampliar o ícone para edição.
- 5 Clique duas vezes em uma das barras diagonais, mantenha pressionada a tecla Shift e clique em cada uma das outras barras. A cor atual aparecerá no Seletor de cor de preenchimento, no Inspetor de propriedades.

- 6 Clique no Seletor de cor de preenchimento, no Inspetor de propriedades, para abri-lo e selecionar a cor #00CC00, que será aplicada às barras diagonais selecionadas.
- 7 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 8 Selecione Controlar > Testar filme.

A ProgressBar deverá aparecer como é mostrada na ilustração a seguir.



Personalizar o componente RadioButton

Você pode transformar um componente RadioButton horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()`.

A caixa delimitadora de um componente RadioButton é invisível e também designa a área de pressionamento do componente. Se você aumentar o tamanho do componente, também aumentará o tamanho da área de pressionamento.

Se a caixa delimitadora do componente for muito pequena para ajustar-se ao seu rótulo, o rótulo será recortado para ajustar-se.

Usar estilos com o componente RadioButton

Você pode definir propriedades de estilo para alterar a aparência de um RadioButton. As propriedades de estilos de RadioButton especificam valores de capas, ícones, formatação de texto e preenchimento quando o componente é desenhado. Os estilos de RadioButton especificam valores para as capas e o preenchimento do layout quando o componente é desenhado.

O exemplo a seguir recupera o estilo `textFormat` de um componente CheckBox e o aplica a um RadioButton para tornar idêntico o estilo de seus rótulos.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente CheckBox para o Palco e atribua a ele o nome de ocorrência de **myCh** no Inspetor de propriedades.
- 3 Arraste um RadioButton para o Palco e atribua a ele o nome de ocorrência de **myRb** no Inspetor de propriedades.
- 4 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo.

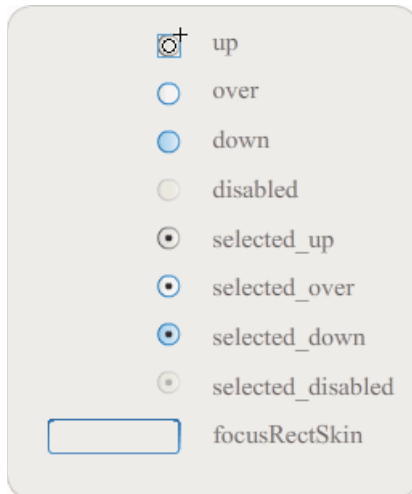
```
var tf:TextFormat = new TextFormat();
tf.color = 0x00FF00;
tf.font = "Georgia";
tf.size = 18;
myCh.setStyle("textFormat", tf);
myRb.setStyle("textFormat", myCh.getStyle("textFormat"));
```

Esse código define o estilo `textFormat` para a CheckBox, e depois o aplica ao RadioButton chamando o método `getStyle()` na CheckBox.

- 5 Selecione Controlar > Testar filme.

As capas e o componente RadioButton

O RadioButton tem as seguintes capas, cuja aparência você pode alterar editando-as:



Capas de RadioButton

Se um RadioButton está ativado e não selecionado, exibe sua capa de mouse acima quando um usuário move o ponteiro sobre ele. Quando um usuário clica em um RadioButton, ele recebe foco de entrada e exibe sua capa `selected_down`. Quando um usuário solta o mouse, o RadioButton exibe a capa `selected_up`. Se um usuário move o ponteiro para fora da área de pressionamento do RadioButton com o botão do mouse pressionado, o RadioButton volta a exibir a capa `up`.

Se um RadioButton está desativado, exibe seu estado desativado, independentemente da interação do usuário.

O exemplo a seguir substitui a capa `selected_up`, que indica o estado selecionado.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente RadioButton para o Palco e clique nele duas vezes para abrir sua paleta de capas.
- 3 Defina o controle de zoom como 800% para ampliar o ícone para edição.
- 4 Clique duas vezes na capa `selected_up` para selecioná-la e pressione a tecla Delete para excluí-la.
- 5 No painel Ferramentas, selecione a ferramenta Retângulo.
- 6 No Inspetor de propriedades, defina a cor da linha como vermelha (#FF0000) e a cor de preenchimento, como preta (#000000).
- 7 A partir das cruzes que marcam o ponto de registro do símbolo (também chamado de *ponto de origem* ou *ponto zero*), clique e arraste o ponteiro, desenhando um retângulo.
- 8 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 9 Selecione Controlar > Testar filme.
- 10 Clique no RadioButton para selecioná-lo.

O RadioButton no estado selecionado deve ter uma aparência semelhante à mostrada na ilustração a seguir.



Personalizar o componente ScrollPane

Você pode transformar um componente ScrollPane horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou qualquer propriedade ou método aplicável da classe ScrollPane, como `height`, `width`, `scaleX` e `scaleY`.

O componente ScrollPane tem as seguintes características gráficas:

- O ponto de registro (também chamado de *ponto de origem* ou *ponto zero*) do conteúdo fica no canto superior esquerdo do painel.
- Quando a barra de rolagem horizontal está desativada, a barra de rolagem vertical é exibida da parte superior até a parte inferior, ao longo do lado direito do painel de rolagem. Quando a barra de rolagem vertical está desativada, a barra de rolagem horizontal é exibida da esquerda para a direita, ao longo da parte inferior do painel de rolagem. Você também pode desativar as duas barras de rolagem.
- Se o painel de rolagem for muito pequeno, talvez o conteúdo não seja exibido corretamente.
- Quando o painel de rolagem é redimensionado, a faixa de rolagem e a caixa de rolagem (miniatura) se expandem ou se contraem, e suas áreas de pressionamento são redimensionadas. Os botões permanecem do mesmo tamanho.

Usar estilos com o componente ScrollPane

As propriedades de estilo do componente ScrollPane especificam valores para as capas e o preenchimento de seu layout quando o componente é desenhado. Os diversos estilos de capas permitem especificar diferentes classes a serem usadas com as capas do componente. Para obter mais informações sobre o uso de estilos de capa, consulte “[Sobre capas](#)” na página 104.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente ScrollPane para o Palco e atribua a ele o nome de ocorrência de **mySp**.
- 3 Clique na guia Parâmetros, no Inspetor de propriedades, e digite o seguinte valor para o parâmetro `source`:
<http://www.helpexamples.com/flash/images/image1.jpg>.
- 4 No Quadro 1 da Linha do tempo principal, adicione o seguinte código ao painel Ações.

```
mySp.setStyle("contentPadding", 5);
```

Observe que o preenchimento é aplicado entre a borda e o conteúdo do componente, fora das barras de rolagem.

- 5 Selecione Controlar > Testar filme.

As capas e o ScrollPane

O componente ScrollPane usa uma borda e barras de rolagem nos ativos de rolagem. Para obter informações sobre a atribuição de capa a barras de rolagem, consulte “[Usar capas com o componente UI ScrollBar](#)” na página 136.

Personalizar o componente Slider

Você pode transformar um componente Slider horizontalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou qualquer propriedade aplicável da classe Slider, como `width` e `scaleX`.

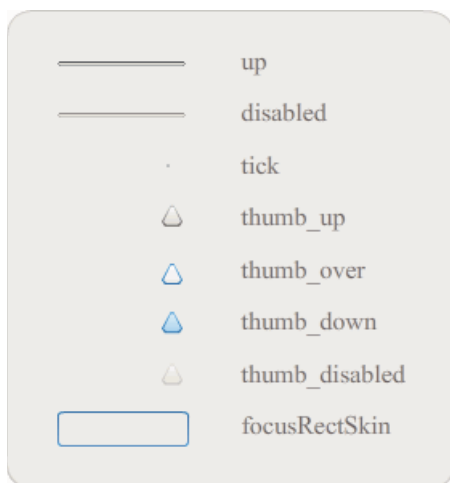
Você só pode tornar um controle deslizante mais longo. Não é possível aumentar sua altura. O Flash ignora a propriedade `height` e o parâmetro de altura do método `setSize()`. No entanto, você pode criar um controle deslizante vertical e torná-lo mais longo verticalmente.

Os estilos e o componente Slider

Os estilos do componente Slider especificam somente as classes de suas capas e um valor para `FocusRectPadding`, que especifica o número de pixels a serem usados no preenchimento entre a caixa delimitadora e o limite externo do componente. Para obter mais informações sobre o uso de estilos de capas, consulte “Sobre capas” na página 104.

As capas e o componente Slider

O componente Slider usa as seguintes capas, cuja aparência você pode alterar editando-as.



Capas de Slider

O exemplo a seguir edita a faixa ativa, alterando sua cor para azul.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente Slider do painel Componentes para o Palco.
- 3 Clique duas vezes no componente Slider para abrir seu painel de capas.
- 4 Clique duas vezes na marca de registro da faixa ativa para abri-la no modo de edição de símbolo.
- 5 Defina o controle de zoom como 800% para ampliar o ícone para edição. Observe que a faixa do Slider consiste em três barras.
- 6 Clique na barra superior para selecioná-la. Quando estiver selecionada, sua cor aparecerá no Seletor de cor de preenchimento, no Inspetor de propriedades.

- 7 Usando o Seletor de cor de preenchimento, no Inspetor de propriedades, selecione a cor #000066 para aplicá-la à barra superior da faixa do Slider.
- 8 Clique na barra central da faixa do Slider para selecioná-la. Quando estiver selecionada, sua cor aparecerá no Seletor de cor de preenchimento, no Inspetor de propriedades.
- 9 Usando o Seletor de cor de preenchimento, no Inspetor de propriedades, selecione a cor #0066FF para aplicá-la à barra central da faixa do Slider.
- 10 Clique na barra inferior da faixa do Slider para selecioná-la. Quando estiver selecionada, sua cor aparecerá no Seletor de cor de preenchimento, no Inspetor de propriedades.
- 11 Usando o Seletor de cor de preenchimento, no Inspetor de propriedades, selecione a cor #00CCFF para aplicá-la à barra inferior da faixa do Slider.
- 12 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 13 Selecione Controlar > Testar filme.

O Slider deverá aparecer como na ilustração a seguir.



Personalizar o componente TextArea

Você pode transformar um componente TextArea horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou qualquer propriedade aplicável da classe TextArea, como `height`, `width`, `scaleX` e `scaleY`.

Quando um componente TextArea é redimensionado, a borda é redimensionada de acordo com a nova caixa delimitadora. As barras de rolagem são colocadas na borda inferior e na borda direita, caso necessário. Então, a área de texto é redimensionada dentro da área restante; não há elementos de tamanho fixo em um componente TextArea. Se a largura do componente TextArea for muito estreita para exibir o tamanho do texto, o texto será recortado.

Os estilos e o componente TextArea

Os estilos do componente TextArea especificam valores para as capas, o preenchimento e o formato de texto quando o componente é desenhado. Os estilos `textFormat` e `disabledTextFormat` controlam o estilo do texto exibido pela TextArea. Para obter mais informações sobre propriedades de estilos de capa, consulte [“Usar capas com o componente TextArea”](#) na página 131.

O exemplo a seguir define o estilo `disabledTextFormat` para alterar a aparência do texto quando a TextArea está desativada, mas o mesmo processo se aplica à definição do estilo `textFormat` para uma TextArea ativada.

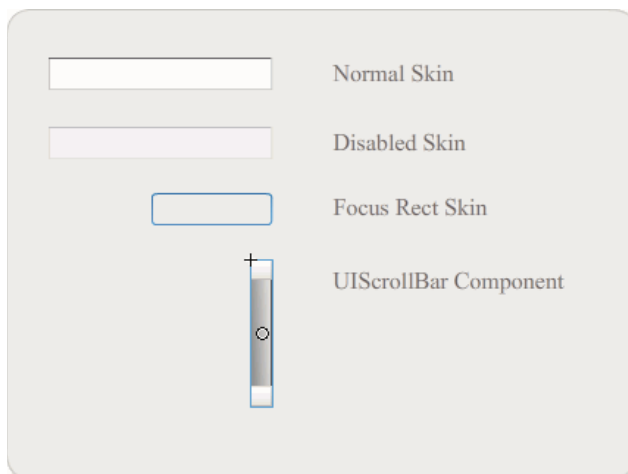
- 1 Crie um novo arquivo Flash.
- 2 Arraste um componente TextArea para o Palco e atribua a ele o nome de ocorrência de `myTa`.
- 3 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo principal.

```
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 Selecione Controlar > Testar filme.

Usar capas com o componente TextArea

O componente TextArea usa as seguintes capas, cuja aparência você pode alterar editando-as.



Capas de TextArea

Nota: Alterar a capa da ScrollBar em um componente causará a mesma alteração em todos os outros componentes que usam a ScrollBar.

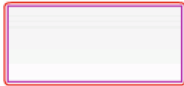
O procedimento a seguir altera as cores da borda da capa Focus Rect, que aparece quando a TextArea tem o foco, e da capa Normal.

- 1 Crie um novo arquivo Flash.
- 2 Arraste um componente TextArea para o Palco e clique nele duas vezes para abrir seu painel de ícones de capa.
- 3 Clique duas vezes na capa Focus Rect.
- 4 Clique na borda da capa Focus Rect para selecioná-la. Quando estiver selecionada, sua cor atual aparecerá no Seletor de cor de preenchimento, no Inspetor de propriedades.
- 5 Clique no Seletor de cor de preenchimento, no Inspetor de propriedades, para abri-lo e selecionar a cor #CC0000, que será aplicada à borda.
- 6 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 7 Clique duas vezes no componente TextArea para abrir seu painel de ícones de capa.
- 8 Clique duas vezes na capa Normal.
- 9 Selecione todas as extremidades da borda da capa Normal, uma de cada vez, e defina sua cor como #990099.

10 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.

11 Selecione Controlar > Testar filme.

Quando você selecionar a TextArea para começar a digitar texto, sua borda deverá aparecer como é mostrada na ilustração a seguir:



A borda externa é a capa Focus Rect e a borda interna é a borda da capa Normal.

Para obter informações sobre a edição da capa UIScrollBar, consulte “[Personalizar o componente UIScrollBar](#)” na página 136.

Personalizar o componente TextInput

Você pode alterar o tamanho de uma ocorrência de TextInput durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou as propriedades aplicáveis da classe TextInput, como `height`, `width`, `scaleX` e `scaleY`.

Quando um componente TextInput é redimensionado, a borda é redimensionada de acordo com a nova caixa delimitadora. O componente TextInput não utiliza barras de rolagem, mas o ponto de inserção rola automaticamente conforme o usuário interage com o texto. Então, o campo de texto é redimensionado dentro da área restante; não há elementos de tamanho fixo em um componente TextInput. Se o componente TextInput for muito pequeno para exibir o texto, então o texto será recortado.

Os estilos e o componente TextInput

Os estilos do componente TextInput especificam valores para as capas, o preenchimento e a formatação de texto quando o componente é desenhado. Os estilos `textFormat` e `disabledTextFormat` controlam o estilo do texto exibido no componente. Para obter mais informações sobre propriedades de estilos de capa, consulte “[As capas e o componente TextInput](#)” na página 133.

O exemplo a seguir define o estilo `textFormat` para definir a fonte, o tamanho e a cor do texto exibido no componente TextInput. O mesmo processo se aplica à definição do estilo `disabledTextFormat`, aplicado quando o componente está desativado.

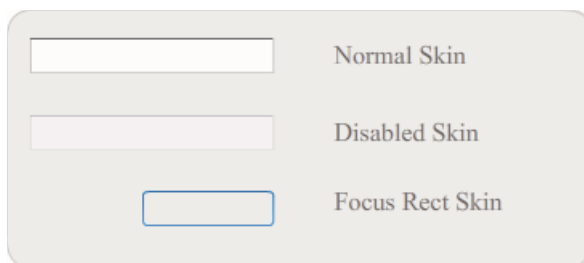
- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste um componente TextInput para o Palco e atribua a ele o nome de ocorrência de `myTi`.
- 3 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo principal.


```
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 Selecione Controlar > Testar filme.

As capas e o componente TextInput

O componente TextInput usa as seguintes capas, cuja aparência você pode alterar editando-as:



Legenda de TextInput

O procedimento a seguir altera as cores de borda e de plano de fundo de um componente TextInput:

- 1 Crie um novo arquivo Flash.
- 2 Arraste um componente TextInput para o Palco e clique nele duas vezes para abrir seu painel de capas.
- 3 Clique duas vezes na capa Normal.
- 4 Defina o controle de zoom como 800% para ampliar o ícone para edição.
- 5 Selecione todas as extremidades da borda da capa Normal, uma de cada vez, e defina a cor a ser aplicada como #993399.
- 6 Clique duas vezes no plano de fundo até que sua cor apareça no Seletor de cor de preenchimento, no Inspetor de propriedades. Selecione a cor #99CCCC para aplicá-la ao plano de fundo.
- 7 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 8 Selecione Controlar > Testar filme.

O componente TextInput deverá aparecer como é mostrado na ilustração a seguir:



Personalizar o componente TileList

Você pode transformar o componente `TileList` horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou as propriedades adequadas, como `width`, `height`, `columnCount`, `rowCount`, `scaleX` e `scaleY`. A `ScrollBar`, contida em `TileList`, é dimensionada com a caixa de listagem.

Os estilos e o componente TileList

Os estilos da `TileList` especificam valores para as capas, o preenchimento e a formatação de texto quando o componente é desenhado. Os estilos `textFormat` e `disabledTextFormat` controlam o estilo do texto exibido no componente. Para obter mais informações sobre as propriedades de estilos de capa, consulte [“Usar capas com o componente TileList”](#) na página 134.

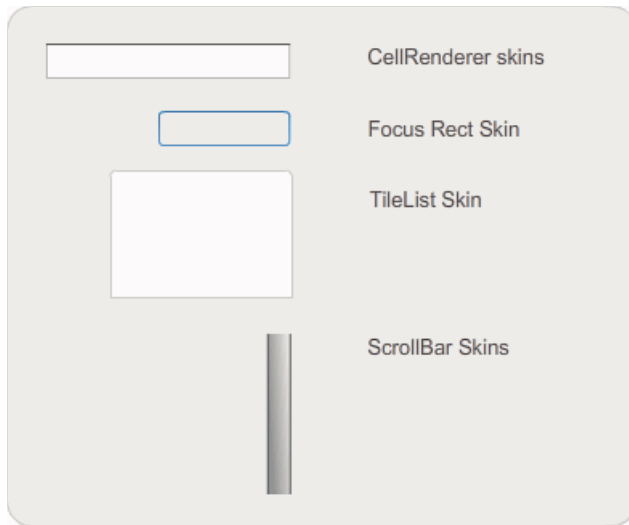
O exemplo a seguir chama o método `setRendererStyle()`, usando o estilo `textFormat` para definir a fonte, o tamanho, a cor e os atributos de texto dos rótulos exibidos em uma ocorrência de `TileList`. O mesmo processo se aplica à definição do estilo `disabledTextFormat`, aplicado quando a propriedade `enabled` é definida como `false`.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente `TileList` para o Palco e atribua a ele o nome de ocorrência de **myTl**.
- 3 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo.

```
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);
```

Usar capas com o componente TileList

O componente `TileList` tem uma capa `TileList`, uma capa `CellRenderer` e uma capa `ScrollBar`. Você pode editar essas capas para alterar a aparência da `TileList`:



Capas de TileList

Nota: Alterar a capa da ScrollBar em um componente causará a mesma alteração em todos os outros componentes que usam a ScrollBar.

O procedimento a seguir altera a cor da capa CellRenderer Selected_Up de TileList.

- 1 Crie um documento Flash (ActionScript 3.0).
- 2 Arraste o componente TileList para o Palco e clique nele duas vezes para abrir seu painel de capas.
- 3 Clique duas vezes na capa CellRenderer. Depois, clique duas vezes na capa Selected_Up e clique no plano de fundo retangular.
- 4 Selecione a cor #99FFFF usando o Seletor de cor de preenchimento, no Inspetor de propriedades, para aplicá-la à capa Selected_Up.
- 5 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 6 Na guia Parâmetros do Inspetor de propriedades, clique duas vezes na segunda coluna da linha dataProvider para abrir a caixa de diálogo Valores. Adicione itens com os seguintes rótulos: 1 item, 2 item, 3 item, 4 item.
- 7 Selecione Controlar > Testar filme.
- 8 Clique em uma das células na TileList para selecioná-la, e depois mova o mouse para fora da célula selecionada.

A célula selecionada deverá aparecer como na ilustração a seguir:



Componente TileList com a cor de capa Selected_Up modificada

Personalizar o componente UILoader

Você pode transformar um componente UILoader horizontal e verticalmente, durante a autoria e em tempo de execução. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou as propriedades adequadas, como `width`, `height`, `scaleX` e `scaleY`.

O comportamento de dimensionamento do componente UILoader é controlado pela propriedade `scaleContent`. Quando `scaleContent` é `true`, o conteúdo é dimensionado para ajustar-se aos limites do carregador (e redimensionado quando `setSize()` é chamado). Quando `scaleContent` é `false`, o tamanho do componente é fixo, de acordo com o tamanho do conteúdo, e `setSize()` e as propriedades de dimensionamento não têm efeito.

O componente UILoader não tem elementos de interface do usuário aos quais você possa aplicar estilos ou capas.

Personalizar o componente UIScrollBar

Você pode transformar um componente UIScrollBar horizontal e verticalmente, durante a autoria e em tempo de execução. No entanto, uma UIScrollBar vertical não permite modificações de largura, e uma UIScrollBar horizontal não permite modificações de altura. Durante a autoria, selecione o componente no Palco e use a ferramenta Transformação livre ou qualquer um dos comandos Modificar > Transformar. Em tempo de execução, use o método `setSize()` ou qualquer propriedade aplicável da classe UIScrollBar, como `width`, `height`, `scaleX` e `scaleY`.

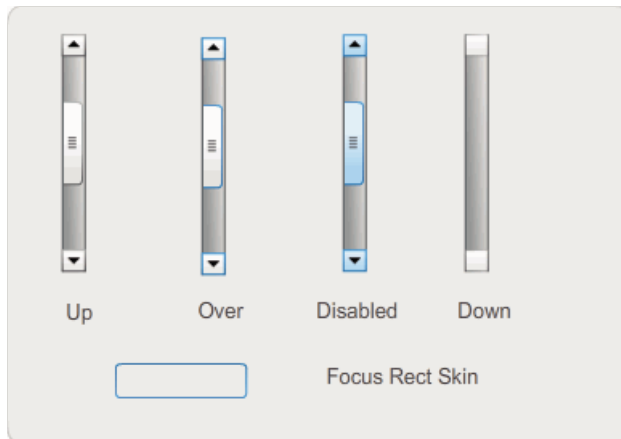
Nota: Se você usar o método `setSize()`, poderá alterar somente a largura de uma barra de rolagem horizontal ou a altura de uma barra de rolagem vertical. Durante a autoria, você pode definir a altura de uma barra de rolagem horizontal ou a largura de uma barra de rolagem vertical, mas os valores serão redefinidos quando o filme for publicado. Somente a dimensão correspondente ao comprimento de uma barra de rolagem pode ser alterada.

Usar estilos com o componente UIScrollBar

Os estilos do componente UIScrollBar especificam somente as classes de suas capas e um valor para `FocusRectPadding`, que especifica o número de pixels a serem usados no preenchimento entre a caixa delimitadora e o limite externo do componente. Para obter mais informações sobre o uso de estilos de capas, consulte [“Sobre capas”](#) na página 104.

Usar capas com o componente UIScrollBar

O componente UIScrollBar usa as capas a seguir.



Capas de UI ScrollBar

As barras de rolagem horizontais e verticais usam as mesmas capas. Quando exibe uma barra de rolagem horizontal, o componente UI ScrollBar gira as capas conforme for adequado.

Nota: Alterar a capa da ScrollBar em um componente causará a mesma alteração em todos os outros componentes que usam a ScrollBar.

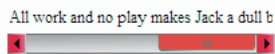
O exemplo a seguir demonstra como alterar a cor da miniatura e dos botões de seta da UI ScrollBar.

- 1 Crie um novo documento Flash (ActionScript 3.0).
- 2 Arraste o componente UI ScrollBar para o Palco e atribua a ele o nome de ocorrência de **mySb**. Na guia Parâmetros, defina a direção como horizontal.
- 3 Clique duas vezes na barra de rolagem para abrir seu painel de capas.
- 4 Clique na capa Up para selecioná-la.
- 5 Defina o controle de zoom como 400% para ampliar o ícone para edição.
- 6 Clique duas vezes no plano de fundo da seta para a direita (ou da seta para cima, em uma barra de rolagem vertical) até que o plano de fundo seja selecionado e sua cor apareça no Seletor de cor de preenchimento, no Inspetor de propriedades.
- 7 Selecione a cor #CC0033 para aplicá-la ao plano de fundo do botão.
- 8 Clique no botão Voltar, do lado esquerdo da barra de edição acima do Palco, para voltar ao modo de edição de documento.
- 9 Repita as etapas 6, 7 e 8 para os elementos de miniatura e de seta para a esquerda (ou de seta para baixo, em uma barra de rolagem vertical).
- 10 Adicione o seguinte código ao painel Ações, no Quadro 1 da Linha do tempo, para anexar a barra de rolagem a um 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 Selecione Controlar > Testar filme.

O componente UIScrollBar deverá aparecer como na ilustração a seguir.



ScrollBar horizontal com miniatura e setas para a esquerda e para a direita em vermelho

Capítulo 6: Uso do componente FLVPlayback

O componente FLVPlayback permite incluir com facilidade um vídeo player em seu aplicativo Adobe Flash CS5 Professional para reproduzir progressivamente arquivos de vídeo baixados em HTTP ou arquivos de vídeo de fluxo contínuo do Adobe Macromedia Flash Media Server ou de um FVSS (Flash Video Streaming Service).

Com o lançamento do Adobe Flash Player 9 Atualização 3 (versão 9.0.115.0 ou posterior), foram incorporadas melhorias significativas para reprodução de conteúdo de vídeo no Flash Player. Esta atualização inclui alterações no componente FLVPlayback que se beneficia do hardware de vídeo do sistema do usuário final para oferecer melhor desempenho de reprodução de vídeo. As alterações no componente FLVPlayback também aumentam a fidelidade de arquivos de vídeo exibidos no modo de tela cheia.

Além disso, o Flash Player 9 Atualização 3 aprimora a funcionalidade do componente FLVPlayback, adicionando suporte para os formatos de vídeo MPEG-4 de alta definição que utilizam a codificação H.264 padrão do setor. Esses formatos incluem MP4, M4A, MOV, MP4V, 3GP e 3G2.

***Nota:** Arquivos MP4 protegidos, como os baixados da Apple® iTunes® ou digitalmente criptografados pelo FairPlay®, não têm suporte.*

O componente FLVPlayback fácil de usar tem as seguintes características e benefícios:

- Pode ser arrastado para o Palco e implementado de maneira rápida e bem-sucedida
- Oferece suporte ao tamanho de tela cheia
- Oferece um conjunto de *capas* previamente criadas, que permitem personalizar a aparência dos respectivos controles de reprodução
- Permite selecionar cor e valores alfa de capas previamente criadas
- Permite que usuários avançados criem suas próprias capas
- Oferece visualização ao vivo durante a criação
- Oferece propriedades de layout para manter o arquivo de vídeo centralizado durante o redimensionamento
- Permite iniciar a reprodução quando o download progressivo já tiver baixado parte suficiente do arquivo de vídeo.
- Oferece pontos de sinalização que permitem sincronizar o vídeo com textos, gráficos e animações
- *Mantém um arquivo SWF de tamanho razoável

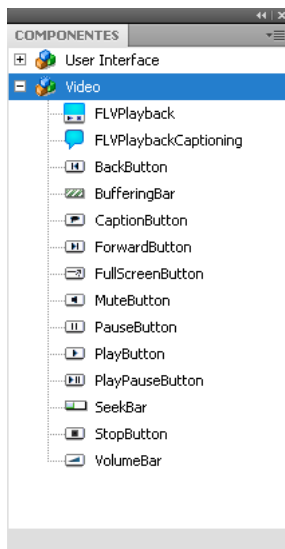
Usar o componente FLVPlayback

Usar o componente FLVPlayback consiste em colocá-lo no Palco e especificar um arquivo de vídeo para que ele reproduza. Além disso, você também pode definir vários parâmetros que controlam seu comportamento e descrevem o arquivo de vídeo.

Uso do componente FLVPlayback

O componente FLVPlayback também inclui uma API (interface de programação de aplicativo) do ActionScript. A API inclui as seguintes classes, descritas por completo na [Referência do ActionScript 3.0 para Adobe Flash Platform](#): CuePointType, FLVPlayback, FLVPlaybackCaptioning, NCManager, NCManagerNative, VideoAlign, VideoError, VideoPlayer, VideoState e várias classes de eventos - AutoLayoutEvent, LayoutEvent, MetadataEvent, SkinErrorEvent, SoundEvent, VideoEvent e VideoProgressEvent.

O componente FLVPlayback inclui os componentes de UI personalizada de reprodução FLV. O componente FLVPlayback é uma combinação da área de exibição ou player de vídeo, em que você exibe o arquivo de vídeo e os controles que permitem operá-lo. Os componentes da UI personalizada de reprodução FLV oferecem botões de controle e mecanismos que podem ser usados para reproduzir, parar, pausar e controlar o arquivo de vídeo. Esses controles incluem BackButton, BufferingBar, CaptionButton (para FLVPlaybackCaptioning), ForwardButton, FullScreenButton, MuteButton, PauseButton, PlayButton, PlayPauseButton, SeekBar, StopButton e VolumeBar. Os controles de componente FLVPlayback e de UI personalizada de reprodução FLV são exibidos no painel Componentes, como mostrado na figura a seguir:



Componentes FLVPlayback no painel Componentes

O processo de adição de controles de reprodução ao componente FLVPlayback é chamado de *atribuição de capa*. O componente FLVPlayback tem uma capa inicial padrão, SkinOverAll.swf, que fornece os controles para reproduzir, parar, voltar, avançar, de barra de pesquisa, mudo, volume, tela cheia e colocação de legendas. Para alterar essa capa, você tem as seguintes opções:

- Selecione em um conjunto de capas previamente criadas
- Crie uma capa personalizada e adicione-a ao conjunto de capas previamente criadas
- Selecione controles individuais nos componentes da UI personalizada de reprodução FLV e os personalize

Quando você seleciona uma capa previamente criada, é possível escolher a cor e os valores alfa da capa separadamente, durante a criação ou no tempo de execução. Para obter mais informações, consulte [“Selecionar uma capa previamente criada”](#) na página 158.

Após selecionar outra capa, a capa selecionada se tornará a nova capa padrão.

Para obter mais informações sobre como selecionar ou criar uma capa para o componente Componente FLVPlayback, consulte [“Personalizar o componente FLVPlayback”](#) na página 157.

Crie um aplicativo com o componente FLVPlayback

Você pode incluir o componente FLVPlayback em seu aplicativo destas formas:

- Arraste o componente FLVPlayback do painel Componentes para o Palco e especifique um valor para o parâmetro `source`.
- Use o assistente de importação de vídeo para criar o componente no Palco e personalize-o selecionando uma capa.
- Use o construtor `FLVPlayback()` para criar dinamicamente uma ocorrência de FLVPlayback no Palco, presumindo que o componente esteja na biblioteca.

Nota: Se você criar uma ocorrência de FLVPlayback com o ActionScript, também deverá atribuir uma capa a ela, definindo a propriedade `skin` com o ActionScript. Quando você aplica uma capa dessa maneira, ela não é publicada automaticamente com o arquivo SWF. Você deve copiar o arquivo de aplicativo SWF e o arquivo de capa SWF no servidor do aplicativo, ou o arquivo de capa SWF não estará disponível quando você executar o aplicativo.

Arraste o componente FLVPlayback do painel Componentes

- 1 No painel Componentes, clique no botão de adição (+) para abrir a entrada de vídeo.
- 2 Arraste o componente FLVPlayback para o Palco.
- 3 Com o componente FLVPlayback selecionado no Palco, localize a célula Valor do parâmetro `source` na guia Parâmetros do Inspetor de componentes; insira uma string que especifique um dos seguintes itens:
 - Um caminho local para o arquivo de vídeo
 - Uma URL para o arquivo de vídeo
 - Uma URL para um arquivo SMIL (Linguagem de integração multimídia sincronizada) que descreva como reproduzir o arquivo de vídeo

Para obter informações sobre como criar um arquivo SMIL para descrever um ou mais arquivos FLV, consulte [“Usar um arquivo SMIL”](#) na página 169.
- 4 Na guia Parâmetros do Inspetor de componentes, com o componente FLVPlayback selecionado no Palco, clique na célula Valor do parâmetro `skin`.
- 5 Clique no ícone de lupa para abrir a caixa de diálogo Selecionar capa.
- 6 Selecione uma destas opções:
 - Na lista suspensa Capa, selecione uma das capas previamente criadas para anexar um conjunto de controles de reprodução ao componente.
 - Se você criou uma capa personalizada, selecione URL de cobertura personalizada, no menu pop-up, e insira na caixa URL, a URL do arquivo SWF que contém a capa.
 - Selecione Nenhum e arraste os componentes individuais da UI personalizada de reprodução FLV para o Palco, para adicionar controles de reprodução.

Nota: Nos dois primeiros casos, uma visualização da capa é exibida no painel de exibição acima do menu pop-up. Você pode usar o seletor de cores para alterar a cor da capa.

Para alterar a cor de um controle de UI personalizada, você deve personalizá-lo. Para obter mais informações sobre como usar controles de UI personalizada, consulte [“Colocar capa em componentes de UI personalizada de reprodução FLV individualmente”](#) na página 159.
- 7 Clique em OK para fechar a caixa de diálogo Selecionar capa.
- 8 Selecione Controle > Testar filme para executar o arquivo SWF e iniciar o vídeo.

O seguinte procedimento usa o assistente de importação de vídeo para adicionar um componente FLVPlayback:

Uso do assistente de importação de vídeo:

- 1 Selecione Arquivo > Importar > Importar vídeo.
- 2 Indique o local do arquivo de vídeo selecionando uma das seguintes opções:
 - Em meu computador local
 - Já implantado em um servidor Web, no Flash Video Streaming Service ou no Flash Media Server
- 3 Dependendo de sua escolha, insira o caminho ou a URL que especifica o local do arquivo de vídeo e, em seguida, clique em Avançar.
- 4 Se você selecionou um caminho de arquivo, verá a caixa de diálogo Implantação em seguida, na qual você poderá selecionar uma das opções listadas para especificar como gostaria de implantar seu vídeo:
 - Download progressivo a partir de um servidor Web padrão
 - Fluxo do Flash Video Streaming Service
 - Fluxo do Flash Media Server
 - Incorporar vídeo em um arquivo SWF e reproduzir na Linha do tempo

***Importante:** Não selecione a opção Incorporar vídeo. O componente FLVPlayback executa apenas vídeos de fluxo contínuo externo. Esta opção não colocará um componente FLVPlayback no Palco.*
- 5 Clique em Avançar.
- 6 Selecione uma destas opções:
 - Na lista suspensa Capa, selecione uma das capas previamente criadas para anexar um conjunto de controles de reprodução ao componente.
 - Se você criou uma capa personalizada para o componente, selecione URL de cobertura personalizada, no menu pop-up, e insira a URL para o arquivo SWF que contém a capa na caixa URL.
 - Selecione Nenhum e arraste os componentes individuais da UI personalizada de reprodução FLV para o Palco, para adicionar controles de reprodução.

***Nota:** Nos dois primeiros casos, uma visualização da capa é exibida no painel de exibição acima do menu pop-up.*
- 7 Clique em OK para fechar a caixa de diálogo Selecionar capa.
- 8 Leia a caixa de diálogo Concluir importação de vídeo para observar o que acontece em seguida; depois, clique em Concluir.
- 9 Se você não salvou seu arquivo FLA, a caixa de diálogo Salvar como será exibida.
- 10 Selecione Controle > Testar filme para executar o arquivo SWF e iniciar o vídeo.

O procedimento seguinte adiciona o componente FLVPlayback usando o ActionScript.

Criar uma ocorrência dinamicamente usando o ActionScript:

- 1 Arraste o componente FLVPlayback do painel Componentes para o painel Biblioteca (Janela > Biblioteca).
- 2 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo. Altere `install_drive` para a unidade na qual você instalou o Flash e modifique o caminho para refletir o local da pasta Capas para instalação:

Em um computador Windows:

Uso do componente FLVPlayback

```
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";
```

Em um computador 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";
```

Nota: Se você não definir as propriedades *source* e *skin*, o clipe de filme gerado parecerá vazio.

- 3 Selecione Controle > Testar filme para executar o arquivo SWF e iniciar o arquivo de vídeo.

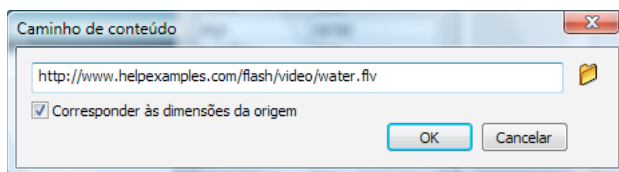
Parâmetros do componente FLVPlayback

Para cada ocorrência do componente FLVPlayback, é possível definir os seguintes parâmetros no Inspetor de componentes ou no Inspetor de propriedades: *align*, *autoplay*, *cuePoints*, *preview*, *scaleMode*, *skin*, *skinAutoHide*, *skinBackgroundAlpha*, *skinBackgroundColor*, *source* e *volume*. Cada um desses parâmetros tem uma propriedade ActionScript correspondente de mesmo nome. Ao atribuir um valor a esses parâmetros, você está definindo o estado inicial da propriedade no aplicativo. A definição da propriedade no ActionScript substitui o valor definido no parâmetro. Para obter informações sobre os possíveis valores desses parâmetros, consulte a classe FLVPlayback na [Referência do ActionScript 3.0 para Flash Professional](#).

Especifique o parâmetro *source* do FLVPlayback

O parâmetro *source* permite especificar o nome e o local do arquivo de vídeo, que informam ao Flash como reproduzir o arquivo.

Abra a caixa de diálogo Caminho de conteúdo, clicando duas vezes na célula Valor do parâmetro de *source*, no Inspetor de componentes.



Caixa de diálogo Caminho de conteúdo do FLVPlayback

A caixa de diálogo Caminho de conteúdo apresenta a caixa de seleção *Corresponder dimensões do FLV de origem*, que especifica se a ocorrência FLVPlayback no Palco deve corresponder às dimensões do arquivo de vídeo de origem. O arquivo de vídeo de origem contém as dimensões de altura e largura recomendadas para reprodução. Se você selecionar essa opção, as dimensões da ocorrência FLVPlayback serão redimensionadas para corresponder a essas dimensões preferenciais.

A origem

Insira a URL ou caminho do local do arquivo de vídeo ou um arquivo XML que descreva como reproduzir o arquivo de vídeo. Se você não souber o local exato do arquivo de vídeo, clique no ícone de pasta para abrir a caixa de diálogo Navegador, para ajudá-lo a encontrar o local correto. Quando estiver procurando um arquivo de vídeo, se ele estiver abaixo ou no local de destino do arquivo SWF, o Flash especifica automaticamente o caminho relativo até esse local, para que você possa servi-lo em um servidor Web. Caso contrário, o caminho será um caminho absoluto do Windows ou do Macintosh. Para inserir o nome de um arquivo XML local, digite o caminho e o nome.

Se você especificar uma URL HTTP, o arquivo de vídeo será reproduzido como download progressivo. Se você especificar uma URL que seja uma URL RTMP, o arquivo de vídeo será transmitido do Flash Media Server ou do FVSS. A URL para o arquivo XML também pode ser um arquivo de vídeo de fluxo contínuo do Flash Media Server ou do FVSS.

Importante:

Também é possível especificar o local de um arquivo SMIL que descreve como reproduzir vários arquivos de vídeo de fluxo contínuo para múltiplas larguras de banda. O arquivo usa a SMIL (Synchronized Multimedia Integration Language - Linguagem de Integração de Multimídia Sincronizada) para descrever os arquivos FLV. Para obter uma descrição do arquivo SMIL, consulte “[Usar um arquivo SMIL](#)” na página 169.

Você também pode especificar o nome e o local do arquivo de vídeo usando o `FLVPlayback` do ActionScript .A propriedade `source` e os métodos `FLVPlayback.play()` e `FLVPlayback.load()`. Essas três alternativas têm prioridade sobre o parâmetro `source` no Inspetor de componentes. Para obter mais informações, consulte as entradas `FLVPlayback.source`, `FLVPlayback.play()` e `FLVPlayback.load()` da classe Classe `FLVPlayback` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

Suporte de tela cheia

A versão do ActionScript 3.0 do componente `FLVPlayback` oferece suporte ao modo de tela cheia, o que requer o Flash Player 9.0.28.0 ou posterior e também que o HTML esteja configurado corretamente para exibição em tela cheia. Algumas capas previamente criadas incluem um botão de alternância para alternar entre o modo de tela cheia ligado e desligado. O ícone `FullScreenButton` é exibido do lado direito da barra de controle na ilustração a seguir.



Ícone Tela cheia na barra de controle

Ocorre suporte de tela cheia apenas se a propriedade `fullScreenTakeOver` estiver configurada como `true`, que é o padrão.

O suporte de tela cheia pode ocorrer com ou sem suporte para aceleração de hardware. Para obter mais informações sobre o suporte para aceleração de hardware, consulte “[Aceleração por hardware](#)” na página 147.

Para implementar o suporte de tela cheia de FLVPlayback:

- 1 Adicione o componente `FLVPlayback` ao seu aplicativo e atribua um arquivo de vídeo a ele.
- 2 Selecione uma capa para o componente `FLVPlayback` que tenha o botão de tela cheia (por ex., `SkinUnderPlaySeekFullscreen.swf`) ou adicione o componente `FullScreenButton` da interface do usuário ao componente `FLVPlayback` na seção `Video` do painel `Componentes`.
- 3 Selecione `Arquivo > Configurações de publicação`.
- 4 Na caixa de diálogo `Configurações de publicação`, clique na guia `HTML`.
- 5 Na guia `HTML`, selecione `Flash com suporte para tela cheia`, no menu pop-up `Modelo`.

- 6 Além disso, na guia HTML, selecione a caixa de seleção Detectar versão do Flash e especifique uma versão de 9.0.28 ou posterior, dependendo da versão de Flash Player que estiver usando.
- 7 Selecione a guia Formatos e certifique-se de que as opções Flash (.swf) e HTML (.html) estejam selecionadas. Você pode substituir os nomes de arquivo padrão.
- 8 Clique em Publicar e, em seguida, clique em OK.

Como alternativa para a etapa 7, você pode clicar em OK e, em seguida, selecionar Arquivo > Publicar visualização > Padrão - (HTML) para abrir automaticamente o arquivo HTML exportado em seu navegador padrão. Ou abra o arquivo HTML exportado com o navegador para testar a opção de tela cheia.

Para adicionar o componente FLVPlayback com suporte de tela cheia à sua página da Web, abra o arquivo HTML exportado e copie o código que incorpora o arquivo SWF no arquivo HTML de sua página da Web. Este código deve se parecer com o exemplo a seguir:

```
//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;align&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>
```

Como alternativa, você pode usar o arquivo HTML exportado como modelo para sua página da Web e adicionar a ele outro conteúdo. Porém, se fizer isso, altere o nome do arquivo HTML para não sobregravá-lo acidentalmente caso exporte posteriormente o arquivo HTML FLVPlayback do Flash outra vez.

De um ou outro modo, você também deve carregar o arquivo AC_RunActiveContent.js para o servidor Web, que é exportado para a mesma pasta do arquivo HTML.

O suporte do ActionScript para o modo de tela cheia inclui as propriedades `fullScreenBackgroundColor`, `fullScreenSkinDelay`, e `fullScreenTakeOver` e o método `enterFullScreenDisplayState()`. Para obter informações sobre esses elementos do ActionScript, consulte a [Referência do ActionScript 3.0 para Flash Professional](#).

Uso de `enterFullScreenDisplayState()`

Você também pode chamar o modo de tela cheia, chamando o método `enterFullScreenDisplayState()` do ActionScript, conforme mostra o exemplo a seguir.

```
function handleClick(e:MouseEvent):void {  
    myFLVPlybk.enterFullScreenDisplayState();  
}  
myButton.addEventListener(MouseEvent.CLICK, handleClick);
```

Neste exemplo, o modo de tela cheia *não* é chamado clicando no botão de alternância de tela cheia em uma capa FLVPlayback, mas, em vez disso, clicando em um botão (MyButton) que o criador da página da Web incluiu para chamar o modo de tela cheia. Clicar no botão aciona o manipulador de eventos `handleClick`, que chama o método `enterFullScreenDisplayState()`.

O método `enterFullScreenDisplayState()` define a propriedade `Stage.displayState` para `StageDisplayState.FULL_SCREEN` e, portanto, carrega as mesmas restrições da propriedade `displayState`. Para obter mais informações sobre o método `enterFullScreenDisplayState()` e a propriedade `Stage.displayState`, consulte a [Referência do ActionScript 3.0 para Flash Professional](#).

Aceleração por hardware

O Flash Player 9.0.115.0 e versões superiores incluem o código que aproveita o hardware de vídeo disponível para melhorar o desempenho e a fidelidade de arquivos FLV que o FLVPlayback reproduz no modo de tela cheia. Se os pré-requisitos forem atendidos e a propriedade `fullScreenTakeOver` estiver definida como `true`, o Flash Player usará a aceleração de hardware para dimensionar o arquivo de vídeo, em vez de dimensioná-lo por meio de software. Se o componente FLVPlayback for executado em uma versão anterior de Flash Player ou se não houver pré-requisitos de aceleração de hardware, o Flash Player dimensionará o próprio arquivo de vídeo, como fez anteriormente.

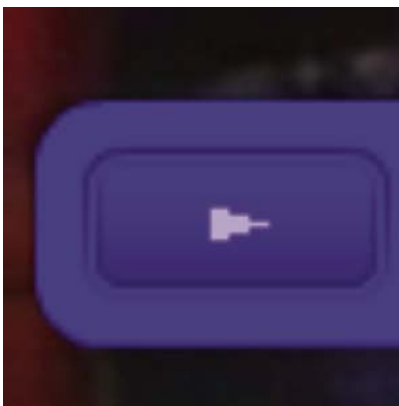
Para aproveitar a aceleração de hardware no suporte de tela cheia, o computador deve ter uma placa de vídeo compatível com DirectX 7 com 4 MB ou mais de VRAM (RAM de vídeo). Esse suporte de hardware está disponível no Windows 2000 ou no Mac OS X 10.2 e versões posteriores desses sistemas operacionais. O Direct X[®] fornece APIs que formam uma interface entre o software e o hardware de vídeo para acelerar gráficos tridimensionais e bidimensionais, entre outras coisas.

Para aproveitar o modo de aceleração de hardware, você também deve chamar o modo de tela cheia de uma das seguintes maneiras:

- Uso do botão de alternância de tela cheia em uma capa FLVPlayback
- Usar o controle de vídeo `FullScreenButton`
- Usar o método `enterFullScreenDisplayState()` do ActionScript. Para obter mais informações, consulte “[Uso de enterFullScreenDisplayState\(\)](#)” na página 146.

Se você chamar o modo de tela cheia definindo a propriedade `Stage.displayState` como `StageDisplayState.FULLSCREEN`, o FLVPlayback não usará a aceleração de hardware, mesmo que o hardware de vídeo e a memória estejam disponíveis.

Uma consequência de usar a aceleração de hardware para suporte de tela cheia é que as capas FLVPlayback são dimensionadas juntamente com o player de vídeo e o arquivo de vídeo. A imagem seguinte mostra o efeito do modo de tela cheia com aceleração de hardware na capa FLVPlayback, um detalhe mostrado aqui em resolução total.



Modo de tela cheia em um monitor de 1600 x 1200 com vídeo de 320x240 pixels

Essa imagem mostra o resultado de usar o modo tela de cheia em um monitor 1600 x 1200 com um arquivo de vídeo que tenha largura de 320 e altura de 240, as dimensões padrão do FLVPlayback. O efeito de distorção na capa é mais pronunciado nos arquivos FLV com dimensões menores ou em um monitor maior. Por outro lado, o efeito de distorção é menos pronunciado em arquivos FLV maiores ou em monitores menores. Por exemplo, alterar de 640 x 480 para 1600 x 1200 ainda aumenta o tamanho da capa, mas ela parece menos distorcida.

Você pode definir a propriedade `skinScaleMaximum` para limitar o dimensionamento da capa FLVPlayback. O valor padrão é 4,0 ou 400%. Porém, limitar o dimensionamento da capa requer uma combinação de hardware e software para dimensionar o FLV e isso pode afetar de forma adversa o desempenho em FLVs com dimensões grandes que sejam codificados em taxas de bits maiores. Se o vídeo for grande (por exemplo, 640 pixels de largura ou mais, 480 pixels de altura ou mais), você não deve definir `skinScaleMaximum` para um valor pequeno, pois isso pode causar problemas de desempenho em monitores grandes. A propriedade `skinScaleMaximum` permite gerenciar a compensação entre desempenho e qualidade e a aparência de uma capa grande.

Sair do modo de tela cheia

Para sair do modo de tela cheia, clique novamente no botão de tela cheia ou pressione a tecla Esc.

Se você definir as propriedades a seguir e chamar os métodos a seguir, poderão ocorrer alterações no layout que causam a saída do componente FLVPlayback do modo de tela cheia: `height`, `registrationHeight`, `registrationWidth`, `registrationX`, `registrationY`, `scaleX`, `scaleY`, `width`, `x`, `y`, `setScale()` ou `setSize()`.

Se você define as propriedades `align` ou `scaleMode`, o FLVPlayback as define como `center` e `maintainAspectRatio` até sair do modo de tela cheia.

Alterar o valor da propriedade `fullScreenTakeOver` de `true` para `false` quando você estiver usando o modo de aceleração de hardware de tela cheia, também faz com que o Flash saia do modo de tela cheia.

Alinhamento de layout para reproduzir vários arquivos de vídeo

O ActionScript 3.0 FLVPlayback tem a propriedade `align`, que especifica se o arquivo de vídeo deve ser centralizado ao ser redimensionado ou posicionando na parte superior, inferior, esquerda ou direita do componente. Além das propriedades `x`, `y`, `width` e `height` do componente, o ActionScript 3.0 também tem as propriedades `registrationX`, `registrationY`, `registrationWidth` e `registrationHeight`. Inicialmente, elas correspondem às propriedades `x`, `y`, `width` e `height`. Ao carregar arquivos de vídeo subsequentes, a redefinição automática de layout não os altera, portanto, o novo arquivo de vídeo pode ser centralizado no mesmo lugar. Se `scaleMode = VideoScaleMode.MAINTAIN_ASPECT_RATIO`, os arquivos FLV subsequentes podem ser ajustados de acordo com as dimensões originais do componente, em vez de fazer com que a largura e altura do componente sejam alterados.

Reprodução automática de arquivos de vídeo com download progressivo

Ao carregar um arquivo de vídeo com download progressivo, o FLVPlayback só começa a reproduzir o arquivo de vídeo quando já tiver sido baixado o suficiente, para que ele possa reproduzir o arquivo de vídeo do início ao fim.

Se você deseja reproduzir o arquivo de vídeo antes que ele tenha sido baixado o suficiente, chame o método `play()` sem nenhum parâmetro.

Se você deseja retornar para o estado de espera de quantidade suficiente do arquivo de vídeo para download, chame o método `pause()` e, em seguida, o método `playWhenEnoughDownloaded()`.

Uso de pontos de sinalização

Ponto de sinalização é um ponto em que o player de vídeo despacha um evento `cuePoint` durante a reprodução de um arquivo de vídeo. Você pode adicionar pontos de sinalização a um arquivo FLV quando desejar que uma ação ocorra em outro elemento na página da Web. Talvez você deseje exibir texto ou um gráfico, por exemplo, ou sincronizar com uma animação Flash ou, ainda, afetar a reprodução do arquivo FLV pausando-o, buscando um ponto diferente no vídeo ou alterando para um arquivo FLV diferente. Os pontos de sinalização permitem receber controle no seu código do ActionScript para sincronizar pontos no arquivo FLV com outras ações na página da Web.

Existem três tipos de pontos de sinalização: navegação, evento e ActionScript. Os pontos de sinalização navegação e evento também são conhecidos como pontos de sinalização *incorporados* porque são incorporados no fluxo de arquivo FLV e no pacote de metadados do arquivo FLV.

O *ponto de sinalização de navegação* permite buscar um quadro específico no arquivo FLV, pois ele cria um quadro-chave no arquivo FLV o mais próximo possível do tempo especificado. O *quadro-chave* é um segmento de dados que ocorre entre os quadros de imagem no fluxo de arquivo FLV. Ao buscar um ponto de sinalização de navegação, o componente busca o quadro-chave e inicia o evento `cuePoint`.

O *ponto de sinalização de evento* permite sincronizar um ponto no tempo no arquivo FLV com um evento externo na página da Web. O evento `cuePoint` ocorre precisamente no tempo especificado. Você pode incorporar pontos de sinalização de navegação e de evento em um arquivo FLV usando o assistente de importação de vídeo ou o Flash Video Encoder. Para obter mais informações sobre o assistente de importação de vídeo e o Flash Video Encoder, consulte o Capítulo 16, "Trabalho com vídeo", em *Uso do Flash*.

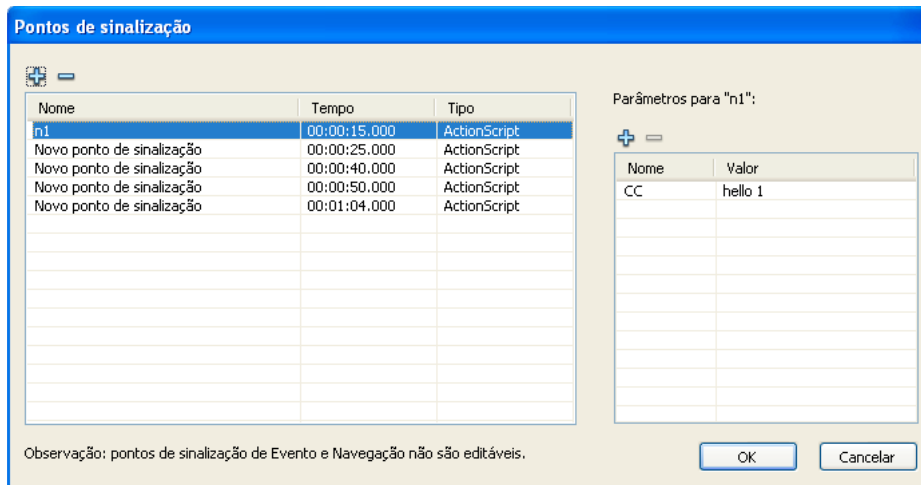
O *ponto de sinalização do ActionScript* é um ponto de sinalização externo que pode ser adicionado por meio da caixa de diálogo Pontos de sinalização Flash Video do componente ou por meio do método `FLVPlayback.addASCuePoint()`. O componente armazena e rastreia os pontos de sinalização do ActionScript, com exceção do arquivo FLV, e, conseqüentemente, eles são menos precisos do que os pontos de sinalização incorporados. Os pontos de sinalização do ActionScript têm precisão de um décimo de segundo. Você pode aumentar a precisão dos pontos de sinalização do ActionScript reduzindo o valor da propriedade `playheadUpdateInterval`, pois o componente gera o evento `cuePoint` para pontos de sinalização do ActionScript quando o indicador de reprodução é atualizado. Para obter mais informações, consulte a propriedade Propriedade `FLVPlayback.playheadUpdateInterval` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

No ActionScript e nos metadados do arquivo FLV, o ponto de sinalização é representado como um objeto com as seguintes propriedades: `name`, `time`, `type` e `parameters`. A propriedade `name` é uma string que contém o nome atribuído do ponto de sinalização. A propriedade `time` é um número representando o tempo em horas, minutos, segundos e milésimos de segundo (HH:MM:SS.mmm) quando ocorre o ponto de sinalização. A propriedade `type` é uma string cujo valor é "navigation", "event" ou "actionscript", dependendo do tipo de ponto de sinalização criado. A propriedade `parameters` é uma matriz de nome e pares de valores especificados.

Quando o evento `cuePoint` ocorre, o objeto de ponto de sinalização fica disponível no objeto de evento por meio da propriedade `info`.

Use a caixa de diálogo Pontos de sinalização Flash Video

Abra a caixa de diálogo Pontos de sinalização Flash Video, clicando duas vezes na célula Valor, do parâmetro `cuePoints`, no Inspetor de componentes. A caixa de diálogo se parece com a figura a seguir:



Caixa de diálogo Pontos de sinalização

A caixa de diálogo exibe pontos de sinalização incorporados e do ActionScript. Você pode usar essa caixa de diálogo para adicionar e excluir pontos de sinalização do ActionScript, bem como parâmetros de pontos de sinalização. Você também pode ativar ou desativar pontos de sinalização incorporados. No entanto, você não pode adicionar, alterar nem excluir pontos de sinalização incorporados.

Adicionar ponto de sinalização do ActionScript:

- 1 Clique duas vezes na célula Valor, do parâmetro `cuePoints`, no Inspetor de componentes, para abrir a caixa de diálogo Pontos de sinalização Flash.
- 2 Clique no sinal de adição (+) no canto superior esquerdo, acima da lista de pontos de sinalização, para adicionar uma entrada de ponto de sinalização do ActionScript.
- 3 Clique no texto Novo ponto de sinalização, na coluna Nome, e edite o texto para nomear o ponto de sinalização.
- 4 Clique no valor Tempo de 00:00:00:000 para editá-lo e atribuir um tempo para que o ponto de sinalização ocorra. Você pode especificar o tempo em horas, minutos, segundos e milésimos de segundos (HH:MM:SS.mmm).
Se houver vários pontos de sinalização, a caixa de diálogo move o novo ponto de sinalização para a respectiva posição cronológica na lista.
- 5 Para adicionar um parâmetro ao ponto de sinalização selecionado, clique no sinal de adição (+) acima da seção Parâmetros e insira valores nas colunas Nome e Valor. Repita essa etapa para cada parâmetro.
- 6 Para adicionar mais pontos de sinalização do ActionScript, repita as etapas de 2 a 5 para cada um.
- 7 Clique em OK para salvar as alterações.

Excluir um ponto de sinalização do ActionScript :

- 1 Clique duas vezes na célula Valor, do parâmetro `cuePoints`, no Inspetor de componentes, para abrir a caixa de diálogo Pontos de sinalização Flash.
- 2 Selecione o ponto de sinalização que deseja excluir.
- 3 Clique no sinal de subtração (-) do canto superior esquerdo, acima da lista de pontos de sinalização, para excluí-lo.
- 4 Repita as etapas 2 e 3 para cada ponto de sinalização que desejar excluir.
- 5 Clique em OK para salvar as alterações.

Para ativar ou desativar o ponto de sinalização do arquivo FLV incorporado:

- 1 Clique duas vezes na célula Valor, do parâmetro `cuePoints`, no Inspetor de componentes, para abrir a caixa de diálogo Pontos de sinalização Flash.
- 2 Selecione o ponto de sinalização que deseja ativar ou desativar.
- 3 Clique no, valor na coluna Tipo, para acionar o menu pop-up, ou clique na seta para baixo.
- 4 Clique no nome do tipo de ponto de sinalização (por exemplo, Evento ou Navegação) para ativá-lo. Clique em Desativado para desativá-lo.
- 5 Clique em OK para salvar as alterações.

Uso de pontos de sinalização com o ActionScript

Você pode usar o ActionScript para adicionar pontos de sinalização do ActionScript, monitorar o evento `cuePoint`, localizar pontos de sinalização de qualquer tipo ou de um tipo específico, buscar um ponto de sinalização de navegação, ativar ou desativar um ponto de sinalização, verificar se um ponto de sinalização está ativado e remover um ponto de sinalização.

Os exemplos nesta seção usam um arquivo FLV chamado `cuepoints.flv`, que contém os três seguintes pontos de sinalização:

| Nome | Tempo | Tipo |
|---------|--------------|-----------|
| ponto 1 | 00:00:00.418 | Navegação |
| ponto 2 | 00:00:07.748 | Navegação |
| ponto 3 | 00:00:16.020 | Navegação |

Adição de pontos de sinalização do ActionScript

Você pode adicionar pontos de sinalização do ActionScript a um arquivo FLV usando o método `addASCuePoint()`. O exemplo a seguir adiciona dois pontos de sinalização do ActionScript ao arquivo FLV quando ele está pronto para reprodução. Ele adiciona o primeiro ponto de sinalização usando um objeto de ponto de sinalização que especifica tempo, nome e tipo do ponto de sinalização nas respectivas propriedades. A segunda chamada especifica o tempo e o nome usando os parâmetros `time` e `name` do método.

```
// 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");
```

Para obter mais informações, consulte a propriedade Método `FLVPlayback.addASCuePoint()` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

Monitoração de eventos cuePoint

O evento `cuePoint` permite receber controle no código do ActionScript quando ocorre um evento `cuePoint`. Quando ocorrem pontos de sinalização no exemplo seguinte, o ouvinte de `cuePoint` chama uma função do manipulador de eventos que exibe o valor da propriedade `playheadTime` e o nome e o tipo do ponto de sinalização. Use este exemplo juntamente com o exemplo da seção anterior, Adicionar pontos de sinalização do ActionScript, para ver os resultados.

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

Para obter mais informações sobre o evento `cuePoint`, consulte o evento `Evento FLVPlayback.cuePoint` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

Localizar pontos de sinalização

Com o uso do ActionScript, você pode localizar um ponto de sinalização de qualquer tipo, localizar o ponto de localização mais próximo de um tempo ou localizar o próximo ponto de sinalização com um nome específico.

O manipulador de eventos `ready_listener()` no exemplo a seguir chama o método `findCuePoint()` para localizar o ponto de sinalização `ASpt1` e, em seguida, chama o método `findNearestCuePoint()` para localizar o ponto de sinalização de navegação mais próximo do tempo do ponto de sinalização `ASpt1`:

```
import fl.video.FLVPlayback;
import fl.video.CuePointType;
import fl.video.VideoEvent;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var rtn_obj:Object; //create cue point object
my_FLVPlybk.addASCuePoint(2.02, "ASpt1");//add AS cue point
function ready_listener(eventObject:VideoEvent):void {
    rtn_obj = my_FLVPlybk.findCuePoint("ASpt1", CuePointType.ACTIONSCRIPT);
    traceit(rtn_obj);
    rtn_obj = my_FLVPlybk.findNearestCuePoint(rtn_obj.time, CuePointType.NAVIGATION);
    traceit(rtn_obj);
}
my_FLVPlybk.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);
}
```

No exemplo a seguir, o manipulador de eventos `ready_listener()` localiza o ponto de sinalização `ASpt` e chama o método `findNextCuePointWithName()` para localizar o próximo ponto de sinalização com o mesmo nome:

Uso do componente FLVPlayback

```

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

```

Para obter mais informações sobre como localizar pontos de sinalização, consulte os métodos Métodos FLVPlayback.findCuePoint(), FLVPlayback.findNearestCuePoint() e FLVPlayback.findNextCuePointWithName() na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

Busca de pontos de sinalização de navegação

Você pode buscar um ponto de sinalização de navegação, buscar o próximo ponto de sinalização de navegação de um tempo especificado e buscar o ponto de sinalização de navegação anterior em um tempo especificado. O exemplo a seguir reproduz o arquivo FLV cuepoints.flv e busca o ponto de sinalização em 7.748 quando o ocorre o evento ready. Quando ocorre o evento cuePoint, o exemplo chama o método seekToPrevNavCuePoint() para buscar o primeiro ponto de sinalização. Quando esse evento cuePoint ocorre, o exemplo chama o método seekToNextNavCuePoint() para buscar o último ponto de sinalização, adicionando 10 segundos a eventObject.info.time, que é o tempo do ponto de sinalização atual.

```

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";

```

Para obter mais informações, consulte a propriedade Métodos FLVPlayback.seekToNavCuePoint(), FLVPlayback.seekToNextNavCuePoint() e FLVPlayback.seekToPrevNavCuePoint() na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

Uso do componente FLVPlayback**Ativação e desativação de pontos de sinalização do arquivo FLV incorporado**

Você pode ativar ou desativar pontos de sinalização do arquivo FLV incorporado usando o método `addASCuePoint()`. Os pontos de sinalização desativados não acionam eventos `cuePoint` e não funcionam com métodos `seekToCuePoint()`, `seekToNextNavCuePoint()` ou `seekToPrevNavCuePoint()`. No entanto, você pode encontrar pontos de sinalização desativados, com os métodos `findCuePoint()`, `findNearestCuePoint()` e `findNextCuePointWithName()`.

Você pode testar se um ponto de sinalização do arquivo FLV incorporado está ativado, usando o método `isFLVCuePointEnabled()`. O exemplo a seguir desativa os pontos de sinalização incorporados `ponto 2` e `ponto 3` quando o vídeo estiver pronto para reprodução. No entanto, quando o primeiro evento `cuePoint` ocorre, o manipulador de eventos testa para observar se o ponto de sinalização `ponto 3` está desativado e, caso esteja, ativa-o.

```
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");
    }
}
```

Para obter mais informações, consulte a propriedade Métodos `FLVPlayback.isFLVCuePointEnabled()` e `FLVPlayback.setFLVCuePointEnabled()` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

Remover um ponto de sinalização do ActionScript

Você pode remover um ponto de sinalização do ActionScript usando o método `removeASCuePoint()`. O exemplo a seguir remove o ponto de sinalização `ASpt2` quando ocorre o ponto de sinalização `ASpt1`:

```
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv";
my_FLVPlybk.addASCuePoint(2.02, "ASpt1");//add AS cue point
my_FLVPlybk.addASCuePoint(3.4, "ASpt2");//add 2nd ASpt
my_FLVPlybk.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace("Cue point name is: " + eventObject.info.name);
    if (eventObject.info.name == "ASpt1") {
        my_FLVPlybk.removeASCuePoint("ASpt2");
        trace("Removed cue point ASpt2");
    }
}
```

Para obter mais informações, consulte `FLVPlayback.removeASCuePoint()` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

Reproduzir vários arquivos de vídeo

Você pode reproduzir arquivos de vídeo seqüencialmente em uma ocorrência FLVPlayback; basta carregar a nova URL na propriedade `source` quando terminar a reprodução do arquivo de vídeo anterior. Por exemplo, o código do ActionScript a seguir monitora o evento `complete`, que ocorre quando termina a reprodução de um arquivo de vídeo. Quando esse evento ocorre, o código define nome e local de um novo arquivo de vídeo na propriedade `source` e chama o método `play()` para reproduzir o novo vídeo.

```
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");
    }
};
```

Usar vários players de vídeo

Você também pode abrir vários players de vídeo em uma única ocorrência do componente FLVPlayback para reproduzir vários vídeos e alternar entre eles, conforme são reproduzidos.

Você cria o player de vídeo inicial arrastando o componente FLVPlayback para o Palco. O componente atribui automaticamente o número 0 ao player de vídeo inicial e o torna o player padrão. Para criar um player de vídeo adicional, basta definir a propriedade `activeVideoPlayerIndex` para um novo número. Definir a propriedade `activeVideoPlayerIndex` também torna o player de vídeo especificado o player de vídeo *active*, que é aquele que será afetado pelas propriedades e métodos da classe FLVPlayback. Porém, definir a propriedade `activeVideoPlayerIndex` não torna o player de vídeo visível. Para fazer com que o player de vídeo fique visível, defina a propriedade `visibleVideoPlayerIndex` com o número do player de vídeo. Para obter mais informações sobre como essas propriedades interagem com os métodos e as propriedades da classe FLVPlayback, consulte as propriedades Propriedades FLVPlayback.`activeVideoPlayerIndex` e FLVPlayback.`visibleVideoPlayerIndex` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

O código do ActionScript a seguir carrega a propriedade `source` para reproduzir um arquivo de vídeo no player de vídeo padrão e adiciona um ponto de sinalização a ele. Quando o evento `ready` ocorre, o manipulador de eventos abre um segundo player de vídeo, definindo a propriedade `activeVideoPlayerIndex` para o número 1. Ele especifica um arquivo FLV e um ponto de sinalização para o segundo player de vídeo e, em seguida, torna novamente padrão o player de vídeo ativo (0).

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

Uso do componente FLVPlayback

Para alternar para outro arquivo FLV durante a reprodução de um arquivo FLV, você deve fazer a alternância no código do ActionScript. Os pontos de sinalização permitem interferir em pontos específicos do arquivo FLV usando um evento `cuePoint`. O código a seguir cria um ouvinte para o evento `cuePoint` e chama a função do manipulador que pausa o player de vídeo ativo (0), alterna para o segundo player (1) e reproduz o respectivo arquivo 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);
    }
};
```

Quando você cria um novo player de vídeo, a ocorrência `FLVPlayback` define as respectivas propriedades com o valor do player de vídeo padrão, exceto nas propriedades `source`, `totalTime` e `isLive`, cujas ocorrências do `FLVPlayback` são sempre definidas para os valores padrão: string vazia, 0 e `false`, respectivamente. Ele define a propriedade `autoPlay`, que assume `true` como o player de vídeo padrão, para `false`. A propriedade `cuePoints` não tem efeito e também não tem efeito em uma carga subsequente no player de vídeo padrão.

Os métodos e as propriedades que controlam volume, posicionamento, dimensões, visibilidade e controles da interface do usuário são sempre globais e os respectivos comportamentos não são afetados pela definição da propriedade `activeVideoPlayerIndex`. Para obter mais informações sobre esses métodos e propriedades e o efeito de definição da propriedade `activeVideoPlayerIndex`, consulte a propriedade `Propriedade FLVPlayback.activeVideoPlayerIndex` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#). As propriedades e os métodos restantes direcionam o player de vídeo identificado pelo valor da propriedade `activeVideoPlayerIndex`.

No entanto, as propriedades e os métodos que controlam dimensões *interagem* com a propriedade `visibleVideoPlayerIndex`. Para obter mais informações, consulte a propriedade `Propriedade FLVPlayback.visibleVideoPlayerIndex` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

Transmitir arquivos FLV do Flash Media Server

Os requisitos para transmitir arquivos FLV do Flash Media Server são diferentes, dependendo de a detecção de largura de banda nativa estar disponível em seu provedor FVSS (Flash Video Streaming Service). A detecção de largura de banda nativa significa que a detecção dessa largura é incorporada no servidor de fluxo e propicia melhor desempenho. Verifique com seu provedor para determinar se a detecção de largura de banda nativa está disponível.

Para acessar os arquivos FLV no Flash Media Server, use uma URL, como `rtmp://meu_nome_do_servidor/meu_aplicativo/fluxo.flv`.

Ao reproduzir um fluxo ao vivo com o Flash Media Server, é necessário definir a propriedade `isLive` do `FLVPlayback` para `true`. Para obter mais informações, consulte a propriedade Propriedade `FLVPlayback.isLive` na [Referência do ActionScript 3.0 para Adobe Flash Platform](#).

Para obter mais informações sobre como administrar o Flash Media Server, incluindo como configurar um fluxo ao vivo, consulte a documentação do Flash Media Server em www.adobe.com/support/documentation/en/flashmediaserver/.

Para detecção de largura de banda nativa ou nenhuma detecção de largura de banda

A classe `NCManagerNative` é uma subclasse de `NCManager` compatível com a detecção de largura de banda nativa para a qual alguns provedores do FVSS (Flash Video Streaming Service) podem oferecer suporte. Quando você usa o `NCManagerNative`, não é necessário nenhum arquivo especial no Flash Media Server. A `NCManagerNative` também permite conexão com qualquer versão do Flash Media Server, sem um arquivo `main.asc`, se a detecção da largura de banda não for obrigatória.

Para usar a `NCManagerNative` em vez da classe `NCManager` padrão, adicione as seguintes linhas de código ao primeiro quadro de seu arquivo FLA:

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

Para detecção de largura de banda não nativa

Se a detecção de largura de banda nativa não estiver disponível no seu provedor FVSS (Flash Video Streaming Service) mas você precisar de detecção de largura de banda, será necessário adicionar o arquivo `main.asc` a seu aplicativo FLV do Flash Media Server. Você pode encontrar o arquivo `main.asc` on-line em www.adobe.com/go/learn_fl_samples_br. Ele se encontra no arquivo `Samples.zip` -- no diretório `Samples\ComponentsAS2\FLVPlayback`.

Para configurar seu Flash Media Server para transmitir em fluxo arquivos FLV:

- 1 Crie uma pasta na pasta do aplicativo do Flash Media Server e dê um nome a ela, como `meu_aplicativo`.
- 2 Copie o arquivo `main.asc` na pasta `meu_aplicativo`.
- 3 Crie uma pasta chamada `streams` na pasta `meu_aplicativo`.
- 4 Crie uma pasta chamada `_definst_` dentro da pasta `streams`.
- 5 Coloque os arquivos FLV na pasta `_definst_`.

Personalizar o componente FLVPlayback

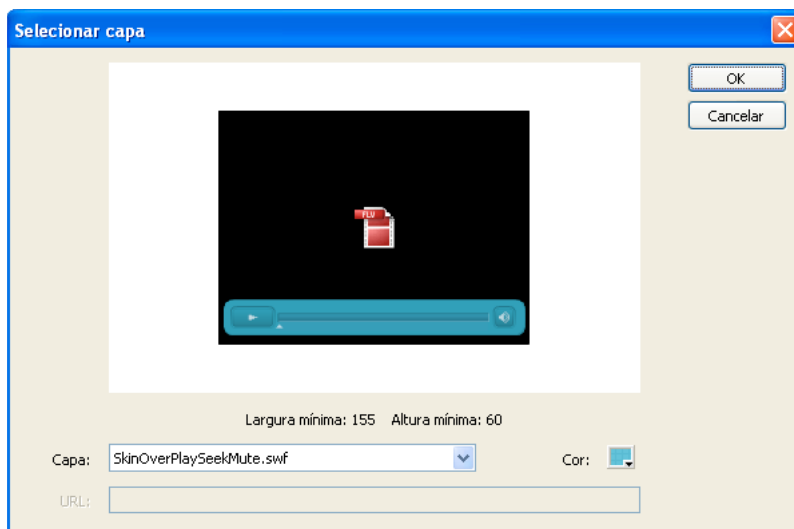
Esta seção explica como personalizar o componente `FLVPlayback`. A maioria dos métodos usados para personalizar outros componentes, no entanto, não funciona com o componente `FLVPlayback`. Para personalizar o componente `FLVPlayback`, use apenas as técnicas descritas nesta seção.

Você tem as seguintes opções de personalização do componente FLVPlayback: selecione uma capa previamente criada, componentes de capa de UI personalizada de reprodução FLV individualmente, ou crie uma nova capa. Você também pode usar as propriedades do FLVPlayback para modificar o comportamento de uma capa.

Nota: Você deve enviar seu arquivo de capa SWF para o servidor da Web, juntamente com o arquivo de aplicativo SWF, para que a capa funcione com seu componente FLVPlayback.

Selecionar uma capa previamente criada

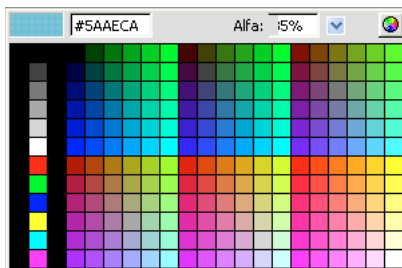
Você pode selecionar uma capa para o componente FLVPlayback, clicando na célula `valor` do parâmetro `skin` no Inspetor de componentes. Em seguida, clique no ícone de lupa para abrir a caixa de diálogo seguinte Selecionar capa, em que você pode selecionar uma capa ou fornecer uma URL que especifique o local do arquivo de capa SWF.



Caixa de diálogo Selecionar capa FLVPlayback

As capas listadas no menu pop-up Capa estão localizadas na pasta do aplicativo Flash/Configuração do Flash/Capas FLVPlayback/ActionScript 3.0. Você pode tornar novas capas disponíveis nesta caixa de diálogo, criando-as e colocando o arquivo SWF na pasta. O nome é exibido no menu pop-up com a extensão `.swf`. Para obter mais informações sobre como criar um conjunto de capas, consulte “[Criar uma nova capa](#)” na página 164.

Para capas que você atribui definindo a propriedade `skin`, definindo o parâmetro da capa durante a criação ou com o ActionScript em tempo de execução, você pode atribuir valores de cor e alfa (transparência) independentemente da capa escolhida. Para atribuir os valores de cor e alfa durante a criação, abra o seletor de cores da caixa de diálogo Selecionar capa, conforme mostrado aqui.



Seletor de cores na caixa de diálogo Selecionar capa

Uso do componente FLVPlayback

Para selecionar a cor, clique em uma amostra no painel ou insira o valor numérico correspondente na caixa de texto. Para selecionar o valor alfa, use o controle deslizante ou digite um percentual na caixa de texto Alfa.

Para atribuir os valores de cor e alfa durante o tempo de execução, defina as propriedades `skinBackgroundColor` e `skinBackgroundAlpha`. Defina a propriedade `skinBackgroundColor` com um valor `0xRRGGBB` (vermelho, verde e azul). Defina a propriedade `skinBackgroundAlpha` para um número entre 0,0 e 1,0. O exemplo a seguir define `skinBackgroundColor` para `0xFF0000` (vermelho) e `skinBackgroundAlpha` para `.5`.

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

Os valores padrão são os últimos valores selecionados pelo usuário.

Se você deseja colocar uma capa no componente FLVPlayback usando os componentes de UI personalizada de reprodução FLV, selecione Nenhum, no menu pop-up.

Colocar capa em componentes de UI personalizada de reprodução FLV individualmente

Os componentes de UI personalizada de reprodução FLV permitem personalizar a aparência dos controles FLVPlayback em seu arquivo FLA e permitem que você veja os resultados ao visualizar sua página da Web. No entanto, esses componentes não foram projetados para ser dimensionados. Você deve editar o clipe de filme e o respectivo conteúdo para se de um tamanho específico. Por essa razão, normalmente é melhor ter o componente FLVPlayback no Palco no tamanho desejado, com `scaleMode` definido para `exactFit`.

Para começar, basta arrastar os componentes de UI personalizada de reprodução FLV desejados, do painel Componentes, colocá-los no lugar onde quiser, no Palco, e dar a eles nomes de ocorrências.

Esses componentes podem funcionar sem nenhum ActionScript. Se você os colocar na mesma Linha do tempo e no mesmo quadro do componente FLVPlayback e não houver nenhuma capa definida no componente, o componente FLVPlayback se conectará a eles automaticamente. Se você tem vários componentes FLVPlayback no Palco ou se o controle personalizado e a ocorrência FLVPlayback não estiverem na mesma Linha do tempo, será necessário o Action.

Assim que os componentes estiverem no Palco, você poderá editá-los como qualquer outro símbolo. Após abrir os componentes, você poderá observar que cada um está configurado de modo um pouco diferente dos demais.

Componentes Button

Os componentes button têm uma estrutura semelhante. Eles incluem BackButton, ForwardButton, MuteButton, PauseButton, PlayButton, PlayPauseButton e StopButton. A maioria tem um único clipe de filme no Quadro 1 com nome de ocorrência `placeholder_mc`. Normalmente é uma ocorrência do estado normal do button, mas não necessariamente. No Quadro 2 há quatro clipes de filme no Palco para cada estado de exibição: normal, sobre, abaixo e desativado. (No tempo de execução, o componente nunca vai efetivamente para o Quadro 2; esses clipes de filme são colocados aqui para tornar a edição mais conveniente e para forçá-los a serem carregados no arquivo SWF sem selecionar Exportar, na caixa de seleção Primeiro quadro, da caixa de diálogo Propriedades do símbolo. No entanto, você ainda deve selecionar a opção Exportar para ActionScript.)

Para colocar capa no button, basta editar cada um desses clipes de filme. Você pode alterar o tamanho deles, bem como a aparência.

Alguns ActionScripts normalmente são exibidos no Quadro 1. Você não terá que alterar esse script. Ele pára, simplesmente, o indicador de reprodução do Quadro 1, e especifica que clipes de filme usar e em que estados.

Buttons PlayPauseButton, MuteButton, FullScreenButton e CaptionButton

Os buttons PlayPauseButton, MuteButton, FullScreenButton e CaptionButton são configurados diferentemente dos outros; eles têm apenas um quadro com duas camadas e nenhum script. Nesse quadro há dois buttons, um em cima do outro; no caso de PlayPauseButton, um botão Reproduzir e Pausar; no caso de MuteButton, um botão Mudo ligado e Mute desligado; no caso de FullScreenButton, um button de tela cheia ligada e tela cheia desligada; no caso de CaptionButton, um button legenda ligada e legenda desligada. Para colocar capa nesses buttons, coloque capa em cada um desses dois buttons internos conforme descrito em [“Colocar capa em componentes de UI personalizada de reprodução FLV individualmente”](#) na página 159; não é necessária nenhuma ação adicional.

O CaptionButton é para o componente FLVPlaybackCaptioning e deve ser anexado a esse componente, não ao componente FLVPlayback.

Buttons BackButton e ForwardButton

Os buttons BackButton e ForwardButton também são configurados diferentemente dos outros. No Quadro 2, eles têm clipes de filme extras que você pode usar como um quadro ao redor de um ou dos dois buttons. Esses clipes de filme não são requeridos e não têm recursos especiais; eles são fornecidos apenas por conveniência. Para usá-los, basta arrastá-los para o Palco do painel Biblioteca e colocá-los onde desejar. Se você não os desejar, não os utilize ou exclua-os do painel Biblioteca.

A maioria dos buttons, conforme fornecidos, se baseiam em um conjunto comum de clipes de filme, de modo que você possa alterar a aparência de todos buttons de uma vez. Você pode usar esse recurso ou substituir esses clipes comuns e fazer com que cada button tenha uma aparência diferente.

Componente BufferingBar

O componente de barra de buffer é simples: Ele consiste em uma animação que se torna visível quando o componente entra no estado de buffer e não requer nenhum ActionScript especial para configurá-lo. Por padrão, é uma barra listrada que se move da esquerda para a direita com uma máscara retangular para dar a ela um efeito de poste de barbearia, mas não há nada de especial sobre essa configuração.

Embora as barras de buffer nos arquivos de capa SWF usem o dimensionamento de 9 fatias, porque precisam ser dimensionadas em tempo de execução, o componente de UI personalizada FLV BufferingBar não usa e *não pode* usar o dimensionamento de 9 fatias, por ter clipes de filme aninhados. Se você desejar tornar a BufferingBar mais larga ou mais alta, pode ser conveniente alterar o conteúdo dela, em vez de dimensioná-la.

Componentes SeekBar e VolumeBar

Os componentes SeekBar e VolumeBar são semelhantes, embora tenham funções diferentes. Cada um tem indicadores, usam os mesmos mecanismos de controle de indicadores e oferecem suporte a clipes aninhados para controlar progresso e integridade.

Há muitos locais em que o código ActionScript do no componente FLVPlayback considera que o ponto de registro (também *ponto de origem* ou *ponto zero*) de seu componente SeekBar ou VolumeBar esteja no canto superior esquerdo do conteúdo, portanto, é importante manter essa convenção. Do contrário, você pode ter problemas com indicadores e com o progresso e a integridade de clipes de filme.

Embora as barras de busca nos arquivos de capa SWF usem o dimensionamento de 9 fatias porque precisam ser dimensionadas em tempo de execução, o componente de UI personalizada FLV SeekBar não usa e *não pode* usar o dimensionamento de 9 fatias, por ter clipes de filme aninhados. Se você desejar tornar a SeekBar mais larga ou mais alta, pode ser conveniente alterar o conteúdo dela, em vez de dimensioná-la.

Indicador

Uma ocorrência do clipe de filme de indicador está no Quadro 2. Assim como os componentes BackButton e ForwardButton, o componente nunca vai efetivamente para o Quadro 2; esses clipes de filme são colocados aqui para tornar a edição mais conveniente e como forma de forçá-los a serem carregados no arquivo SWF sem selecionar Exportar, na caixa de seleção Primeiro quadro, da caixa de diálogo Propriedades do símbolo. No entanto, você ainda deve selecionar a opção Exportar para ActionScript.

Você pode perceber que o clipe de filme de indicador tem um retângulo no plano de fundo com alfa definido para 0. Esse retângulo aumenta o tamanho da área de ocorrência da alça, tornando mais fácil segurar sem alterar sua aparência, semelhante ao estado de ocorrência de um botão. Como o indicador é criado dinamicamente em tempo de execução, ele deve ser um clipe de filme e não um botão. Esse retângulo com alfa definido para 0 não é necessário por nenhum outro motivo e, geralmente, você pode substituir o interior do indicador com qualquer imagem que desejar. No entanto, ele funciona melhor para manter o ponto de registro centralizado horizontalmente no meio do clipe de filme de indicador.

O código ActionScript seguinte encontra-se no Quadro 1 do componente SeekBar para gerenciar o indicador:

```
stop();  
handleLinkageID = "SeekBarHandle";  
handleLeftMargin = 2;  
handleRightMargin = 2;  
handleY = 11;
```

A chamada para a função `stop()` é necessária devido ao conteúdo do Quadro 2.

A segunda linha especifica que símbolo usar como indicador, e você não precisará fazer alterações se simplesmente editar a ocorrência de clipe de filme de indicador no Quadro 2. Em tempo de execução, o componente FLVPlayback cria uma ocorrência do clipe de filme especificado no Palco como um irmão da ocorrência do componente Bar, o que significa que eles têm o mesmo clipe de filme pai. Portanto, se sua barra estiver no nível raiz, o identificador também deverá estar nesse nível.

A variável `handleLeftMargin` determina a localização original do indicador (0%) e a variável `handleRightMargin` determina onde ela está no final (100%). Os números fornecem os deslocamentos das extremidades esquerda e direita da barra de controle, com números positivos marcando os limites dentro da barra e números negativos marcando os limites fora da barra. Esses deslocamentos especificam onde o indicador pode ir, com base no respectivo ponto de registro. Se você colocar o ponto de registro no meio do indicador, as extremidades esquerda e direita passarão das margens. O clipe de filme de barra de busca deve ter seu ponto de registro como o canto superior esquerdo do respectivo conteúdo, para funcionar corretamente.

A variável `handleY` determina a posição `y` do indicador em relação à ocorrência da barra. Isso tem como base os pontos de registro de cada clipe de filme. O ponto de registro no indicador de amostra está na ponta do triângulo para colocá-lo relativo à parte visível, desconsiderando o retângulo de estado de ocorrência invisível. Além disso, o clipe de filme de barra deve manter seu ponto de registro como o canto superior esquerdo do conteúdo, para funcionar corretamente.

Portanto, por exemplo, com esses limites, se a barra de controle está definida em (100, 100) e tem 100 pixels de largura, o indicador pode variar horizontalmente de 102 a 198 e permanecer verticalmente em 111. Se você alterar `handleLeftMargin` e `handleRightMargin` para -2 e `handleY` para -11, o indicador pode variar horizontalmente de 98 a 202 e permanecer verticalmente em 89.

Progresso e integridade de clipes de filme

O componente SeekBar tem um clipe de filme *progress* e o VolumeBar tem um clipe de filme *fullness*, mas, na prática, qualquer SeekBar ou VolumeBar pode ter um, nenhum ou ambos os clipes de filme. Em termos de estrutura, eles são os mesmos e se comportam de maneira semelhante, mas controlam valores diferentes. O clipe de filme de progresso é preenchido conforme o download do arquivo FLV (que é útil somente em um download HTTP, porque ele está sempre completo, se estiver transmitindo em fluxo contínuo de FMS), e o clipe de filme de integridade é preenchido à medida que o indicador se move da esquerda para a direita.

O componente FLVPlayback localiza essas ocorrências de clipe de filme procurando por um nome de ocorrência específico, portanto, sua ocorrência de clipe de filme de progresso deve ter seu clipe de filme de barra como pai e ter o nome de ocorrência `progress_mc`. A ocorrência de clipe de filme de integridade deve ter o nome de ocorrência `fullness_mc`.

Você pode definir os clipes de filme de progresso e integridade com ou sem a ocorrência de clipe de filme `fill_mc` aninhada internamente. O clipe de filme `fullness_mc` da VolumeBar mostra o método `com` o clipe de filme `fill_mc`, e o clipe de filme `progress_mc` da SeekBar mostra o método `sem` o clipe de filme `fill_mc`.

O método `com` o clipe de filme `fill_mc` aninhado internamente é útil quando se deseja um preenchimento que não pode ser dimensionado sem distorcer a aparência.

No clipe de filme `fullness_mc` da VolumeBar, a ocorrência do clipe de filme `fill_mc` está com máscara. Você também pode colocar máscara nela quando criar o clipe de filme ou uma máscara será criada dinamicamente em tempo de execução. Se você colocar máscara nela com um clipe de filme, nomeie a ocorrência `mask_mc` e configure-a de forma que `fill_mc` seja exibido como seria se o percentual fosse 100%. Se você não colocar máscara em `fill_mc`, a máscara criada dinamicamente será retangular e terá o mesmo tamanho que `fill_mc` em 100%.

O clipe de filme `fill_mc` com a máscara é revelado de uma das duas maneiras, dependendo de `fill_mc.slideReveal` ser `true` ou `false`.

Se `fill_mc.slideReveal` for `true`, `fill_mc` se move da esquerda para a direita para expô-lo através da máscara. Em 0%, ele está inteiramente para a esquerda, portanto, nenhuma parte é mostrada através da máscara. Conforme o percentual aumenta, ele se move para a esquerda até 100%, de volta onde foi criado no Palco.

Se `fill_mc.slideReveal` for `false` ou indefinido (o comportamento padrão), a máscara será redimensionada da esquerda para a direita para revelar mais de `fill_mc`. Quando ele está em 0%, a máscara será dimensionada horizontalmente para 05 e, conforme o percentual aumenta, a `scaleX` aumenta até 100% e revela completamente `fill_mc`. Não é necessariamente `scaleX = 100` porque `mask_mc` pode ter sido dimensionado quando foi criado.

O método `sem` `fill_mc` é mais simples do que o método com `fill_mc`, mas ele distorce o preenchimento horizontalmente. Se você não desejar essa distorção, deverá usar `fill_mc`. O `progress_mc` da SeekBar ilustra esse método.

O clipe de filme de progresso ou integridade é dimensionado horizontalmente, com base no percentual. Em 0%, a `scaleX` da ocorrência é definida como 0, tornando-a invisível. Conforme o percentual cresce, a `scaleX` é ajustada até 100%, o clipe fica com o mesmo tamanho que tinha no Palco, quando foi criado. Mais uma vez, não é necessariamente `scaleX = 100`, porque a ocorrência de clipe pode ter sido dimensionada quando foi criada.

Conecte os componentes da UI personalizada de reprodução FLV

Se você colocar os componentes da UI na mesma linha do tempo e quadro do componente FLVPlayback e não tiver definido a propriedade `skin`, o FLVPlayback se conectará automaticamente a eles sem necessidade de nenhum ActionScript.

Se você tiver vários componentes FLVPlayback no Palco ou se o controle personalizado e o FLVPlayback não estiverem na mesma Linha do tempo, você deve escrever um código ActionScript para conectar os componentes da UI personalizada à ocorrência do componente FLVPlayback. Primeiramente, você deve atribuir um nome para a ocorrência FLVPlayback e, em seguida, usar o ActionScript para atribuir as ocorrências de componente da UI personalizada de reprodução FLV às propriedades de FLVPlayback correspondentes. No exemplo seguinte, a ocorrência FLVPlayback é `my_FLVPlaybk`, os nomes de propriedade de FLVPlayback seguem os pontos (.) e as ocorrências de controle da UI personalizada de reprodução FLV ficam à direita dos sinais de igual (=):

```
//FLVPlayback instance = my_FLVPlaybk  
my_FLVPlaybk.playButton = playbtn; // set playButton prop. to playbtn, etc.  
my_FLVPlaybk.pauseButton = pausebtn;  
my_FLVPlaybk.playPauseButton = playpausebtn;  
my_FLVPlaybk.stopButton = stopbtn;  
my_FLVPlaybk.muteButton = mutebtn;  
my_FLVPlaybk.backButton = backbtn;  
my_FLVPlaybk.forwardButton = forbtn;  
my_FLVPlaybk.volumeBar = volbar;  
my_FLVPlaybk.seekBar = seekbar;  
my_FLVPlaybk.bufferingBar = bufbar;
```

As etapas a seguir criam controles personalizados de StopButton, PlayPauseButton, MuteButton e SeekBar:

- 1 Arraste o componente FLVPlayback para o Palco e dê a ele o nome de ocorrência `my_FLVPlaybk`.
- 2 Defina o parâmetro `source` por meio do Inspetor de componentes para <http://www.helpexamples.com/flash/video/cuepoints.flv>.
- 3 Defina o parâmetro Skin para Nenhum.
- 4 Arraste StopButton, PlayPauseButton e MuteButton para o Palco e coloque-os sobre a ocorrência FLVPlayback, empilhando-os verticalmente à esquerda. Dê um nome de ocorrência a cada botão no Inspetor de propriedades (como `my_stopbtn`, `my_playpausebtn` e `my_mutebtn`).
- 5 No painel Biblioteca, abra a pasta Capas FLVPlayback e, em seguida, abra a pasta SquareButton, abaixo dela.
- 6 Selecione o clipe de filme SquareBgDown e clique duas vezes nele para abri-lo no Palco.
- 7 Clique com o botão direito (Windows) ou Ctrl+clique (Macintosh), selecione Todos, no menu, e exclua o símbolo.
- 8 Selecione a ferramenta Oval, desenhe uma forma oval no mesmo local e defina o preenchimento como azul (`#0033FF`).
- 9 No Inspetor de propriedades, defina a largura (W:) como **40** e a altura (H:) como **20**. Defina a coordenada x (X:) como **0,0** e a coordenada y (Y:) como **0,0**.
- 10 Repita as etapas 6 a 8 para SquareBgNormal, mas altere o preenchimento para amarelo (`#FFFF00`).
- 11 Repita as etapas 6 a 8 para SquareBgOver, mas altere o preenchimento para verde (`#006600`).
- 12 Edite os cliques de filme de vários ícones de símbolo nos botões (PauseIcon, PlayIcon, MuteOnIcon, MuteOffIcon e StopIcon). Você pode encontrar esses cliques de filme no painel Biblioteca, em FLV Playback Skins/Label Button/Assets, onde *Label* é o nome do botão, como Reproduzir, Pausar e assim por diante. Siga estas etapas para cada um:
 - a Selecione a opção Selecionar todos.
 - b Altere a cor para vermelho (`#FF0000`).
 - c Dimensione em 300%.
 - d Altere o valor X: local do conteúdo para **7.0** para alterar o posicionamento horizontal do ícone em cada estado de botão.

Uso do componente FLVPlayback

Nota: Alterando o local dessa maneira, você evita abrir cada estado de botão e mover o ícone da ocorrência de clipe de filme.

- 13 Clique na seta Voltar, azul, acima da Linha do tempo, para retornar à Cena 1 do Quadro 1.
- 14 Arraste o componente SeekBar para o Palco e coloque-o no canto inferior direito da ocorrência FLVPlayback.
- 15 No painel Biblioteca, clique duas vezes na SeekBar para abri-la no Palco.
- 16 Dimensione-a para 400%.
- 17 Selecione o contorno e defina a cor como vermelho (#FF0000).
- 18 Clique duas vezes em SeekBarProgress na pasta FLVPlayback Skins/Seek Bar e defina a cor como amarelo (#FFFF00).
- 19 Clique duas vezes em SeekBarHandle, na pasta FLVPlayback Skins/Seek Bar, e defina a cor como vermelho (#FF0000).
- 20 Clique na seta Voltar, azul, acima da Linha do tempo, para retornar à Cena 1 do Quadro 1.
- 21 Selecione a ocorrência SeekBar no Palco e dê a ela o nome de ocorrência **my_seekbar**.
- 22 No painel Ações do Quadro 1 da Linha do tempo, adicione uma instrução `import` às classes de vídeo e atribua os nomes de botão e barra de busca às propriedades FLVPlayback correspondentes, conforme mostrado no seguinte exemplo:

```
import fl.video.*;
my_FLVPlybk.stopButton = my_stopbbtn;
my_FLVPlybk.playPauseButton = my_plypausbbtn;
my_FLVPlybk.muteButton = my_mutebbtn;
my_FLVPlybk.seekBar = my_seekbar;
```
- 23 Pressione Control+Enter para testar o filme.

Criar uma nova capa

A melhor maneira de criar um arquivo de capa SWF é copiar um dos arquivos de capa que vem com o Flash e usá-lo como ponto de partida. Você pode encontrar os arquivos FLA dessas capas na pasta do aplicativo Flash, em Configuration/FLVPlayback Skins/FLA/ActionScript 3.0/. Para disponibilizar seu arquivo de capa SWF finalizado como opção na caixa de diálogo Selecionar capa, coloque-o na pasta Configuration/FLVPlayback Skins/ActionScript 3.0, na pasta do aplicativo Flash ou em uma pasta Configuration/FLVPlayback Skins/ActionScript 3.0 local do usuário.

Como você pode definir a cor da capa independentemente de escolher a capa, não é necessário editar o arquivo FLA para modificar a cor. Se você criar uma capa com uma cor específica e não desejar que ela esteja editável na caixa de diálogo Selecionar capa, defina `this.border_mc.colorMe = false`; no código ActionScript do arquivo de capa FLA. Para obter informações sobre como definir a cor da capa, consulte [“Selecionar uma capa previamente criada”](#) na página 158.

Ao observarmos os arquivos instalados de capa FLA do Flash, pode parecer que algumas coisas no Palco são desnecessárias, mas muitas dessas coisas são colocadas em camadas de guia. Com a visualização ao vivo usando escala 9, você pode visualizar rapidamente qual será a aparência real do arquivo SWF.

As seções seguintes cobrem personalizações e alterações mais complexas nos clipes de filme de SeekBar, BufferingBar e VolumeBar.

Uso do layout de capa

Ao abrir um arquivo de capa FLA do Flash, você encontra os cliques de filme da capa dispostos na Linha do tempo principal. Esses cliques e o código ActionScript que você encontra no mesmo quadro definem como os controles serão dispostos em tempo de execução.

Embora a camada Layout se pareça muito com o modo como a capa parecerá no tempo de execução, o conteúdo dessa camada não estará visível no tempo de execução. Ela só é usada para calcular onde colocar os controles. Os outros controles no Palco são usados em tempo de execução.

Na camada Layout há um alocador de espaço para o componente FLVPlayback chamado `video_mc`. Todos os outros controles estão dispostos em relação ao `video_mc`. Se você iniciar com um dos arquivos FLA do Flash e alterar o tamanho dos controles, provavelmente será possível corrigir o layout movendo esses cliques de alocador de espaço.

Cada um dos cliques de alocador de espaço tem um nome de ocorrência específico. Os nomes dos cliques de alocador de espaço são `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` e `volumeBarHandle_mc`. A parte que é novamente colorida quando você seleciona uma cor de capa é chamada de `border_mc`.

O tipo de clipe usado em um controle não é importante. Em botões, geralmente o clipe de estado normal é usado. Em outros controles, o clipe correspondente é usado, mas só por conveniência. O importante é a localização de *x* (horizontal) e *y* (vertical) e a altura e largura do alocador de espaço.

Você também pode ter quantos cliques adicionais desejar, além dos controles padrão. O único requisito desses cliques é que os respectivos símbolos da biblioteca estejam com a opção Exportar para ActionScript marcada na caixa de diálogo Ligação. Os cliques personalizados na camada Layout podem ter qualquer nome de ocorrência que não sejam os nomes de ocorrência reservados listados acima. O nome de ocorrência só é necessário para definir o ActionScript nos cliques para determinar o layout.

O clipe `border_mc` é especial. Se você definir a propriedade `FlvPlayback.skinAutoHide` como `true`, a capa é exibida quando o mouse estiver sobre o clipe `border_mc`. Isso é importante nas capas que são exibidas fora dos limites do player de vídeo. Para obter informações sobre a propriedade `skinAutoHide`, consulte [“Modificar comportamento de capa”](#) na página 168.

Nos arquivos FLA do Flash, `border_mc` é usado para o cromo e para a borda ao redor dos botões Avançar e Voltar.

O clipe `border_mc` também faz parte da capa que tem alfa e cor alterados pelas propriedades `skinBackgroundAlpha` e `skinBackgroundColor`. Para permitir cor e alfa personalizáveis, o ActionScript no arquivo de capa FLA deve incluir o seguinte:

```
border_mc.colorMe = true;
```

ActionScript e layout de capa

O seguinte código ActionScript se aplica normalmente a todos os controles. Alguns controles têm um ActionScript específico que define comportamento adicional e que é explicado na seção desse controle.

O ActionScript inicial é uma seção extensa que especifica os nomes de classe de cada estado para cada componente. Você pode visualizar todos esses nomes de classe no arquivo `SkinOverAll.fla`. Por exemplo, nos botões Pausar e Reproduzir, o código é semelhante a:

```
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";
```

Os nomes de classe não têm arquivos de classe externos reais, eles são apenas especificados na caixa de diálogo Ligação para todos os clipes de filme na biblioteca.

No componente ActionScript 2.0, havia clipes de filme no Palco que, na realidade, eram usados em tempo de execução. No componente ActionScript 3.0, esses clipes de filme ainda estão no arquivo FLA, mas apenas para tornar a edição conveniente. Agora, eles estão todos nas camadas de guia e não são exportados. Todos os ativos de capas na biblioteca estão definidos para exportar no primeiro quadro. Eles são criados dinamicamente com um código como este, por exemplo:

```
new fl.video.skin.PauseButtonDisabled();
```

Depois dessa seção está o código ActionScript, que define largura e altura mínimas da capa. A caixa de diálogo Selecionar capa mostra esses valores e eles são usados em tempo de execução para impedir que a capa seja dimensionada abaixo do tamanho mínimo. Se você não deseja especificar um tamanho mínimo, deixe como indefinido ou menor ou igual a zero.

```
// 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;
```

Cada alocador de espaço pode ter as seguintes propriedades aplicadas:

| Propriedade | Descrição |
|--------------|---|
| anchorLeft | Booleano. Posiciona o controle em relação ao lado esquerdo da ocorrência FLVPlayback. Assume <code>true</code> como padrão, a menos que <code>anchorRight</code> esteja definido explicitamente como <code>true</code> e ele então assume <code>false</code> como padrão. |
| anchorRight | Booleano. Posiciona o controle em relação ao lado direito da ocorrência FLVPlayback. Assume <code>false</code> como padrão. |
| anchorBottom | Booleano. Posiciona o controle em relação à parte inferior da ocorrência FLVPlayback. Assume <code>true</code> como padrão, a menos que <code>anchorTop</code> esteja definido explicitamente como <code>true</code> e ele então assume <code>false</code> como padrão. |
| anchorTop | Booleano. Posiciona o controle em relação à parte superior da ocorrência FLVPlayback. Assume <code>false</code> como padrão. |

Se as propriedades `anchorLeft` e `anchorRight` forem `true`, o controle é dimensionado horizontalmente em tempo de execução. Se as propriedades `anchorTop` e `anchorBottom` forem `true`, o controle é dimensionado verticalmente em tempo de execução.

Para observar os efeitos dessas propriedades, veja como elas são usadas nas capas de Flash. Os controles `BufferingBar` e `SeekBar` são os únicos que dimensionam, e eles são dispostos um em cima do outro e têm as propriedades `anchorLeft` e `anchorRight` definidas como `true`. Todos os controles à esquerda da `BufferingBar` e `SeekBar` têm `anchorLeft` definido como `true` e todos os controles à direita têm `anchorRight` definido como `true`. Todos os controles têm `anchorBottom` definido como `true`.

Você pode tentar editar os clipes de filme na camada Layout para fazer uma capa onde os controles estão situados, na parte superior, em vez de na parte inferior. Você só precisa mover os controles para cima, em relação a `video_mc`, e definir `anchorTop` igual a `true` em todos os controles.

Barra de buffer

A barra de buffer tem dois clipes de filme: `bufferingBar_mc` e `bufferingBarFill_mc`. Cada posição do clipe no Palco em relação a outro clipe é importante, pois esse posicionamento relativo é mantido. A barra de buffer usa dois clipes, distintos porque o componente dimensiona `bufferingBar_mc`, mas não `bufferingBarFill_mc`.

O clipe `bufferingBar_mc` tem dimensionamento de 9 fatias aplicado, portanto, as bordas não serão distorcidas quando ele for dimensionado. O clipe `bufferingBarFill_mc` é extremamente largo, de modo que seja sempre largo o suficiente sem precisar ser dimensionado. Ele é mascarado automaticamente em tempo de execução para mostrar somente a parte acima do `bufferingBar_mc` esticado. Por padrão, as dimensões exatas da máscara manterão uma margem à esquerda e à direita na `bufferingBar_mc`, com base na diferença entre as posições x (horizontal) de `bufferingBar_mc` e `bufferingBarFill_mc`. Você pode personalizar a posição com o código ActionScript.

Se sua barra de buffer não precisar ser dimensionada ou não usar o dimensionamento de 9 fatias, você pode defini-la como o componente `BufferingBar` da UI personalizada de reprodução FLV. Para obter mais informações, consulte “[Componente BufferingBar](#)” na página 160.

A barra de buffer tem as seguintes propriedades adicionais:

| Propriedade | Descrição |
|--------------------------------|---|
| <code>fill_mc:MovieClip</code> | Especifica o nome de ocorrência do preenchimento da barra de buffer. Assume o padrão <code>bufferingBarFill_mc</code> . |

Barra de busca e barra de volume

A barra de busca também tem dois clipes de filme: `seekBar_mc` e `seekBarProgress_mc`. Cada posição do clipe na camada Layout em relação a outro clipe é importante, pois esse posicionamento relativo é mantido. Embora os dois clipes sejam dimensionados, o `seekBarProgress_mc` não pode ser aninhado dentro de `seekBar_mc` porque este último usa dimensionamento de 9 fatias, que não funciona bem com clipes de filme aninhados.

O clipe `seekBar_mc` tem dimensionamento de 9 fatias aplicado, portanto, as bordas não serão distorcidas quando ele for dimensionado. O clipe `seekBarProgress_mc` também é dimensionado, mas sem distorcer. Ele não usa dimensionamento de 9 fatias por ser um preenchimento, que fica bem quando distorcido.

O clipe `seekBarProgress_mc` funciona sem um `fill_mc`, de modo muito semelhante como funciona o clipe `progress_mc` em componentes da UI personalizada de reprodução FLV. Em outras palavras, ele não é mascarado e é dimensionado horizontalmente. As dimensões exatas do `seekBarProgress_mc` em 100% são definidas pelas margens esquerda e direita no clipe `seekBarProgress_mc`. Essas dimensões são, por padrão, iguais e baseadas na diferença entre as posições x (horizontal) de `seekBar_mc` e `seekBarProgress_mc`. Você pode personalizar as dimensões com o ActionScript no clipe de filme de barra de busca, conforme mostrado no seguinte exemplo:

```
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;
```

Você pode colocar esse código na Linha do tempo do clipe de filme SeekBar ou colocá-lo com o outro código ActionScript na Linha do tempo principal. Se você personalizar com código em vez de modificar o layout, o preenchimento não precisa estar no Palco. Ele só precisa estar na biblioteca, definido para exportação para o ActionScript no Quadro 1 com nome de classe correto.

Como ocorre com o componente SeekBar da UI personalizada de reprodução FLV, é possível criar um clipe de filme de integridade para a barra de busca. Se a barra de busca não precisa ser dimensionada ou se é dimensionada mas não usa dimensionamento de 9 fatias, você pode configurar `progress_mc` ou `fullness_mc` usando qualquer um dos métodos usados em componentes da UI personalizada de reprodução FLV. Para obter mais informações, consulte .

Como a barra de volume nas capas do Flash não é dimensionada, ela é construída da mesma maneira que o componente VolumeBar da UI personalizada de reprodução FLV. Para obter mais informações, consulte os “[Componentes SeekBar e VolumeBar](#)” na página 160. A exceção é que o indicador é implementado de maneira diferente.

Indicadores de Seekbar e VolumeBar

Os indicadores de SeekBar e VolumeBar são colocados na camada Layout próximo à barra. Por padrão, os valores de margem esquerda, margem direita e eixo y são definidos pelas respectivas posições em relação ao clipe de filme da barra. A margem esquerda é definida pela diferença entre a localização de x (horizontal) do indicador e a localização x (horizontal) da barra e a margem direita é igual à margem esquerda. Você pode personalizar esses valores por meio do ActionScript no clipe de filme SeekBar ou VolumeBar. O exemplo seguinte é o mesmo código ActionScript usado em componentes da UI personalizada de reprodução FLV:

```
this.seekBar_mc.handleLeftMargin = 2;  
this.seekBar_mc.handleRightMargin = 2;  
this.seekBar_mc.handleY = 11;
```

Você pode colocar esse código na Linha do tempo do clipe de filme SeekBar ou colocá-lo com o outro código ActionScript na Linha do tempo principal. Se você personalizar com código em vez de modificar o layout, o indicador não precisará estar no Palco. Ele só precisa estar na biblioteca, definido para exportação para o ActionScript no Quadro 1 com nome de classe correto.

Além dessas propriedades, os indicadores são clipes de filme simples, configurados da mesma maneira que nos componentes da UI personalizada de reprodução FLV. Ambos têm planos de fundo retangulares com a propriedade `alpha` definida como 0. Estão presentes apenas para aumentar a região de ocorrência e não são necessários.

Clipes de segundo e primeiro planos

Os clipes de filme `chrome_mc` e `forwardBackBorder_mc` são implementados como clipes de segundo plano.

Dos clipes de filme `ForwardBackBorder`, `ForwardBorder` e `BackBorder` no Palco e o alocador de espaço dos botões Avançar e Voltar, o único que *não* é uma camada de guia é o `ForwardBackBorder`. Ele só está nas capas que efetivamente usam os botões Avançar e Voltar.

O único requisito desses clipes é que eles precisam ser exportados para o ActionScript no Quadro 1 da biblioteca.

Modificar comportamento de capa

As propriedades `bufferingBarHidesAndDisablesOthers` e `skinAutoHide` permitem personalizar o comportamento de sua capa FLVPlayback.

Definir a propriedade `bufferingBarHidesAndDisablesOthers` como `true` faz com que o componente FLVPlayback oculte a SeekBar e o respectivo indicador, além de desativar os botões Reproduzir e Pausar, quando o componente entra no estado de buffer. Isso pode ser útil quando o arquivo FLV for fluxo contínuo de FMS em uma conexão lenta com definição alta da propriedade `bufferTime` (10, por exemplo). Nessa situação, um usuário impaciente pode tentar começar uma busca, clicando nos botões Reproduzir e Pausar, o que pode atrasar a reprodução do arquivo ainda mais. Você pode impedir essa atividade, definindo `bufferingBarHidesAndDisablesOthers` como `true` e desativando o elemento SeekBar e os botões Pausar e Reproduzir enquanto o componente estiver no estado de buffer.

A propriedade `skinAutoHide` só afeta arquivos de capa SWF previamente criados, e não controles criados de componentes da UI personalizada de reprodução FLV. Se definido como `true`, o componente FLVPlayback oculta a capa quando o mouse não estiver sobre a área de exibição. O valor padrão dessa propriedade é `false`.

Usar um arquivo SMIL

Para controlar vários fluxos de várias larguras de banda, a classe `VideoPlayer` usa uma classe auxiliar (`NCManager`) que oferece suporte a um subconjunto de SMIL. O SMIL é usado para identificar a localização do fluxo de vídeo, o layout (largura e altura) do arquivo FLV e os arquivos de origem FLV correspondentes às larguras de bandas diferentes. Ele também pode ser usado para especificar a taxa de bits e a duração do arquivo FLV.

Use o parâmetro `source` ou a propriedade `Propriedade FLVPlayback.source` (ActionScript) para especificar a localização de um arquivo SMIL. Para obter mais informações, consulte a propriedade `FLVPlayback.source` na [Referência do ActionScript 3.0 para Flash Professional](#).

O exemplo a seguir mostra um arquivo SMIL que transmite vários arquivos FLV de largura de banda de um FMS usando 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>
```

A tag `<head>` pode conter as tags `<meta>` e `<layout>`. A tag `<meta>` só oferece suporte ao atributo `base`, que é usado para especificar a URL do vídeo de fluxo contínuo (RTMP de um FMS).

A tag `<layout>` só oferece suporte ao elemento `root-layout`, que é usado para definir os atributos `height` e `width` e, portanto, determina o tamanho da janela em que o arquivo FLV é processado. Esses atributos só aceitam valores em pixel, não percentuais.

No corpo do arquivo SMIL, você pode incluir também um link único para um arquivo de origem FLV ou, se você estiver transmitindo vários arquivos de várias larguras de bandas de um FMS (como no exemplo anterior), será possível usar a tag `<switch>` para listar os arquivos de origem.

As tags `video` e `ref` na tag `<switch>` são sinônimos e podem usar o atributo `src` para especificar arquivos FLV. Além disso, cada um pode usar os atributos `region`, `system-bitrate` e `dur` para especificar a região e banda de largura mínima necessária e a duração do arquivo FLV.

Na tag `<body>`, somente uma ocorrência da tag `<video>`, `<src>` ou `<switch>` é permitida.

O exemplo a seguir mostra um download progressivo de um arquivo FLV único que não usa detecção de largura de banda:

```
<smil>
  <head>
    <layout>
      <root-layout width="240" height="180" />
    </layout>
  </head>
  <body>
    <video src="myvideo.flv" />
  </body>
</smil>
```

<smil>

Disponibilidade

Flash Professional 8.

Uso

```
<smil>
...
child tags
...
</smil>
```

Atributos

Nenhum.

Tags-filha

`<head>`, `<body>`

Tag-mãe

Nenhum.

Descrição

Tag de primeiro nível, que identifica um arquivo SMIL.

Exemplo

O exemplo seguinte mostra um arquivo SMIL especificando três arquivos FLV:

Uso do componente FLVPlayback

```
<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>**Disponibilidade**

Flash Professional 8.

Uso

```
<head>
...
child tags
...
</head>
```

Atributos

Nenhum.

Tags-filha

<meta>, <layout>

Tag-mãe

<smil>

Descrição

Com suporte às tags <meta> e <layout>, especifica a localização e o layout padrão (altura e largura) dos arquivos FLV de origem.

Exemplo

O exemplo a seguir define o layout da raiz de 240 pixels por 180 pixels:

```
<head>
  <meta base="rtmp://myserver/myapp/" />
  <layout>
    <root-layout width="240" height="180" />
  </layout>
</head>
```

<meta>

Disponibilidade

Flash Professional 8.

Uso

```
<meta/>
```

Atributos

base

Tags-filha

```
<layout>
```

Tag-mãe

Nenhum.

Descrição

Contém o atributo `base`, que especifica a localização (URL RTMP) dos arquivos FLV de origem.

Exemplo

O exemplo seguinte mostra uma meta tag de localização base no `myserver`:

```
<meta base="rtmp://myserver/myapp/" />
```

<layout>

Disponibilidade

Flash Professional 8.

Uso

```
<layout>  
...  
child tags  
...  
</layout>
```

Atributos

Nenhum.

Tags-filha

```
<root-layout>
```

Tag-mãe

```
<meta>
```

Descrição

Especifica a largura e altura do arquivo FLV.

Exemplo

O exemplo seguinte especifica o layout de 240 pixels por 180 pixels:

```
<layout>
  <root-layout width="240" height="180" />
</layout>
```

<root-layout>

Disponibilidade

Flash Professional 8.

Uso

```
<root-layout...attributes.../>
```

Atributos

Width, height

Tags-filha

Nenhum.

Tag-mãe

```
<layout>
```

Descrição

Especifica a largura e altura do arquivo FLV.

Exemplo

O exemplo seguinte especifica o layout de 240 pixels por 180 pixels:

```
<root-layout width="240" height="180" />
```

<body>

Disponibilidade

Flash Professional 8.

Uso

```
<body>
...
child tags
...
</body>
```

Atributos

Nenhum.

Tags-filha

<video>, <ref>, <switch>

Tag-mãe

<smil>

Descrição

Contém as tags <video>, <ref> e <switch>, que especificam o nome do arquivo FLV de origem, a largura de banda mínima e a duração do arquivo FLV. Só há suporte para o atributo `system-bitrate` ao usar a tag <switch>. Na tag <body>, somente uma ocorrência da tag <switch>, <video> ou <ref> é permitida.

Exemplo

O exemplo seguinte especifica três arquivos FLV, dois usando a tag `video` e um usando a tag `ref`:

```
<body>
  <switch>
    <ref src="myvideo_cable.flv" dur="3:00.1"/>
    <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
    <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1"/>
  </switch>
</body>
```

<video>

Disponibilidade

Flash Professional 8.

Uso

```
<video...attributes.../>
```

Atributos

`src`, `system-bitrate`, `dur`

Tags-filha

Nenhum.

Tag-mãe

<body>

Descrição

Sinônimo da tag <ref>. Oferece suporte aos atributos `src` e `dur`, que especificam o nome do arquivo FLV de origem e sua duração. O atributo `dur` oferece suporte aos formatos de hora completo (00:03:00:01) e parcial (03:00:01).

Exemplo

O exemplo seguinte define a origem e duração de um vídeo:

```
<video src="myvideo_mdm.flv" dur="3:00.1"/>
```

<ref>

Disponibilidade

Flash Professional 8.

Uso

```
<ref...attributes.../>
```

Atributos

`src`, `system-bitrate`, `dur`

Tags-filha

Nenhum.

Tag-mãe

```
<body>
```

Descrição

Sinônimo da tag `<video>`. Oferece suporte aos atributos `src` e `dur`, que especificam o nome do arquivo FLV de origem e sua duração. O atributo `dur` oferece suporte aos formatos de hora completo (00:03:00:01) e parcial (03:00:01).

Exemplo

O exemplo seguinte define a origem e duração de um vídeo:

```
<ref src="myvideo_cable.flv" dur="3:00.1"/>
```

<switch>

Disponibilidade

Flash Professional 8.

Uso

```
<switch>  
...  
child tags  
...  
</switch/>
```

Atributos

Nenhum.

Tags-filha

`<video>`, `<ref>`

Tag-mãe

```
<body>
```

Descrição

Usado com as tags-filha <video> ou <ref> para listar os arquivos FLV de fluxo de vídeo de várias larguras de banda. A tag <switch> oferece suporte ao atributo `system-bitrate`, que especifica a largura de banda mínima, bem como os atributos `src` e `dur`.

Exemplo

O exemplo seguinte especifica três arquivos FLV, dois usando a tag `video` e um usando a tag `ref`:

```
<switch>
  <ref src="myvideo_cable.flv" dur="3:00.1"/>
  <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
  <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1" />
</switch>
```

Capítulo 7: Uso do componente de legenda FLVPlayback

O componente FLVPlayback permite que você inclua um player de vídeo no aplicativo Adobe Flash CS5 Professional para executar arquivos descarregados Adobe Flash Video (FLV ou F4V) e os arquivos FLV ou F4V de fluxo. Para obter mais informações sobre o componente FLVPlayback, consulte “[Uso do componente FLVPlayback](#)” na página 139.

O componente FLVPlaybackCaptioning permite que você inclua suporte de legenda fechado para o seu vídeo. O componente de legenda suporta o formato XML Timed Text padrão do W3C e inclui os seguintes recursos:

Legendas com pontos de sinalização de eventos incorporados Associe os pontos de sinalização de eventos incorporados de um arquivo FLV ao XML para fornecer legendas, em vez de usar um arquivo XML Timed Text.

Várias legendas FLVPlayback Crie várias instâncias de legenda FLVPlayback para várias ocorrências de FLVPlayback.

Controle do botão de alternância Permita a interação do usuário com legendas através de um botão de alternância de legendas.

Uso do componente FLVPlaybackCaptioning

O componente FLVPlaybackCaptioning é usado com um ou mais componentes FLVPlayback. No cenário mais simples, você arrasta um componente FLVPlayback no palco, arrasta um componente FLVPlaybackCaptioning no mesmo palco, identifica a URL da legenda e define as legendas a serem exibidas. Além disso, você também pode definir vários parâmetros para personalizar as legendas FLVPlayback.

Adição de legendas ao componente FLVPlayback

Você pode adicionar o componente FLVPlaybackCaptioning a qualquer componente FLVPlayback. Para obter informações sobre como adicionar componentes FLVPlayback ao aplicativo, consulte “[Crie um aplicativo com o componente FLVPlayback](#)” na página 141.

Adição do componente FLVPlaybackCaptioning a partir do painel Componentes:

- 1 No painel Componentes, abra a pasta de vídeo.
- 2 Arraste o componente FLVPlaybackCaptioning (ou clique duas vezes nele) e adicione-o ao mesmo palco que o componente FLVPlayback ao qual você deseja adicionar legendas.

Nota: A Adobe oferece dois arquivos para que você se familiarize rapidamente com o componente FLVPlaybackCaptioning: *caption_video.flv* (uma amostra do FLVPlayback) e *caption_video.xml* (uma amostra de legendas). Acesse esses arquivos em www.helpexamples.com/flash/video/caption_video.flv e www.helpexamples.com/flash/video/caption_video.xml.

- 3 (Opcional) Arraste o componente CaptionButton para o mesmo palco que os componentes FLVPlayback e FLVPlaybackCaptioning. O componente CaptionButton permite que um usuário ative e desative as legendas.

Nota: Para ativar o componente CaptionButton, arraste-o para o mesmo palco dos componentes FLVPlayback e FLVPlaybackCaptioning.

4 Com o componente FLVPlaybackCaptioning selecionado no Palco, na aba Parâmetros do Inspetor de propriedades, especifique as seguintes informações necessárias:

- Defina `showCaptions` como `true`.
- Especifique a propriedade `source` do arquivo XML Timed Text a ser baixado.



Enquanto trabalha no Flash para testar as legendas, defina a propriedade `showCaptions` como `true`. Entretanto, se você incluir o componente `CaptionButton` para permitir que os usuários ativem e desativem as legendas, defina a propriedade `showCaptions` como `false`.

Há outros parâmetros disponíveis que podem ajudar você a personalizar o componente FLVPlaybackCaptioning. Para obter mais informações, consulte “[Personalização do componente FLVPlaybackCaptioning](#)” na página 187 e a *Referência do ActionScript 3.0 para Flash Professional*.

5 Selecione Controlar > Testar filme para iniciar o vídeo.

Criar uma ocorrência dinamicamente usando o ActionScript:

- 1 Arraste o componente FLVPlayback do painel Componentes para o painel Biblioteca (Janelas > Biblioteca).
- 2 Arraste o componente FLVPlaybackCaptioning do painel Componentes para o painel Biblioteca.
- 3 Adicione o seguinte código ao painel Ações no Quadro 1 da Linha do tempo.

```
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 Altere `install_drive` para que seja a unidade em que você instalou o Flash e modifique o caminho de modo que ele reflita o local da pasta de capas da instalação:

Nota: Se você criar uma ocorrência de FLVPlayback com o ActionScript, deverá atribuir dinamicamente uma capa a ela, definindo a propriedade `skin` com o ActionScript. Quando você aplica uma capa com o ActionScript, ela não é automaticamente publicada com o arquivo SWF. Copie o arquivo de capa SWF e o arquivo de aplicativo SWF para o servidor. Do contrário, o arquivo de capa não estará disponível quando um usuário decidir executá-lo.

Definição dos parâmetros do componente FLVPlaybackCaptioning

Para cada ocorrência do componente FLVPlaybackCaptioning, você pode definir os seguintes parâmetros no Inspetor de propriedades ou no Inspetor de componentes para personalizar o componente. A lista a seguir identifica e fornece uma breve explicação das propriedades:

autoLayout Determina se o componente FLVPlaybackCaptioning controlará o tamanho da área de legenda. O padrão é `true`.

captionTargetName Identifica o nome de ocorrência TextField ou MovieClip que contém legendas. O padrão é `auto`.

flvPlaybackName Identifica o nome de ocorrência FLVPlayback no qual você deseja inserir as legendas. O padrão é `auto`.

simpleFormatting Limita as instruções de formatação do arquivo XML Timed Text quando definida como true. O padrão é false.

showCaptions Determina se as legendas serão exibidas. O padrão é true.

source Identifica o local do arquivo XML Timed Text.

Para obter mais informações sobre todos os parâmetros do FLVPlaybackCaptioning, consulte a *Referência do ActionScript 3.0 para Flash Professional*.

Especificação do parâmetro source

Use o parâmetro `source` para especificar o nome e o local do arquivo XML Timed Text que contém as legendas do filme. Digite o caminho da URL diretamente na célula de origem no Inspetor de componentes.

Exibição das legendas

Para exibir legendas, defina o parâmetro `showCaptions` como true.

Para obter mais informações sobre todos os parâmetros do componente FLVPlaybackCaptioning, consulte a *Referência do ActionScript 3.0 para Flash Professional*.

Nos exemplos anteriores, você aprendeu a criar e ativar o componente FLVPlaybackCaptioning para exibir legendas. Você pode usar duas origens de legenda: (1) um arquivo XML Timed Text que contém as legendas ou (2) um arquivo XML com o texto das legendas que você associa aos pontos de sinalização de eventos incorporados.

Uso de legendas Timed Text

O componente FLVPlaybackCaptioning ativa as legendas do componente FLVPlayback associado baixando um arquivo XML Timed Text (TT). Para obter mais informações sobre o formato Timed Text, consulte AudioVideo Timed Text em www.w3.org.

Esta seção fornece uma visão geral das tags Timed Text suportadas, as tags de arquivo de legenda necessárias e um exemplo de arquivo XML Timed Text. Para obter informações detalhadas sobre todas as tags Timed Text suportadas, consulte “[Tags Timed Text](#)” na página 180.

O componente FLVPlaybackCaptioning suporta as seguintes tags Timed Text:

| Categoria | Tarefa |
|-----------------------------------|---|
| Suporte à formatação de parágrafo | Alinhamento de um parágrafo à direita, à esquerda ou ao centro |
| Suporte à formatação de texto | <ul style="list-style-type: none">Definição do tamanho do texto com tamanhos de pixel absolutos ou estilo delta (por exemplo, +2, -4)Definição da cor e da fonte do textoAplicação de negrito e itálico ao textoDefinição da justificação do texto |
| Outro suporte à formatação | <ul style="list-style-type: none">Definição da cor de fundo do TextField das legendasDefinição da cor de fundo do TextField das legendas para transparente (alfa 0)Definição da quebra de texto do TextField das legendas (ativado ou desativado) |

O componente FLVPlaybackCaptioning corresponde ao código de tempo do arquivo FLV. Toda legenda deve ter um atributo `begin`, que determina quando a legenda deve ser exibida. Se a legenda não tiver um atributo `dur` ou `end`, ela desaparecerá quando a próxima legenda for exibida ou quando o arquivo FLV chegar ao fim.

Este é um exemplo de arquivo XML Timed Text. Esse arquivo (`caption_video.xml`) oferece legendas para o arquivo `caption_video.flv`. Acesse esses arquivos em www.helpexamples.com/flash/video/caption_video.flv e www.helpexamples.com/flash/video/caption_video.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
  <tt xml:lang="en"
xmlns="http://www.w3.org/2006/04/ttaf1"xmlns:tts="http://www.w3.org/2006/04/ttaf1#styling">
  <head>
    <styling>
      <style id="1" tts:textAlign="right"/>
      <style id="2" tts:color="transparent"/>
      <style id="3" style="2" tts:backgroundColor="white"/>
      <style id="4" style="2 3" tts:fontSize="20"/>
    </styling>
  </head>
  <body>
    <div xml:lang="en">
      <p begin="00:00:00.00" dur="00:00:03.07">I had just joined <span
tts:fontFamily="monospaceSansSerif,proportionalSerif,TheOther"tts:fontSize="+2">Macromedia</span>
in 1996,</p>
      <p begin="00:00:03.07" dur="00:00:03.35">and we were trying to figure out what to do about the
internet.</p>
      <p begin="00:00:06.42" dur="00:00:03.15">And the company was in dire straights at the time.</p>
      <p begin="00:00:09.57" dur="00:00:01.45">We were a CD-ROM authoring company,</p>
      <p begin="00:00:11.42" dur="00:00:02.00">and the CD-ROM business was going away.</p>
      <p begin="00:00:13.57" dur="00:00:02.50">One of the technologies I remember seeing was
Flash.</p>
      <p begin="00:00:16.47" dur="00:00:02.00">At the time, it was called <span
tts:fontWeight="bold" tts:color="#ccc333">FutureSplash</span>.</p>
      <p begin="00:00:18.50" dur="00:00:01.20">So this is where Flash got its start.</p>
      <p begin="00:00:20.10" dur="00:00:03.00">This is smart sketch running on the <span
tts:fontStyle="italic">EU-pin computer</span>,</p>
      <p begin="00:00:23.52" dur="00:00:02.00">which was the first product that FutureWave did.</p>
      <p begin="00:00:25.52" dur="00:00:02.00">So our vision for this product was to</p>
      <p begin="00:00:27.52" dur="00:00:01.10">make drawing on the computer</p>
      <p begin="00:00:29.02" dur="00:00:01.30" style="1">as <span tts:color="#ccc333">easy</span>
as drawing on paper.</p>
    </div>
  </body>
</tt>
```

Tags Timed Text

O componente FLVPlaybackCaptioning suporta tags Timed Text para arquivos XML de legenda. Para obter mais informações sobre as tags Audio Video Timed Text, consulte www.w3.org. A tabela a seguir lista as tags suportadas e não suportadas.

Uso do componente de legenda FLVPlayback

| Função | Tag/valor | Uso/descrição | Exemplo |
|------------------------------|--|--|-----------------------------|
| Tags ignoradas | metadata | Ignorada/permitida em qualquer nível do documento | |
| | set | Ignorada/permitida em qualquer nível do documento | |
| | xml:lang | Ignorada | |
| | xml:space | Ignorada/o comportamento é substituído por: xml:space="default" | |
| | layout | Ignorada/incluindo qualquer tag de região em uma seção de tag de layout | |
| | tag br | Todos os atributos e conteúdo são ignorados. | |
| Tempo da mídia para legendas | atributos begin | Permitido somente em tags p. Necessário para a implantação do tempo de mídia das legendas. | <p begin="3s"> |
| | atributos dur | Permitido somente em tags p. Recomendado. Se não for incluída, a legenda terminará com o arquivo FLV ou quando outra legenda começar. | |
| | atributos end | Permitido somente em tags p. Recomendado. Se não for incluída, a legenda terminará com o arquivo FLV ou quando outra legenda começar. | |
| Duração das legendas | 00:03:00.1 | Formato de relógio total | |
| | 03:00.1 | Formato de relógio parcial | |
| | 10 | Tempos de ajuste sem unidades. O ajuste representa os segundos. | |
| | 00:03:00:05 00:03:00:05.1 30f 30t | Não suportado. Os formatos de tempo que incluem quadros ou tiques não são suportados. | |
| Tag body | body | Necessária/suporte apenas para uma tag body. | <body><div>...</div></body> |
| Tag content | tag div | É permitido zero ou um valor superior. A primeira tag é usada | |
| | tag p | É permitido zero ou um valor superior. | |
| | tag span | Um contêiner lógico para uma seqüência de unidades de conteúdo textual. Não há suporte para extensões aninhadas. Há suporte para tags style do atributo. | |
| | tag br | Representa uma quebra de linha explícita. | |

| Função | Tag/valor | Uso/descrição | Exemplo |
|---|----------------------|---|--|
| Tags de estilo (Todas as tags style são usadas na tag p) | style | Faz referência a um ou mais elementos de estilo. Pode ser usada como uma tag e um atributo. Como uma tag, é necessário um atributo ID (o estilo pode ser reutilizado no documento). Há suporte para uma ou mais tags style dentro da tag style. | |
| | tts:background Color | Especifique uma propriedade style que defina a cor de fundo de uma região. Alfa será ignorado, a menos que seja definido como zero (alfa 0) para tornar o fundo transparente. O formato de cor é #RRGGBBAA. | |
| | tts:color | Especifique uma propriedade style que defina a cor do primeiro plano. Alfa não suportado para todas as cores. O valor transparent é convertido em preto. | <pre><style id="3" style="2" tts:backgroundColor="white"/> "transparent" = #00000000 "black"=#000000FF "silver"=#C0C0C0FF "grey"=#808080FF "white"=#FFFFFFFF "maroon"=#800000FF "red"=#FF0000FF "purple"=#800080FF "fuchsia"("magenta")= #FF00FFFF "green"=#008000FF "lime"=#00FF00FF "olive"=#808000FF "yellow"=#FFFF00FF "navy"=#000080FF "blue"=#0000FFFF "teal"=#008080FF "aqua"("cyan")=#00FFFFFF</pre> |
| | tts:fontFamily | Especifique uma propriedade style que defina a família de fontes. | <pre>"default" = _serif "monospace" = _typewriter "sansSerif" = _sans "serif" = _serif "monospaceSansSerif" = _typewriter "monospaceSerif" = _typewriter "proportionalSansSerif" = _sans</pre> |

| Função | Tag/valor | Uso/descrição | Exemplo |
|--------|----------------|--|---|
| | tts:fontSize | Especifique uma propriedade style que defina o tamanho da fonte. Somente o primeiro valor (vertical) será usado se os dois forem fornecidos. Os valores de percentual e as unidades são ignorados. Suporte para tamanhos de pixel absoluto (por exemplo, 12) e de estilo relativo (por exemplo, +2). | |
| | tts:fontStyle | Especifique uma propriedade style que defina o estilo da fonte. | "normal" "italic" "inherit"* * O comportamento padrão; herda o estilo da tag delimitada. |
| | tts:fontWeight | Especifique uma propriedade style que defina a espessura da fonte. | "normal" "bold" "inherit"* * O comportamento padrão; herda o estilo da tag delimitada. |

| Função | Tag/valor | Uso/descrição | Exemplo |
|--------------------------|--|--|---|
| | tts:textAlign | Especifique uma propriedade style que defina como as áreas inline são alinhadas em uma área de bloco de conteúdo. | "left" "right" "center" "start" ("=left") "end" ("=right") "inherit"* *Herda o estilo da tag delimitada. Se nenhuma tag textAlign for definida, o padrão será "left". |
| | tts:wrapOption | Especifique uma propriedade style que defina se a quebra de linha automática se aplicará ou não ao contexto do elemento afetado. Essa configuração afeta todos os parágrafos do elemento de legenda. | "wrap" "noWrap" "inherit"* *Herda o estilo da tag delimitada. Se nenhuma tag wrapOption for definida, o padrão será "wrap". |
| Atributos não suportados | 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 | | |

Uso de pontos de sinalização com legendas

Os pontos de sinalização permitem que você interaja com um vídeo. Por exemplo, é possível afetar a execução de um arquivo FLV ou exibir texto em momentos específicos no vídeo. Se você não tiver um arquivo XML Timed Text a ser usado com um arquivo FLV, poderá incorporar os pontos de sinalização de eventos em um arquivo FLV e associá-los ao texto. Esta seção apresenta informações sobre os padrões de pontos de sinalização do componente FLVPlaybackCaptioning e uma visão geral de como associar esses pontos de sinalização ao texto das legendas. Para obter mais informações sobre como incorporar pontos de sinalização de eventos no assistente de importação de vídeo ou no Flash Video Encoder, consulte o Capítulo 16, “Trabalho com vídeo”, em *Uso do Flash*.

Noções básicas dos padrões de ponto de sinalização do FLVPlaybackCaptioning

Nos metadados do arquivo FLV, um ponto de sinalização é representado como um objeto com as seguintes propriedades: `name`, `time`, `type` e `parameters`. Os pontos de sinalização do ActionScript do FLVPlaybackCaptioning têm os seguintes atributos:

name A propriedade `name` é uma string que contém o nome atribuído do ponto de sinalização. A propriedade `name` deve iniciar com o prefixo `fl.video.caption.2.0.` e ter uma string após o prefixo. A string é uma série de números inteiros positivos que são incrementados a cada momento para manter a exclusividade de cada nome. O prefixo inclui o número de versão que também corresponde ao número de versão do FLVPlayback. Para o Adobe Flash CS4 e superior, você deve definir o número de versão como `2.0`.

time A propriedade `time` é a hora em que a legenda deve ser exibida.

type A propriedade `type` é uma string cujo valor é `"event"`.

parameters A propriedade `parameters` é uma matriz que suporta os seguintes pares de nome/valor:

- **text:String** O texto em formato HTML da legenda. Esse texto é passado diretamente para a propriedade `TextField.htmlText`. O componente FLVPlaybackCaptioning suporta uma propriedade `text:n` opcional, que aceita o uso de várias faixas de idioma. Para obter mais informações, consulte [“Suporte para várias faixas de idioma com pontos de sinalização incorporados”](#) na página 187.
- **endTime:Number** A hora em que a legenda deve desaparecer. Se você não especificar essa propriedade, o componente FLVPlaybackCaptioning assumirá que ela não é um número (NaN), e a legenda será exibida até que o arquivo FLV seja concluído (a ocorrência de FLVPlayback despacha o evento `VideoEvent.COMPLETE`). Especifique a propriedade `endTime:Number` em segundos.
- **backgroundColor:uint** Este parâmetro define `TextField.backgroundColor`. Essa propriedade é opcional.
- **backgroundColorAlpha:Boolean** Se `backgroundColor` tiver um alfa equivalente a 0%, o parâmetro definirá `TextField.background = !backgroundColor`. Essa propriedade é opcional.
- **wrapOption:Boolean** Este parâmetro define `TextField.wordWrap`. Essa propriedade é opcional.

Noções básicas da criação de legendas para pontos de sinalização de eventos incorporados

Se você não tiver um arquivo XML Timed Text que contenha legendas para o arquivo FLV, poderá criar legendas associando um arquivo XML que contenha legendas aos pontos de sinalização de eventos incorporados. A amostra XML assume que você executou as seguintes etapas para criar pontos de sinalização de eventos incorporados no vídeo:

- Adicione os pontos de sinalização de eventos (de acordo os padrões de FLVPlaybackCaptioning) e decodifique o vídeo.
- No Flash, arraste um componente FLVPlayback e um componente FLVPlaybackCaptioning para o Palco.
- Defina as propriedades de origem dos componentes FLVPlayback e FLVPlaybackCaptioning (o local do arquivo FLV e o local do arquivo XML).
- Publique.

A amostra a seguir importa o XML para o codificador:

```
<?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>
```

O componente FLVPlaybackCaptioning também suporta várias faixas de idioma com o ponto de sinalização incorporado. Para obter mais informações, consulte [“Suporte para várias faixas de idioma com pontos de sinalização incorporados”](#) na página 187.

Suporte para várias faixas de idioma com pontos de sinalização incorporados

A propriedade `track` do `FLVPlaybackCaptioning` suporta várias faixas de idioma com pontos de sinalização incorporados, contanto que o arquivo XML Timed Text esteja de acordo com os padrões de ponto de sinalização do `FLVPlaybackCaptioning`. (Para obter mais informações, consulte “[Noções básicas dos padrões de ponto de sinalização do FLVPlaybackCaptioning](#)” na página 185.) No entanto, o componente `FLVPlaybackCaptioning` não suporta várias faixas de idioma em arquivos XML separados. Para usar a propriedade `track`, defina-a com um valor diferente de 0. Por exemplo, se você definir a propriedade `track` como 1 (`track == 1`), o componente `FLVPlaybackCaptioning` pesquisará os parâmetros de ponto de sinalização. Se uma correspondência não for encontrada, a propriedade `text` dos parâmetros de ponto de sinalização será usada. Para obter mais informações, consulte a propriedade `track` na *Referência do ActionScript 3.0 para o Flash Professional*.

Execução de vários arquivos FLV com legendas

Você pode abrir vários players de vídeo em uma única ocorrência do componente `FLVPlayback` para executar vários vídeos e alternar entre eles enquanto são executados. Também é possível associar as legendas a cada player de vídeo no componente `FLVPlayback`. Para obter mais informações sobre como abrir vários players de vídeo, consulte “[Usar vários players de vídeo](#)” na página 155. Para usar legendas em vários players de vídeo, crie uma ocorrência do componente `FLVPlaybackCaptioning` para cada `VideoPlayer` e defina o `videoPlayerIndex` do `FLVPlaybackCaptioning` para o índice correspondente. O índice `VideoPlayer` assume como padrão o número 0 quando existe apenas um `VideoPlayer`.

Este é um exemplo de código que atribui legendas exclusivas a vídeos exclusivos. Para executar esse exemplo, suas URLs fictícias devem ser substituídas pelas URLs funcionais.

```
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");
```

Personalização do componente FLVPlaybackCaptioning

Para começar a utilizar o componente `FLVPlaybackCaptioning` rapidamente, opte pelos valores padrão do `FLVPlaybackCaptioning` que colocam as legendas diretamente sobre o componente `FLVPlayback`. Talvez seja necessário personalizar o componente `FLVPlaybackCaptioning` para mover as legendas para fora do vídeo.

O código a seguir demonstra como criar dinamicamente um objeto `FLVPlayback` com o botão de alternância de legendas:

- 1 Posicione o componente `FLVPlayback` no palco em 0,0 e forneça o nome de ocorrência **player**.
- 2 Coloque o componente `FLVPlaybackCaptioning` no palco em 0,0 e forneça o nome de ocorrência **captioning**.
- 3 Posicione o componente `CaptionButton` no palco.
- 4 No exemplo de código a seguir, defina a variável `testVideoPath:String` como um arquivo FLV (usando um caminho absoluto ou relativo).

Nota: O exemplo de código define a variável `testVideoPath` como a amostra de vídeo Flash `caption_video.flv`. Altere essa variável para que se torne o caminho do componente de legenda de vídeo ao qual você está adicionando um componente `Button` de legenda.

- 5 No exemplo de código a seguir, defina a variável `testCaptioningPath:String` como um arquivo XML Timed Text apropriado (usando um caminho absoluto ou relativo).

Nota: O exemplo de código define a variável `testCaptioningPath` como o arquivo XML Timed Text `caption_video.xml`. Altere essa variável para que se torne o caminho do arquivo XML Timed Text que contém as legendas do vídeo.

- 6 Salve o código a seguir como `FLVPlaybackCaptioningExample.as` no mesmo diretório do arquivo FLA.

- 7 Defina `DocumentClass` no arquivo FLA como `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;
        }
    }
}
```

Para obter mais informações sobre todos os parâmetros do `FLVPlaybackCaptioning`, consulte a *Referência do ActionScript 3.0 para Flash Professional*.