

# Uso de componentes ACTIONSCRIPT® 3.0

## **Avisos legales**

Para ver los avisos legales, consulte [http://help.adobe.com/es\\_ES/legalnotices/index.html](http://help.adobe.com/es_ES/legalnotices/index.html).

# Contenido

## Capítulo 1: Introducción

Destinatarios .....	1
Requisitos del sistema .....	1
Documentación .....	2
Convenciones tipográficas .....	2
Términos utilizados en este manual .....	2
Recursos adicionales .....	2

## Capítulo 2: Componentes ActionScript 3.0

Ventajas de utilizar componentes .....	4
Tipos de componentes .....	5
Añadir y eliminar componentes de un documento .....	7
Búsqueda de la versión del componente .....	9
Modelo de gestión de eventos de ActionScript 3.0 .....	10
Una aplicación sencilla .....	10

## Capítulo 3: Trabajo con componentes

Arquitectura de componentes .....	18
Trabajo con archivos de componente .....	20
Depuración de aplicaciones de componentes .....	22
Configuración de parámetros y propiedades .....	22
Biblioteca .....	24
Ajuste del tamaño de los componentes .....	24
Previsualización dinámica .....	24
Gestión de eventos .....	25
Trabajo con la lista de visualización .....	26
Trabajo con FocusManager .....	29
Trabajo con componentes basados en List .....	30
Trabajo con DataProvider .....	31
Trabajo con CellRenderer .....	39
Hacer los componentes accesibles .....	46

## Capítulo 4: Uso de los componentes de la interfaz de usuario

Uso del componente Button .....	47
Uso del componente CheckBox .....	50
Uso del componente ColorPicker .....	53
Uso del componente ComboBox .....	55
Uso del componente DataGrid .....	58
Uso del componente Label .....	64
Uso del componente List .....	66
Uso del componente NumericStepper .....	71
Uso del componente ProgressBar .....	74
Uso del componente RadioButton .....	79

**Contenido**

Uso del componente ScrollPane .....	82
Uso del componente Slider .....	85
Uso del componente TextArea .....	88
Uso del componente TextInput .....	91
Uso del componente TileList .....	94
Uso del componente UILoader .....	97
Uso del componente UIScrollBar .....	99

**Capítulo 5: Personalización de los componentes de la interfaz de usuario**

Personalización de componentes de interfaz de usuario .....	102
Definición de estilos .....	102
Aspectos .....	105
Personalización del componente Button .....	108
Personalización del componente CheckBox .....	110
Personalización del componente ColorPicker .....	111
Personalización del componente ComboBox .....	113
Personalización del componente DataGrid .....	115
Personalización del componente Label .....	120
Personalización del componente List .....	121
Personalización del componente NumericStepper .....	124
Personalización del componente ProgressBar .....	125
Personalización del componente RadioButton .....	127
Personalización del componente ScrollPane .....	129
Personalización del componente Slider .....	130
Personalización del componente TextArea .....	131
Personalización del componente TextInput .....	133
Personalización del componente TileList .....	135
Personalización del componente UILoader .....	137
Personalización del componente UIScrollBar .....	137

**Capítulo 6: Uso del componente FLVPlayback**

Uso del componente FLVPlayback .....	140
Personalización del componente FLVPlayback .....	159
Uso de un archivo SMIL .....	171

**Capítulo 7: Uso del componente FLVPlaybackCaptioning**

Uso del componente FLVPlaybackCaptioning .....	179
Uso de texto temporizado .....	181
Uso de puntos de referencia con ActionScript .....	187
Reproducción de varios archivos FLV con subtítulos .....	189
Personalización del componente FLVPlaybackCaptioning .....	190

# Capítulo 1: Introducción

Adobe® Flash® CS5 es la herramienta estándar de edición profesional para la creación de publicaciones web de gran impacto. Los componentes son bloques de creación para aplicaciones complejas de Internet que proporcionan dicho impacto. Un *componente* es un clip de película con parámetros que permiten personalizar el componente con métodos, propiedades y eventos de Adobe® ActionScript®, durante la edición en Flash o en tiempo de ejecución. El diseño de los componentes permite a los desarrolladores volver a utilizar y compartir código, así como encapsular complejas funciones que los diseñadores podrán emplear y personalizar sin necesidad de utilizar ActionScript.

Los componentes permiten crear con facilidad y rapidez aplicaciones sólidas con apariencia y comportamiento uniformes. En este manual se describe la forma de crear aplicaciones con componentes Adobe ActionScript 3.0. En *Referencia del lenguaje y componentes Adobe® ActionScript® 3.0*, se describe la API (interfaz de programación de aplicaciones) de cada uno de los componentes.

Puede utilizar los componentes creados por Adobe®, descargar componentes creados por otros desarrolladores o crear los suyos propios.

## Destinatarios

Este manual está dirigido a desarrolladores que crean aplicaciones de Flash y desean utilizar componentes para agilizar el desarrollo. Ya deberá estar familiarizado con el desarrollo de aplicaciones en Flash y con la programación en ActionScript.

Si tiene menos experiencia programando en ActionScript, puede añadir componentes a un documento, establecer sus parámetros en el inspector de propiedades o el inspector de componentes, y usar el panel Comportamientos para gestionar sus eventos. Por ejemplo, puede asociar un comportamiento Ir a página web a un componente Button que abre una URL en un navegador web cuando se hace clic en el botón sin tener que escribir código ActionScript.

Si es un programador que desea crear aplicaciones más sólidas, puede crear componentes de forma dinámica, utilizar ActionScript para establecer propiedades y llamar a métodos en tiempo de ejecución, así como utilizar el modelo de detector de eventos para gestionar los eventos.

Para obtener más información, consulte “[Trabajo con componentes](#)” en la página 18.

## Requisitos del sistema

Los componentes de Flash no tienen ningún requisito del sistema adicional a los de Flash.

Cualquier archivo SWF que utilice componentes de Flash CS3 o posterior debe verse con Adobe® Flash® Player 9.0.28.0 o posterior, y debe publicarse para ActionScript 3.0 (puede definir este valor en Archivo > Configuración de publicación, en la ficha Flash).

## Documentación

En este documento se explican los detalles de la utilización de componentes para desarrollar aplicaciones Flash. Se supone que tiene un conocimiento general de Flash y ActionScript 3.0. Se encuentra disponible de forma separada documentación sobre Flash y los productos relacionados.

Este documento está disponible en formato de archivo PDF y como Ayuda en línea. Para ver la Ayuda en línea, inicie Flash y seleccione Ayuda > Ayuda de Flash > Uso de componentes Adobe ActionScript 3.0.

Para más información sobre Flash, consulte los siguientes documentos:

- *Uso de Flash*
- *Guía del desarrollador de ActionScript 3.0*
- *Referencia de Adobe ActionScript 3.0 para Flash Professional*

## Convenciones tipográficas

En este manual se utilizan las siguientes convenciones tipográficas:

- La *f fuente en cursiva* indica un valor que se debe sustituir (por ejemplo, en una ruta de carpeta).
- La *f fuente* para `código` indica que se trata de código ActionScript, incluidos los nombres de métodos y propiedades.
- La *f fuente para código en cursiva* indica un elemento de código que se debe sustituir (por ejemplo, un parámetro de ActionScript).
- La **f fuente en negrita** indica un valor introducido por el usuario.

## Términos utilizados en este manual

En este manual se utilizan los términos siguientes:

**en tiempo de ejecución** Cuando el código se ejecuta en Flash Player.

**durante la edición** Mientras se trabaja en el entorno de edición de Flash.

## Recursos adicionales

Además del contenido de estos manuales, Adobe proporciona regularmente artículos actualizados, ideas de diseño y ejemplos en el Centro de desarrollo de Adobe y el Centro de diseño de Adobe.

Puede encontrar ejemplos adicionales de componentes en [www.adobe.com/go/learn\\_fl\\_samples\\_es](http://www.adobe.com/go/learn_fl_samples_es).

### Centro de desarrollo de Adobe

El Centro de desarrollo de Adobe contiene la información más actualizada sobre ActionScript, artículos sobre el desarrollo de aplicaciones reales e información sobre nuevos problemas importantes. Consulte el Centro de desarrollo en [www.adobe.com/go/flash\\_devcenter\\_es](http://www.adobe.com/go/flash_devcenter_es).

**Centro de diseño de Adobe**

Póngase al día en diseño digital y gráficos en movimiento. Examine la obra de importantes artistas, descubra las nuevas tendencias de diseño y mejore sus conocimientos con tutoriales, flujos de trabajo clave y técnicas avanzadas. Consulte el centro cada quince días para ver tutoriales y artículos nuevos e inspirarse con las creaciones de las galerías. Consulte el Centro de diseño en [www.adobe.com/go/fl\\_designcenter\\_es](http://www.adobe.com/go/fl_designcenter_es).

# Capítulo 2: Componentes ActionScript 3.0

Los componentes de Adobe® Flash® Professional CS5 son clips de película con parámetros que permiten modificar su aspecto y comportamiento. Un componente puede ser un control simple de la interfaz de usuario, como los componentes RadioButton o CheckBox, o bien, puede incluir contenido, como los componentes List o DataGrid.

Los componentes permiten crear con facilidad y rapidez Flash aplicaciones sólidas con apariencia y comportamiento uniformes. En lugar de crear botones personalizados, cuadros combinados y listas, se pueden utilizar los componentes de Flash que implementan dichos controles. Basta con arrastrar dichos componentes del panel Componentes al documento de aplicación. Asimismo, la apariencia de estos componentes se puede personalizar fácilmente y conseguir, de esta manera, una mayor adaptación al diseño de la aplicación.

Es posible realizar todo esto sin tener conocimientos avanzados de ActionScript y también utilizar ActionScript 3.0 para modificar el comportamiento de un componente o implementar uno nuevo. Cada componente dispone de un conjunto único de métodos, propiedades y eventos de ActionScript, llamado *interfaz de programación de aplicaciones* (API). La API permite crear y manipular componentes mientras se ejecuta la aplicación.

La API también permite crear componentes nuevos personalizados. Se pueden descargar componentes creados por miembros de la comunidad de Flash desde el sitio web de la zona de intercambio de Adobe, en la dirección [www.adobe.com/go/flash\\_exchange\\_es](http://www.adobe.com/go/flash_exchange_es). Para obtener información sobre la creación de un componente, consulte [www.adobe.com/go/learn\\_fl\\_creating\\_components\\_es](http://www.adobe.com/go/learn_fl_creating_components_es).

La arquitectura de componentes ActionScript 3.0 incluye clases en las que se basan todos los componentes, estilos y aspectos que permiten personalizar la apariencia, un modelo de gestión de eventos, la administración de la selección, una interfaz de accesibilidad, etc.

**Nota:** Adobe Flash CS5 incluye componentes ActionScript 2.0 y componentes ActionScript 3.0. Estos dos conjuntos de componentes no se pueden mezclar. Debe utilizarse uno de los dos conjuntos en una aplicación concreta. Flash CS5 presenta componentes ActionScript 2.0 o componentes ActionScript 3.0 en función de si se abre un archivo de ActionScript 2.0 o de ActionScript 3.0. Al crear un documento de Flash, debe elegirse entre Archivo de Flash (ActionScript 3.0) o Archivo de Flash (ActionScript 2.0). Cuando se abre un documento existente, Flash examina la Configuración de publicación para determinar el conjunto de componentes que se debe utilizar. Para obtener información sobre los componentes ActionScript 2.0, consulte *Uso de componentes Adobe® ActionScript® 2.0*.

Para ver una lista completa de los componentes ActionScript 3.0 para Flash, consulte “Tipos de componentes” en la página 5.

## Ventajas de utilizar componentes

Los componentes permiten separar el proceso de designación de la aplicación del proceso de programación. Permiten a los desarrolladores crear funciones que los diseñadores pueden utilizar en las aplicaciones. Los desarrolladores pueden encapsular en componentes las funciones utilizadas con mayor frecuencia y los diseñadores pueden personalizar el tamaño, la ubicación y el comportamiento de los componentes cambiando sus parámetros. También pueden cambiar la apariencia de un componente mediante la edición de sus elementos gráficos o aspectos.

Los componentes comparten una funcionalidad básica, como los estilos, los aspectos y la administración de la selección. Cuando se añade el primer componente a una aplicación, esta funcionalidad básica ocupa aproximadamente 20 kilobytes del tamaño. Cuando se añaden otros componentes, la asignación de memoria inicial se comparte con los componentes añadidos, lo que reduce el crecimiento de tamaño de la aplicación.

En esta sección se describen algunas de las ventajas de los componentes ActionScript 3.0.

**ActionScript 3.0** proporciona un avanzado lenguaje de programación orientado a objetos, de grandes prestaciones, que representa un paso importante en la evolución de las capacidades de Flash Player. El lenguaje se ha diseñado para crear aplicaciones complejas de Internet basadas en un código reutilizable. ActionScript 3.0 se basa en ECMAScript, el lenguaje estandarizado internacional para la creación de secuencias de comandos, y es compatible con la especificación de lenguaje ECMAScript (ECMA-262) edición 3. Para ver una introducción detallada a ActionScript 3.0, consulte la *Guía del desarrollador de ActionScript 3.0*. Para obtener información de referencia sobre el lenguaje, consulte la [Referencia de ActionScript 3.0](#).

**Los componentes de interfaz de usuario basados en FLA** brindan un fácil acceso a los aspectos para personalizarlos durante el proceso de edición. Estos componentes también proporcionan estilos, incluidos estilos de aspecto, que permiten personalizar la apariencia de los componentes y cargar aspectos en tiempo de ejecución. Para obtener más información, consulte “[Personalización de los componentes de la interfaz de usuario](#)” en la página 102 y la [Referencia de ActionScript 3.0](#).

**El nuevo componente FVLPlayback añade el componente FLVPlaybackCaptioning** además de compatibilidad de pantalla completa, una mejor previsualización dinámica, aspectos que permiten añadir configuración de color y de alfa, y funciones de diseño y descarga de FLV mejoradas.

**El inspector de propiedades y el inspector de componentes** permiten cambiar los parámetros de componentes durante el proceso de edición en Flash. Para más información, consulte “[Trabajo con archivos de componente](#)” en la página 20 y “[Configuración de parámetros y propiedades](#)” en la página 22.

**El nuevo cuadro de diálogo de colección** de los componentes ComboBox, List y TileList permite llenar la propiedad `dataProvider` de estos componentes a través de la interfaz de usuario. Para más información, consulte “[Creación de un objeto DataProvider](#)” en la página 31.

**El modelo de eventos de ActionScript 3.0** permite que la aplicación detecte los eventos y llame a los controladores de eventos para responder. Para obtener más información, consulte “[Modelo de gestión de eventos de ActionScript 3.0](#)” en la página 10 y “[Gestión de eventos](#)” en la página 25.

**Las clases de administrador** proporcionan una forma sencilla de gestionar la selección y administrar los estilos en una aplicación. Para obtener más información, consulte la [Referencia de ActionScript 3.0](#).

**La clase base de UIComponent** proporciona propiedades, eventos y métodos esenciales a los componentes que lo forman. Todos los componentes de interfaz de usuario de ActionScript 3.0 heredan de la clase UIComponent. Para obtener más información consulte la clase UIComponent en la [Referencia de ActionScript 3.0](#).

**La utilización de un archivo SWC** en los componentes de interfaz de usuario basados en FLA proporciona definiciones de ActionScript como un activo incluido en la línea de tiempo del componente con el fin de acelerar la compilación.

**La jerarquía de clases ampliable fácilmente** mediante ActionScript 3.0 le permite crear espacios de nombres exclusivos, importar clases cuando sea necesario y poner fácilmente en subclases para ampliar componentes.

Para obtener más información, consulte la [Referencia de ActionScript 3.0](#).

*Nota:* Flash CS5 admite componentes basados en FLA y componentes basados en SWC. Para más información, consulte “[Arquitectura de componentes](#)” en la página 18.

## Tipos de componentes

Los componentes de Flash se instalan durante la instalación de Flash CS5.

Los componentes ActionScript 3.0 incluyen los siguientes componentes de interfaz de usuario (IU):

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

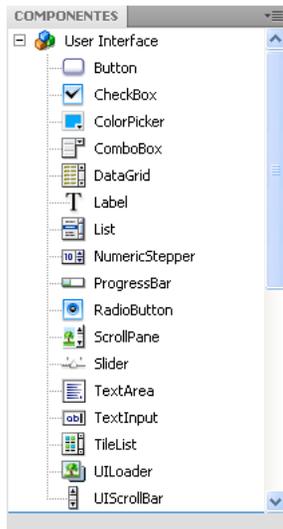
Además de los componentes de interfaz de usuario, los componentes ActionScript 3.0 de Flash incluyen los siguientes componentes y clases de soporte:

- El componente FLVPlayback (fl.video.FLVPlayback), que es un componente basado en SWC.  
El componente FLVPlayback permite incluir fácilmente un reproductor de vídeo en la aplicación Flash para reproducir vídeo transmitido de forma progresiva a través de HTTP desde Adobe® Flash® Video Streaming Service (FVSS) o desde Macromedia® Flash® Media Server de Adobe (FMS). Para obtener más información, consulte [“Uso del componente FLVPlayback”](#) en la página 140.
- Los componentes de interfaz de usuario personalizados FLVPlayback, que se basan en FLA y funcionan en las versiones ActionScript 2.0 y ActionScript 3.0 del componente FLVPlayback. Para obtener más información, consulte [“Uso del componente FLVPlayback”](#) en la página 140.
- El componente FLVPlaybackCaptioning, que proporciona subtitulación oculta para FLVPlayback. Consulte [“Uso del componente FLVPlaybackCaptioning”](#) en la página 179.  
Para obtener una lista completa de los componentes ActionScript 3.0 y las clases que admiten, consulte la [Referencia de ActionScript 3.0](#).

#### Ver componentes de Flash:

Para ver los componentes ActionScript 3.0 de Flash en el panel Componentes, siga los pasos que se indican a continuación.

- 1 Inicie Flash.
- 2 Cree un archivo de Flash (ActionScript 3.0) o abra un documento de Flash ya existente que tenga definido ActionScript 3.0 en la Configuración de publicación.
- 3 Seleccione Ventana > Componentes para abrir el panel Componentes si aún no está abierto.



Panel Componentes con componentes de interfaz de usuario

También se pueden descargar componentes adicionales del sitio web de la zona de intercambio de Adobe en [www.adobe.com/go/flash\\_exchange\\_es](http://www.adobe.com/go/flash_exchange_es). Para instalar componentes descargados de la zona de intercambio, es necesario descargar e instalar Adobe® Extension Manager en [www.adobe.com/go/exchange\\_es](http://www.adobe.com/go/exchange_es). Haga clic en el vínculo Página principal de la zona de intercambio de Adobe y busque el vínculo Extension Manager.

Cualquier componente puede aparecer en el panel Componentes de Flash. Realice estos pasos para instalar componentes en un equipo con Windows® o Macintosh®

### Instalar componentes en un equipo con Windows o Macintosh:

- 1 Salga de Flash.
- 2 Coloque el archivo SWC o FLA que contiene el componente en la siguiente carpeta del disco duro:
  - En Windows:  
C:\Archivos de programa\Adobe\Adobe FlashCS5\idioma\Configuration\Components
  - En Macintosh:  
Macintosh HD:Aplicaciones:Adobe Flash CS5:Configuration:Components
- 3 Inicie Flash.
- 4 Seleccione Ventana > Componentes para ver los componentes del panel Componentes, si aún no está abierto.  
Para más información sobre archivos de componentes, consulte “Trabajo con archivos de componente” en la página 20.

## Añadir y eliminar componentes de un documento

Cuando se arrastra un componente basado en FLA del panel Componentes al escenario, Flash importa un clip de película editable en la biblioteca. Cuando se arrastra un componente basado en SWC al escenario, Flash importa un clip compilado en la biblioteca. Tras importar un componente en la biblioteca, se pueden arrastrar instancias del mismo al escenario, desde la biblioteca o desde el panel Componentes.

## Añadir componentes durante edición

Para añadir un componente a un documento, basta con arrastrarlo desde el panel Componentes. Se pueden definir las propiedades de cada instancia de un componente en la ficha Parámetros o en inspector de propiedades en la ficha del inspector de componentes.

- 1 Seleccione Ventana > Componentes.
- 2 Haga doble clic en el componente en el panel Componentes o arrastre el componente al escenario.
- 3 Seleccione el componente en el escenario.
- 4 Si el inspector de propiedades no está visible, seleccione Ventana > Propiedades > Propiedades.
- 5 En el inspector de propiedades, introduzca un nombre para la instancia de componente.
- 6 Seleccione Ventana > inspector de componentes y seleccione la ficha Parámetros para configurar los parámetros para la instancia.

Para más información, consulte [“Configuración de parámetros y propiedades”](#) en la página 22.

- 7 Cambie el tamaño del componente como desee, editando los valores de anchura (An.:. ) y altura (Al.:. ).

Para más información sobre la dimensión de tipos de componente específicos, consulte [“Personalización de los componentes de la interfaz de usuario”](#) en la página 102.

- 8 Seleccione > Control Probar película o presione Control+Intro para compilar el documento y ver los resultados de la configuración.

También es posible cambiar el color y el formato de texto de un componente si se definen propiedades de estilo o se personaliza su apariencia mediante la edición de los aspectos del componente. Para más información sobre estos temas, consulte [“Personalización de los componentes de la interfaz de usuario”](#) en la página 102.

Si se arrastra un componente al escenario durante la edición, se puede hacer referencia al componente a través de su nombre de instancia (por ejemplo, myButton).

## Añadir componentes en tiempo de ejecución mediante ActionScript

Para añadir un componente a un documento en tiempo de ejecución mediante ActionScript, el componente debe hallarse en la biblioteca de la aplicación (Ventana > Biblioteca) cuando se compila el archivo SWF. Para añadir un componente a la biblioteca, arrastre el componente del panel Componentes al panel Biblioteca. Para más información sobre la biblioteca, consulte [“Biblioteca”](#) en la página 24.

También es necesario importar el archivo de clase del componente para que su API esté disponible para la aplicación. Los archivos de clase de componente se instalan en *paquetes* que contienen una o varias clases. Para importar una clase de componente, se debe utilizar la sentencia `import` y especificar el nombre de paquete y el nombre de clase. Por ejemplo, se podría importar la clase Button con la siguiente sentencia `import`:

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

Para obtener información sobre qué componente contiene cada paquete, consulte la [Referencia de ActionScript 3.0](#). Para más información sobre la ubicación de los archivos de origen de componentes, consulte [“Trabajo con archivos de componente”](#) en la página 20.

Para crear una instancia del componente, es necesario llamar al método constructor de ActionScript para el componente. Por ejemplo, la siguiente sentencia crea una instancia de Button denominada aButton:

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

Finalmente, es necesario llamar al método estático `addChild()` para añadir la instancia del componente al escenario o contenedor de aplicación. Por ejemplo, la siguiente sentencia añade la instancia aButton:

```
addChild(aButton);
```

En este punto, se puede utilizar la API del componente para especificar de forma dinámica el tamaño y la posición del componente en el escenario, detectar eventos y definir propiedades para modificar su comportamiento. Para obtener más información sobre la API de un componente concreto, consulte la [Referencia de ActionScript 3.0](#).

Para más información sobre el método `addChild()`, consulte “[Trabajo con la lista de visualización](#)” en la página 26.

## Eliminación de un componente

Para eliminar una instancia de un componente del escenario durante la edición, basta con seleccionarla y presionar la tecla Supr. De esta forma, se eliminará la instancia del escenario pero no se quitará el componente de la aplicación.

Para eliminar un componente de un documento de Flash después de haberlo colocado en el escenario o en la biblioteca, es necesario eliminar el componente y sus activos asociados de la biblioteca. No basta con eliminar el componente del escenario. Si no se elimina de la biblioteca, se incluirá en la aplicación cuando se compile.

- 1 En el panel Biblioteca, seleccione el símbolo del componente.
- 2 Haga clic en el botón Eliminar situado en la parte inferior del panel Biblioteca, o seleccione Eliminar en el menú del panel Biblioteca.

Repita estos pasos para eliminar los activos asociados al componente.

Para información sobre la eliminación de un componente del contenedor durante la ejecución de la aplicación, consulte “[Eliminación de un componente de la lista de visualización](#)” en la página 28.

## Búsqueda de la versión del componente

Los componentes ActionScript 3.0 de Flash tienen una propiedad de versión que puede mostrarse si es necesario proporcionarla al servicio técnico de Adobe o si se precisa saber la versión del componente que se está utilizando.

### Para mostrar el número de versión de un componente de interfaz de usuario:

- 1 Cree un documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre el componente al escenario y asígnele un nombre de instancia. Por ejemplo, arrastre una instancia de ComboBox al escenario y denomínela **aCb**.
- 3 Presione la tecla **F9** o seleccione Ventana > Acciones para abrir el panel Acciones.
- 4 Haga clic en el primer fotograma de la línea de tiempo principal y añada el siguiente código al panel Acciones:

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

En el panel Salida debería aparecer un número de versión similar al de la siguiente ilustración.

En los componentes FLVPlayback y FLVPlaybackCaptioning, es necesario hacer referencia al nombre de clase y no al nombre de instancia porque el número de versión se almacena en una constante de clase.

### Mostrar el nombre de versión de los componentes FLVPlayback y FLVPlaybackCaptioning:

- 1 Cree un documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre los componentes FLVPlayback y FLVPlaybackCaptioning al panel Biblioteca.
- 3 Presione la tecla **F9** o seleccione Ventana > Acciones para abrir el panel Acciones.
- 4 Haga clic en el primer fotograma de la línea de tiempo principal y añada el siguiente código al panel Acciones.

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

Los números de versión aparecen en el panel Salida.

## Modelo de gestión de eventos de ActionScript 3.0

ActionScript 3.0 presenta un único modelo de gestión de eventos que sustituye a todos los mecanismos que existían en las versiones anteriores de ActionScript. El nuevo modelo de eventos se basa en la especificación de eventos DOM (modelo de objetos de documento) de nivel 3.

Para los desarrolladores con experiencia en el uso del método `addListener()` de ActionScript 2.0, puede resultar útil señalar las diferencias entre el modelo de detectores de eventos de ActionScript 2.0 y el de ActionScript 3.0. En la siguiente lista se describen algunas de las diferencias principales entre los dos modelos de eventos:

- Para añadir detectores de eventos en ActionScript 2.0, es necesario usar `addListener()` en algunos casos y `addEventListener()` en otros, mientras que en ActionScript 3.0 siempre se utiliza `addEventListener()`.
- En ActionScript 2.0 no existe el flujo del evento, lo que quiere decir que el método `addListener()` sólo se puede llamar en el objeto que difunde el evento, mientras que en ActionScript 3.0, el método `addEventListener()` se puede llamar en cualquier objeto que forme parte del flujo del evento.
- En ActionScript 2.0, los detectores de eventos pueden ser funciones, métodos u objetos, mientras que en ActionScript 3.0 sólo las funciones o los métodos pueden ser detectores de eventos.
- La sintaxis `on(evento)` ya no se utiliza en ActionScript 3.0, de modo que no es posible adjuntar código de eventos de ActionScript a un clip de película. Sólo se puede utilizar `addEventListener()` para añadir un detector de eventos.

En el siguiente ejemplo, donde se detecta un evento `MouseEvent.CLICK` en un componente `Button` denominado `aButton`, se ilustra el modelo básico de gestión de eventos de 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 más información sobre la gestión de eventos de ActionScript 3.0, consulte *Programación con ActionScript 3.0*. Para más información sobre la gestión de eventos de ActionScript 3.0 en los componentes, consulte “[Gestión de eventos](#)” en la página 25.

## Una aplicación sencilla

En esta sección se describen los pasos para crear una sencilla aplicación ActionScript 3.0 con los componentes de Flash y la herramienta de edición de Flash. El ejemplo se ofrece como un archivo FLA con el código ActionScript incluido en la línea de tiempo y también como un archivo de clase ActionScript externo con un archivo FLA que contiene sólo los componentes de la biblioteca. Por lo general, se desarrollarán aplicaciones más extensas con archivos de clase externos para poder compartir código entre las clases y aplicaciones, y para facilitar el mantenimiento de las aplicaciones. Para más información sobre la programación con ActionScript 3.0, consulte *Programación con ActionScript 3.0*.

## Diseño de la aplicación

El primer ejemplo de una aplicación de componentes ActionScript es una variación de la aplicación "Hello World" estándar, de modo que su diseño es bastante sencillo:

- La aplicación se denominará Greetings.
- Utiliza un componente TextArea para mostrar un saludo que inicialmente es "Hello World".
- Utiliza un componente ColorPicker que permite cambiar el color del texto.
- Utiliza tres componentes RadioButton que permiten definir el tamaño del texto en pequeño, mayor o máximo.
- Utiliza un componente ComboBox que permite seleccionar un saludo distinto en una lista desplegable.
- La aplicación utiliza componentes del panel Componentes y también crea elementos de aplicación a través de código ActionScript.

Teniendo en cuenta esta definición, ya puede empezar a crear la aplicación.

## Creación de la aplicación Greetings

A continuación se describen los pasos para crear la aplicación Greetings utilizando la herramienta de edición de Flash para crear un archivo FLA, colocar componentes en el escenario y añadir código ActionScript a la línea de tiempo.

### Crear la aplicación Greetings en un archivo FLA:

- 1 Seleccione Archivo > Nuevo.
- 2 En el cuadro de diálogo Nuevo documento, seleccione Archivo de Flash (ActionScript 3.0) y haga clic en Aceptar.  
Se abre una nueva ventana de Flash.
- 3 Seleccione Archivo > Guardar, asigne al archivo de Flash el nombre **Greetings.fla** y haga clic en el botón Guardar.
- 4 En el panel Componentes de Flash, seleccione un componente TextArea y arrástrelo al escenario.
- 5 En la ventana Propiedades, con el componente TextArea seleccionado en el escenario, escriba **aTa** como nombre de instancia e introduzca la siguiente información:
  - Introduzca **230** en el valor An. (anchura).
  - Introduzca **44** en el valor Al. (altura).
  - Introduzca **165** en el valor X (posición horizontal).
  - Introduzca **57** en el valor Y (posición vertical).
  - Introduzca **Hello World!** en el parámetro de texto, en la ficha Parámetros.
- 6 Arrastre un componente ColorPicker al escenario, colóquelo a la izquierda del componente TextArea y asígnele el nombre de instancia **txtCp**. Introduzca la siguiente información en el inspector de propiedades:
  - Introduzca **96** en el valor X.
  - Introduzca **72** en el valor Y.
- 7 Arrastre tres componentes RadioButton al escenario, de forma consecutiva, y asígneles los nombres de instancia **smallRb**, **largerRb** y **largestRb**. Introduzca la siguiente información para dichos componentes en el inspector de propiedades:
  - Introduzca **100** en el valor de anchura y **22** en el valor de altura en cada uno de ellos.
  - Introduzca **155** en el valor X.
  - Introduzca **120** en el valor Y de smallRb, **148** en el de largerRb y **175** en el de largestRb.

- Introduzca **fontRbGrp** en el parámetro `groupName` de cada uno de ellos.
  - Introduzca etiquetas en la ficha Parámetros de **Small, Larger, Largest**.
- 8 Arrastre un componente `ComboBox` al escenario y asígnele el nombre de instancia **msgCb**. Introduzca la siguiente información para dicho componente en el inspector de propiedades:
- Introduzca **130** en el valor de anchura.
  - Introduzca **265** en el valor X.
  - Introduzca **120** en el valor Y.
  - En la ficha Parámetros, introduzca **Greetings** en el parámetro `prompt`.
  - Haga doble clic en el campo de texto del parámetro `dataProvider` para abrir el cuadro de diálogo Valores.
  - Haga clic en el signo más y reemplace el valor de la etiqueta con **Hello World!**
  - Repita el paso anterior para añadir los valores de etiqueta **Have a nice day!** y **Top of the Morning!**
  - Haga clic en Aceptar para cerrar el cuadro de diálogo Valores.
- 9 Guarde el archivo.
- 10 Si aún no está abierto, abra el panel Acciones presionando **F9** o seleccionando Acciones en el menú Ventana. Haga clic en el primer fotograma de la línea de tiempo principal y añada el siguiente código al panel Acciones:

```
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);
```

Las primeras tres líneas importan las clases de eventos que utiliza la aplicación. Un evento se produce cuando un usuario interactúa con uno de los componentes. Las siguientes cinco líneas registran controladores de eventos para los eventos que la aplicación desea detectar. Un evento `click` se produce en un componente `RadioButton` cuando un usuario hace clic en él. Un evento `change` se produce cuando un usuario selecciona un color distinto en el componente `ColorPicker`. Un evento `change` se produce en el componente `ComboBox` cuando un usuario elige otro saludo de la lista desplegable.

La cuarta línea importa la clase `RadioButtonGroup` para que la aplicación pueda asignar un detector de eventos en el grupo de componentes `RadioButton` en lugar de asignar el detector a cada botón de forma individual.

- 11 Añada la siguiente línea de código al panel Acciones para crear el objeto `TextFormat tf` que la aplicación utiliza para cambiar las propiedades de estilo `size` y `color` del texto en el componente `TextArea`.

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

- 12 Añada el siguiente código para crear la función de gestión de eventos `rbHandler`. Esta función gestiona un evento `click` cuando un usuario hace clic en uno de los 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);
}
```

Esta función utiliza una sentencia `switch` para examinar la propiedad `target` del objeto `event` con el fin de determinar cuál de los componentes `RadioButton` activó el evento. La propiedad `currentTarget` contiene el nombre del objeto que activó el evento. Según cuál sea el componente `RadioButton` en el que haya hecho clic el usuario, la aplicación cambia el tamaño del texto en el componente `TextArea` a 14, 18 ó 24 puntos.

- 13 Añada el siguiente código para implementar la función `cpHandler()`, que gestiona un cambio en el valor de `ColorPicker`:

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

Esta función establece la propiedad `color` del objeto `TextFormat tf` en el color seleccionado en `ColorPicker` y luego llama a `setStyle()` para aplicarlo al texto en la instancia `aTa` de `TextArea`.

- 14 Añada el siguiente código para implementar la función `cbHandler()`, que gestiona un cambio en la selección de `ComboBox`:

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

Esta función simplemente reemplaza el texto de `TextArea` con el texto seleccionado en `ComboBox`, `event.target.selectedItem.label`.

- 15 Seleccione `Control > Probar película` o presione `Control+Intro` para compilar el código y probar la aplicación `Greetings`.

En la siguiente sección se muestra cómo generar la misma aplicación con una clase `ActionScript` externa y un archivo `FLA` que sólo contiene los componentes necesarios en la biblioteca.

#### Para crear la aplicación `Greetings2` con un archivo de clase externo:

- 1 Seleccione `Archivo > Nuevo`.
- 2 En el cuadro de diálogo `Nuevo documento`, seleccione `Archivo de Flash (ActionScript 3.0)` y haga clic en `Aceptar`.  
Se abre una nueva ventana de `Flash`.
- 3 Seleccione `Archivo > Guardar`, asigne al archivo de `Flash` el nombre **`Greetings2.fla`** y haga clic en el botón `Guardar`.
- 4 Arrastre cada uno de los componentes siguientes del panel `Componentes` a la biblioteca:
  - `ColorPicker`
  - `ComboBox`

- RadioButton
- TextArea

El archivo SWF compilado utilizará todos estos activos, de forma que hay que añadirlos a la biblioteca. Arrastre los componentes a la parte inferior del panel Biblioteca. A medida que añada estos componentes a la biblioteca, se añadirán automáticamente otros activos (como List, TextInput y UIScrollBox).

5 En la ventana Propiedades, escriba **Greetings2** en Clase de documento.

Si Flash muestra una advertencia indicando que "no se pudo encontrar una definición para la clase del documento", páselo por alto. Definirá la clase Greetings2 en los siguientes pasos. Esta clase define la funcionalidad principal de la aplicación.

6 Guarde el archivo Greetings2.fla.

7 Seleccione Archivo > Nuevo.

8 En el cuadro de diálogo Nuevo documento, seleccione Archivo ActionScript y haga clic en Aceptar.

Se abre una nueva ventana Script.

9 Añada el siguiente código a la ventana 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() {
```

El script define una clase de ActionScript 3.0 denominada Greetings2. El script realiza lo siguiente:

- Importa las clases que se utilizarán en el archivo. Normalmente, estas sentencias `import` se añadirían a medida que se hace referencia a las distintas clases en el código pero, para ser breves, en este ejemplo se importan todas en un solo paso.
- Declara variables que representan los distintos tipos de objetos de componente que se añadirán al código. Otra variable crea el objeto `TextFormat tf`.
- Define una función constructora, `Greetings2()`, para la clase. En los siguientes pasos se añadirán líneas a esta función y otros métodos a la clase.

10 Seleccione Archivo > Guardar, asigne al archivo el nombre **Greetings2.as** y haga clic en el botón Guardar.

11 Añada las siguientes líneas de código a la función `Greeting2()`:

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

La función debería tener el siguiente aspecto:

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

12 Añada las siguientes líneas de código después de la llave de cierre del 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);
    }
```

Estas líneas realizan lo siguiente:

- Crean instancias de los componentes que se utilizan en la aplicación.
- Definen el tamaño, la posición y las propiedades de cada componente.
- Añaden cada componente al escenario, a través del método `addChild()`.

**13** Después de la llave de cierre del método `buildRadioButtons()`, añada el siguiente código para el 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;
}
}
}
```

Estas funciones definen los detectores de eventos de los componentes.

**14** Seleccione Archivo > Guardar para guardar el archivo.

**15** Seleccione Control > Probar película o presione Control+Intro para compilar el código y probar la aplicación Greetings2.

## Desarrollo y ejecución de ejemplos posteriores

Después de desarrollar y ejecutar la aplicación Greetings, habrá adquirido los conocimientos básicos que necesita para ejecutar los demás ejemplos de código presentados en este manual. En ellos se resaltarán y explicarán el código ActionScript 3.0 relevante de cada ejemplo y debería poder cortar y pegar cada uno de los ejemplos de este manual en un archivo FLA, compilarlo y ejecutarlo.

# Capítulo 3: Trabajo con componentes

## Arquitectura de componentes

Los componentes Adobe® ActionScript® 3.0 se admiten en Adobe® Flash Player 9.0.28.0 y versiones posteriores. Dichos componentes no son compatibles con los componentes creados antes de Flash CS4. Para obtener información sobre el uso de los componentes Adobe® ActionScript® 2.0, consulte *Uso de componentes Adobe® ActionScript® 2.0* y la Referencia del lenguaje de componentes *Adobe® ActionScript® 2.0*.

Los componentes de interfaz de usuario Adobe ActionScript 3.0 se implementan como componentes basados en FLA, aunque CS5 admite los componentes basados en SWC y FLA. Por ejemplo, los componentes FLVPlayback y FLVPlaybackCaptioning son componentes basados en SWC. Se puede colocar cualquier tipo de componente en la carpeta Components para que aparezca en el panel Componentes. Estos dos tipos de componentes se crean de forma diferente, de manera que se describen individualmente en esta documentación.

## Componentes ActionScript 3.0 basados en FLA

Los componentes de interfaz de usuario ActionScript 3.0 son archivos basados en FLA (.fla) con aspectos incorporados; se puede acceder a ellos y editarlos haciendo doble clic en el componente en el escenario. Los aspectos y otros activos del componente se colocan en el fotograma 2 de la línea de tiempo. Al hacer doble clic en el componente, Flash salta automáticamente al fotograma 2 y abre la paleta de los aspectos del componente. En la siguiente ilustración se muestra la paleta de aspectos que se muestran para el componente Button.



*Aspectos del componente Button*

Para más información sobre aspectos de los componentes y personalización de los componentes, consulte [“Personalización de los componentes de la interfaz de usuario”](#) en la página 102 y [“Personalización del componente FLVPlayback”](#) en la página 159.

Para acelerar la compilación de las aplicaciones y evitar conflictos con la configuración de ActionScript 3.0, los componentes de interfaz de usuario basados en FLA de Flash CS5 también contienen un SWC que incluye el código ActionScript ya compilado del componente. El archivo SWC ComponentShim se coloca en el fotograma 2 del escenario, en cada componente de interfaz de usuario, para que las definiciones precompiladas estén disponibles. Para que un componente esté disponible para ActionScript, debe estar en el escenario o en la biblioteca con la opción Exportar en primer fotograma seleccionada en sus Propiedades de vinculación. Para crear un componente con ActionScript, también se debe importar la clase con una sentencia `import` para acceder a ella. Para obtener información sobre la sentencia `import`, consulte la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Componentes basados en SWC

Los componentes basados en SWC tienen un archivo de clase FLA y ActionScript, pero no se han compilado ni exportado como SWC. Un archivo SWC es un paquete de símbolos de Flash y código ActionScript precompilados que permiten evitar una nueva compilación de símbolos y código que no vayan a cambiar.

Los componentes FLVPlayback y FLVPlaybackCaptioning son componentes basados en SWC. Tienen aspectos externos, en lugar de incorporados. El componente FLVPlayback tiene un aspecto predeterminado que se puede cambiar seleccionando uno de una colección de aspectos prediseñados, mediante la personalización de controles de interfaz de usuario en el panel Componentes (BackButton, BufferingBar, etc.) o mediante la creación de un aspecto personalizado. Para obtener más información, consulte “[Personalización del componente FLVPlayback](#)” en la página 159.

En Flash, se puede convertir un clip de película en un clip compilado de la forma siguiente:

### Compilación de un clip de película

- Haga clic con el botón derecho del ratón (Windows) o presione Control y haga clic (Macintosh) en el clip de película en el panel Biblioteca y, a continuación, seleccione Convertir en clip compilado.

El clip compilado se comporta como el clip de película desde el que se compiló, aunque los clips compilados se muestran y se publican más rápido que los clips de película normales. Los clips compilados no se pueden editar, pero sus propiedades pueden aparecer en el inspector de propiedades y en el inspector de componentes.

Los componentes SWC contienen un clip compilado, las definiciones de ActionScript precompiladas del componente, así como otros archivos que describen el componente. Si se crea un componente personalizado, se puede exportar como archivo SWC para distribuirlo.

### Exportación de un archivo SWC

- Seleccione el clip de película en el panel Biblioteca, haga clic con el botón derecho del ratón (Windows) o presione la tecla Control y haga clic (Macintosh), y seleccione Exportar archivo SWC.

***Nota:** el formato de un archivo SWC de Flash CS4 o posterior es compatible con el formato SWC de Flex, de manera que los archivos SWC se pueden intercambiar entre ambos productos, pero no necesariamente sin modificaciones.*

Para obtener información sobre la creación de un componente, consulte [http://www.adobe.com/go/learn\\_fl\\_creating\\_components\\_es](http://www.adobe.com/go/learn_fl_creating_components_es).

## API de componentes ActionScript 3.0

Cada componente ActionScript 3.0 se crea en una clase de ActionScript 3.0 ubicada en una carpeta de paquete, que tiene un nombre con el formato `fl.nombrepaquete.nombreclase`. El componente Button, por ejemplo, es una instancia de la clase Button y tiene el nombre de paquete `fl.controls.Button`. Se debe hacer referencia al nombre de paquete al importar una clase de componente en la aplicación. Se podría importar la clase Button con la sentencia siguiente:

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

Para más información sobre la ubicación de los archivos de clase de componentes, consulte “[Trabajo con archivos de componente](#)” en la página 20.

Una clase de componente define los métodos, las propiedades, los eventos y los estilos que permiten interactuar con él en la aplicación. Los componentes de interfaz de usuario ActionScript 3.0 son subclases de las clases Sprite y UIComponent, y heredan propiedades, métodos y eventos de ellas. La clase Sprite es el bloque de creación básico de la lista de visualización y se asemeja a un componente MovieClip, pero no tiene línea de tiempo. La clase UIComponent es la clase base para todos los componentes visuales, tanto interactivos como no interactivos. La ruta de herencia de cada componente, así como sus propiedades, métodos, eventos y estilos se describen en la *Referencia de ActionScript 3.0 para Adobe Flash Professional*.

Todos los componentes ActionScript 3.0 utilizan el modelo de gestión de eventos de ActionScript 3.0. Para más información sobre la gestión de eventos, consulte “[Gestión de eventos](#)” en la página 25 y *Programación con ActionScript 3.0*.

## Trabajo con archivos de componente

En esta sección se explica dónde se almacenan los archivos de componente, dónde buscar los archivos de origen ActionScript, y cómo añadir y eliminar componentes del panel Componentes.

### Ubicación de almacenamiento de los archivos de componente

Los componentes de Flash se almacenan en la carpeta de configuración de nivel de aplicación.

*Nota:* para obtener información sobre estas carpetas, consulte “*Carpetas de configuración instaladas con Flash*” en *Uso de Flash*.

Los componentes se instalan en las siguientes ubicaciones:

- Windows 2000 o Windows XP: C:\Archivos de programa\Adobe\Adobe FlashCS5\idioma\Configuration\Components
- Mac OS X: Macintosh HD:Aplicaciones:Adobe Flash CS5:Configuration:Components

En la carpeta Components, los componentes de interfaz de usuario se encuentran en el archivo User Interface fla. Los componentes FLVPlayback (FLVPlaybackAS3.swc) y FLVPlaybackCaptioning están en la carpeta Video.

También se pueden almacenar componentes en las ubicaciones basadas en usuario siguientes:

- Windows 2000 o Windows XP: C:\Documents and Settings\usuario\Configuration local\Datos de programa\Adobe\Adobe Flash CS5\en\Configuration\Components
- Windows Vista: C:\Users\nombreusuario\Configuration local\Application Data\Adobe\Adobe Flash CS5\en\Configuration\nombrefiltro.xml

*Nota:* en Windows, la carpeta Datos de programa está oculta de forma predeterminada. Para mostrar carpetas y archivos ocultos, seleccione Mi PC para abrir Windows Explorer, seleccione Herramientas > Opciones de carpeta y después elija la ficha Ver. En la ficha Ver, seleccione el botón de opción Mostrar todos los archivos y carpetas ocultos.

- Mac OS X: Macintosh HD:Usuarios:<nombre de usuario>:Librería:Application Support:Adobe Flash CS5:Configuration:Components

## Ubicación de los archivos de origen de componente

Los archivos de clase ActionScript (.as) (o *archivos de origen*) para los componentes se instalan en las siguientes carpetas de aplicación de Windows 2000 o Windows XP:

**Componentes de interfaz de usuario** C:\Archivos de programa\Adobe\Adobe Flash CS5\en\Configuration\Component Source\ActionScript 3.0\User Interface\fl

**FLVPlayback** C:\Archivos de programa\Adobe\Adobe Flash CS5\en\Configuration\Component Source\ActionScript 3.0\FLVPlayback\fl\video

**FLVPlaybackCaptioning** C:\Archivos de programa\Adobe\Adobe Flash CS5\en\Configuration\Component Source\ActionScript 3.0\FLVPlaybackCaptioning\fl\video

En Mac OS X, los archivos de origen de componente se ubican aquí:

**Componentes de interfaz de usuario** Macintosh HD:Aplicaciones:Adobe Flash CS5:Configuration:Component Source>ActionScript 3.0>User Interface:fl

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

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

## Archivos de origen de componente y Classpath

Puesto que los componentes ActionScript 3.0 tienen su código compilado, no se debe especificar la ubicación de los archivos de clase ActionScript en la variable Classpath. Si se incluye su ubicación en Classpath, aumentará el tiempo necesario para compilar las aplicaciones. No obstante, si Flash encuentra archivos de clase en la configuración de Classpath, el archivo de clase siempre tendrá prioridad sobre el código compilado del componente.

Es posible que se desee añadir la ubicación de los archivos de origen de componente a la configuración de Classpath cuando se depure una aplicación con componentes. Para más información, consulte [“Depuración de aplicaciones de componentes”](#) en la página 22.

## Modificación de los archivos de componentes

Si se actualizan, se añaden o se eliminan componentes basados en SWC, o bien se añaden nuevos componentes basados en FLA a Flash, deben volverse a cargar en el panel Componentes para que estén disponibles. Los componentes pueden volverse a cargar reiniciando Flash o seleccionando Volver a cargar en el menú del panel Componentes. Esto hará que Flash detecte los componentes que se hayan añadido a la carpeta Components.

### Volver a cargar los componentes en el panel Componentes mientras se ejecuta Flash:

- Seleccione Volver a cargar en el menú del panel Componentes.

### Eliminación de un componente del panel Componentes:

- Elimine el archivo FLA, SWC, o MXP de la carpeta Components y reinicie Flash o seleccione Volver a cargar en el menú del panel Componentes. Un archivo MXP es un archivo de componente que se ha descargado de la zona de intercambio de Adobe.

Se pueden eliminar y reemplazar componentes basados en SWC mientras se ejecuta Flash. Si se vuelven a cargar, se reflejarán los cambios, pero si se modifican o se eliminan componentes basados en FLA, dichos cambios no se reflejarán hasta que se completen las operaciones y se reinicie Flash. No obstante, se pueden añadir componentes basados en FLA y cargarlos con el comando Volver a cargar.

 Adobe recomienda que primero se hagan copias de todos los archivos de componente Flash (.fla o .as) que se vayan a modificar. Después, si es necesario, se pueden restaurar.

## Depuración de aplicaciones de componentes

Los componentes ActionScript 3.0 contienen todo su código fuente para reducir el tiempo de compilación cuando se compile la aplicación. Sin embargo, el depurador de Flash no puede inspeccionar código dentro de clips compilados. Por tanto, si se desea depurar la aplicación en el código fuente del componente, se deben añadir los archivos de origen de dicho componente en la configuración de Classpath.

La ubicación de las carpetas de paquetes de componente es relativa a la ubicación de archivos de origen del tipo de componente. Para hacer referencia a todos los archivos de origen de ActionScript 3.0, añada la ubicación siguiente a la configuración de Classpath de los paquetes de interfaz de usuario:

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

**Nota:** esta operación sustituirá el código compilado de todos los componentes de interfaz de usuario y aumentará el tiempo de compilación de la aplicación. Si se ha cambiado el archivo de origen del componente por algún motivo, dicho componente podría mostrar un comportamiento diferente.

Para configurar la variable Classpath, seleccione Preferencias en el menú Edición y, a continuación, seleccione ActionScript en la lista de categorías y haga clic en el botón Configuración de ActionScript 3.0. Para añadir una nueva entrada, haga clic en el signo más situado encima de la ventana que muestra la configuración actual.

La variable \$(AppConfig) hace referencia a la carpeta Configuration de Flash CS5 en la ubicación en la que se ha instalado Flash CS5. Normalmente, la ruta de acceso tiene este aspecto:

- Windows 2000 o Windows XP: C:\Archivos de programa\Adobe\Adobe FlashCS 5\idioma\Configuration\
- Mac OS X: Macintosh HD:Aplicaciones:Adobe Flash CS5:Configuration

**Nota:** si se debe cambiar un archivo de origen de componente, Adobe recomienda encarecidamente copiar el archivo de origen original en una ubicación diferente y añadirla a Classpath.

Para más información sobre la ubicación de los archivos de origen de componentes, consulte [“Ubicación de los archivos de origen de componente”](#) en la página 21.

## Configuración de parámetros y propiedades

Todos los componentes tienen parámetros que se pueden definir para cambiar su apariencia y comportamiento. Un parámetro es una propiedad de la clase de componente, que aparece en el inspector de propiedades y en el inspector de componentes. Las propiedades que se utilizan con más frecuencia aparecen como parámetros de edición; las otras deben establecerse mediante código ActionScript. Todos los parámetros que se pueden definir durante la edición también se pueden definir mediante código ActionScript. Los valores de los parámetros establecidos mediante ActionScript sustituyen cualquier otro valor que se haya establecido durante la edición.

La mayoría de los componentes de interfaz de usuario ActionScript 3.0 heredan propiedades y métodos de la clase `UIComponent`, así como de una clase base. Por ejemplo, las clases `Button` y `CheckBox` heredan propiedades de la clase `UIComponent` y `BaseButton`. Se puede acceder a las propiedades heredadas del componente y a sus propias propiedades de clase. Por ejemplo, el componente `ProgressBar` hereda la propiedad `ProgressBar.enabled` de `UIComponent`, pero también tiene su propia propiedad `ProgressBar.percentComplete`. Se puede acceder a estas dos propiedades para interactuar con una instancia del componente `ProgressBar`. Para obtener más información sobre las propiedades de un componente, consulte la entrada de su clase en la [Referencia de ActionScript 3.0](#).

Se pueden definir parámetros de una instancia de componente con el inspector de propiedades o el inspector de componentes.

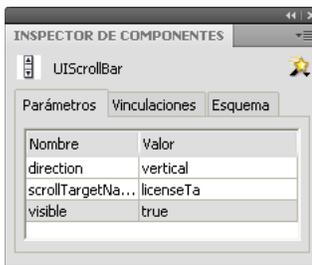
### Introducción de un nombre de instancia para un componente en el inspector de propiedades:

- 1 Seleccione Ventana > Propiedades > Propiedades.
- 2 Seleccione una instancia de componente en el escenario.
- 3 Introduzca un nombre para la instancia de componente en el cuadro <Nombre de instancia>, ubicado debajo de la lista desplegable Clip de película. Asimismo, puede hacer clic en la ficha Parámetros e introducir el nombre en el cuadro debajo de la palabra *Componente*. Introduzca los valores de los parámetros que desea definir.

Es recomendable añadir al nombre de la instancia un sufijo que indique el tipo de componente que es; esto facilitará la lectura del código ActionScript. Por ejemplo, el nombre de instancia es `licenseSb`, ya que el componente es una barra de desplazamiento que recorre un contrato de licencia en el área de texto `licenseTa`.

### Introducción de parámetros para una instancia de componente en el inspector de componentes:

- 1 Seleccione Ventana > Inspector de componentes.
- 2 Seleccione una instancia de componente en el escenario.
- 3 Haga clic en la ficha Parámetros e introduzca los valores de los parámetros mostrados.



Parámetros de componente en el inspector de componentes

## Configuración de propiedades de componentes en ActionScript

En ActionScript, el operador de punto (`.`) (sintaxis con puntos) permite acceder a propiedades y métodos que pertenecen a un objeto o instancia del escenario. Una expresión que utiliza la sintaxis con puntos empieza por el nombre de la instancia, seguido de un punto, y termina con el elemento que se desea especificar. Por ejemplo, el código ActionScript establece la propiedad `width` de la instancia de `CheckBox` `aCh` para que tenga una anchura de 50 píxeles:

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

La siguiente sentencia `if` comprueba si el usuario ha activado la casilla de verificación:

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

## Biblioteca

Al añadir por primera vez un componente a un documento, Flash lo importa como un clip de película en el panel Biblioteca. También se puede arrastrar un componente del panel Componentes directamente al panel Biblioteca y, a continuación, añadir una instancia de dicho componente al escenario. En cualquier caso, se debe añadir un componente a la biblioteca para poder acceder a sus elementos de clase.

Si se añade un componente a la biblioteca y se crea una instancia del mismo con ActionScript, primero se debe importar su clase con la sentencia `import`. En la sentencia `import`, se debe especificar el nombre de paquete del componente y su nombre de clase. Por ejemplo, la siguiente sentencia importa la clase `Button`:

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

Al colocar un componente en la biblioteca, Flash también importa una carpeta de sus activos, que contienen los aspectos de sus diferentes estados. Los *aspectos* del componente incluyen la colección de símbolos que forman su visualización gráfica en la aplicación. Un aspecto es la representación gráfica, o clip de película, que indica un estado determinado del componente.

El contenido de la carpeta `Component Assets` permite cambiar los aspectos del componente si desea realizar dicha operación. Para más información, consulte “[Personalización de los componentes de la interfaz de usuario](#)” en la página 102.

Una vez que el componente se encuentra en la biblioteca, se pueden añadir más instancias del mismo al documento arrastrando su icono al escenario, desde el panel Componentes o desde el panel Biblioteca.

## Ajuste del tamaño de los componentes

Utilice la herramienta Transformación libre o el método `setSize()` para cambiar el tamaño de las instancias de componente. Se puede llamar al método `setSize()` desde cualquier instancia de componente (consulte `UIComponent.setSize()`) para cambiar su tamaño. El código siguiente cambia el tamaño de una instancia del componente `List` a 200 píxeles de ancho por 300 píxeles de alto:

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

Un componente no cambia de tamaño automáticamente para ajustarse a su etiqueta. Si una instancia de componente que se ha añadido a un documento no es lo suficientemente grande para mostrar la etiqueta, el texto de la etiqueta aparece recortado. Debe cambiar el tamaño del componente para que quepa en su etiqueta.

Para obtener más información sobre el cambio de tamaño de los componentes, consulte las entradas concretas en la [Referencia de ActionScript 3.0 para Flash Professional](#).

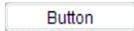
## Previsualización dinámica

La función Previsualización dinámica, que se activa de forma predeterminada, permite ver los componentes del escenario tal como aparecerán en el contenido de Flash publicado, con su tamaño aproximado.

Para activar o desactivar la función Previsualización dinámica:

- Seleccione **Control > Activar vista previa dinámica**. Si aparece una marca de verificación junto a la opción, indica que está activada.

La previsualización dinámica refleja parámetros diferentes para componentes diferentes. Para obtener información sobre qué parámetros de componente se reflejan en la previsualización dinámica, consulte la entrada de cada componente en la [Referencia de ActionScript 3.0 para Flash Professional](#).



Componente Button con previsualización dinámica activada



Componente Button con previsualización dinámica desactivada

En la previsualización dinámica, los componentes no son funcionales. Para probar la funcionalidad de los componentes, se debe utilizar el comando Control > Probar película.

## Gestión de eventos

Cada componente difunde eventos cuando el usuario interactúa con él. Cuando el usuario hace clic en un componente Button, por ejemplo, distribuye un evento `MouseEvent.CLICK`; cuando el usuario selecciona un elemento en un componente List, éste distribuye un evento `Event.CHANGE`. También se puede producir un evento cuando le sucede algo significativo a un componente, por ejemplo, cuando termina de cargarse el contenido de una instancia de `UILoader` y se genera un evento `Event.COMPLETE`. Para gestionar un evento, escriba código ActionScript que se ejecute cuando se produzca el evento.

Los eventos de un componente incluyen los eventos de cualquier clase de la que hereda el componente. Esto implica que todos los componentes de interfaz de usuario ActionScript 3.0 heredan eventos de la clase `UIComponent`, ya que ésta constituye la clase base de dichos componentes. Para consultar una lista con los eventos que emite un componente, consulte la sección Eventos de la entrada de la clase del componente en la [Referencia de ActionScript 3.0 para Flash Professional](#).

Para obtener una explicación exhaustiva de la gestión de eventos en ActionScript 3.0, consulte *Programación con ActionScript 3.0*.

## Detectores de eventos

Los siguientes puntos clave se aplican a la gestión de eventos de componentes ActionScript 3.0:

- Todos los eventos se difunden a través de una instancia de una clase de componente. La instancia del componente es el *difusor*.
- Un *detector* de eventos se registra llamando al método `addEventListener()` de la instancia del componente. Por ejemplo, la línea de código siguiente añade un detector para el evento `MouseEvent.CLICK` a la instancia de Button `aButton`:

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

El segundo parámetro del método `addEventListener()` registra el nombre de la función, `clickHandler`, que se llamará cuando se produzca el evento. Esta función también se conoce como *funcióncallback*.

- Se pueden registrar varios detectores en una instancia de componente.

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

- Se puede registrar un detector en varias instancias de componente.

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

- La función de gestión de eventos se pasa a un objeto de evento que contiene información sobre el tipo de evento y la instancia que difunden el evento. Para obtener más información, consulte “Objeto de evento” en la página 26.
- El detector se mantiene activo hasta que termine la aplicación, o bien hasta que se elimine explícitamente con el método `removeEventListener()`. Por ejemplo, la línea de código siguiente elimina el detector para el evento `MouseEvent.CLICK` en `aButton`:

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

## Objeto de evento

El objeto de evento hereda de la clase de objeto `Event` y tiene propiedades que contienen información sobre el evento que se ha producido, incluidas las propiedades `target` y `type`, que proporcionan información esencial sobre el evento:

Propiedad	Descripción
<code>type</code>	Cadena que indica el tipo de evento.
<code>target</code>	Referencia a la instancia del componente que difunde el evento.

Si un evento tiene más propiedades, aparecen en la descripción de la clase del evento en la [Referencia de ActionScript 3.0 para Flash Professional](#).

El objeto de evento se genera automáticamente y se pasa a la función de gestión de eventos cuando se produce un evento.

El objeto de evento se puede utilizar dentro de la función para acceder al nombre del evento que se difundió o al nombre de instancia del componente que difundió el evento. A partir del nombre de la instancia, puede acceder a otras propiedades de componente. Por ejemplo, el código siguiente utiliza la propiedad `target` del objeto de evento `evtObj` para acceder a la propiedad `label` de `aButton` y mostrarla en el panel Salida:

```
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");  
}
```

## Trabajo con la lista de visualización

Todos los componentes ActionScript 3.0 heredan la clase `DisplayObject` y, por tanto, tienen acceso a sus métodos y propiedades para interactuar con la lista de visualización. La *lista de visualización* es la jerarquía de objetos mostrados y elementos visuales de una aplicación. Esta jerarquía incluye los elementos siguientes:

- El escenario, que es el contenedor de nivel superior.
- Los objetos de visualización, que incluyen formas, clips de película y campos de texto, entre otros.

- Los contenedores de objetos de visualización, que son tipos especiales de objetos de visualización que pueden incluir objetos de visualización secundarios.

El orden de los objetos de la lista de visualización determina su profundidad en el contenedor principal. La profundidad de un objeto hace referencia a su posición de arriba abajo, o de delante atrás en el escenario o en su contenedor de visualización. El orden de profundidad es aparente cuando los objetos se solapan, pero también existe aunque no se solapan. Cada objeto de la lista de visualización tiene una profundidad correspondiente en el escenario. Si se desea cambiar la profundidad de un objeto colocándolo delante o detrás de otros objetos, se debe cambiar su posición en la lista de visualización. El orden predeterminado de los objetos en la lista de visualización es el orden en el que se colocan en el escenario. La posición 0 de la lista de visualización es el objeto situado en la parte inferior del orden de profundidad.

## Añadir un componente a la lista de visualización

Se puede añadir un objeto a un objeto `DisplayObjectContainer` llamando al método `addChild()` o `addChildAt()` del contenedor. En el caso del escenario, también se puede añadir un objeto a su lista de visualización durante la edición creándolo o, en el caso de los componentes, arrastrándolo al escenario del panel Componentes. Para añadir un objeto a un contenedor con `ActionScript`, cree primero una instancia invocando su constructor con el operador `new` y, a continuación, llame al método `addChild()` o `addChildAt()` para colocarlo en el escenario y en la lista de visualización. El método `addChild()` coloca el objeto en la siguiente posición de la lista de visualización, mientras que `addChildAt()` especifica la posición en la que se añadirá el objeto. Si se especifica una posición que ya está ocupada, el objeto de dicha posición, y los de posiciones superiores, se mueven 1 posición hacia arriba. La propiedad `numChildren` de un objeto `DisplayObjectContainer` incluye el número de objetos de visualización que contiene. Se puede recuperar un objeto de la lista de visualización llamando al método `getChildAt()` y especificando la posición o, si se conoce el nombre del objeto, llamando al método `getChildByName()`.

**Nota:** si se añade un componente con `ActionScript`, se debe asignar un nombre a su propiedad `name`, en el caso de que se desee acceder a él mediante nombre en la lista de visualización.

En el ejemplo siguiente se muestran los nombres y las posiciones de los tres componentes en la lista de visualización. En primer lugar, arrastre un componente `NumericStepper`, un componente `Button` y un componente `ComboBox` al escenario para que se solapen entre sí, y asígneles los nombres de instancia `aNs`, `aButton` y `aCb`. Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo:

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

Se deberán ver las líneas siguientes en el panel Salida:

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

## Desplazamiento de un componente en la lista de visualización

Se puede cambiar la posición de un objeto en la lista de visualización, así como su profundidad de visualización, llamando al método `addChildAt()` y proporcionando el nombre de objeto y la posición en la que se desea colocar como parámetros del método. Por ejemplo, añada el código siguiente al ejemplo anterior para colocar `NumericStepper` en la parte superior y repetir el bucle para mostrar las nuevas posiciones del componente en la lista de visualización:

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

Se deberá ver lo siguiente en el panel Salida:

```
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
```

El componente NumericStepper también debe aparecer delante de los otros componentes en la pantalla.

Tenga en cuenta que `numChildren` es el número de objetos (de 1 a  $n$ ) de la lista de visualización, mientras que la primera posición de la lista es 0. De este modo, si hay tres objetos en la lista, la posición de índice del tercer objeto es 2. Esto significa que se puede hacer referencia a la última posición de la lista de visualización, o bien al objeto superior en cuanto a profundidad de visualización, como `numChildren - 1`.

## Eliminación de un componente de la lista de visualización

Se puede eliminar un componente de un contenedor de objeto de visualización y su lista de visualización con los métodos `removeChild()` y `removeChildAt()`. En el ejemplo siguiente, se colocan tres componentes Button, uno delante de otro, en el escenario, y se añade un detector de eventos para cada uno de ellos. Al hacer clic en cada componente Button, el detector de eventos lo elimina de la lista de visualización y del escenario.

- 1 Cree un documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre un componente Button del panel Componentes al panel Biblioteca.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código siguiente:

```
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 obtener una explicación completa de la lista de visualización, consulte “Programación de visualización” en *Programación con ActionScript 3.0*.

## Trabajo con FocusManager

Cuando el usuario presiona la tecla Tabulador para navegar en una aplicación de Flash o hace clic en otra aplicación, la clase FocusManager determina el componente que recibe la selección de entrada. No es preciso añadir una instancia de FocusManager a una aplicación, ni escribir código para activar FocusManager, a menos que se cree un componente.

Si se selecciona un objeto RadioButton, FocusManager examina el objeto, así como todos los objetos con el mismo valor de `groupName`, y define la selección del objeto estableciendo el valor `true` para la propiedad `selected`.

Cada componente de ventana modal contiene una instancia de FocusManager, de forma que los controles de esa ventana se conviertan en su propio grupo de tabulación. Esto evita que un usuario desplace accidentalmente la selección a componentes de otras ventanas al presionar la tecla Tabulador.

FocusManager utiliza el nivel de profundidad (u orden *z*) de los elementos en el contenedor como esquema de navegación predeterminado o *bucle de tabulación*. Un usuario normalmente navega por el bucle de tabulación mediante el tabulador, que mueve la selección desde el primer componente seleccionado al último y luego de vuelta al primero. Los niveles de profundidad se configuran principalmente con el orden en el que los componentes se arrastran al escenario. Sin embargo, también se pueden utilizar los comandos Modificar > Organizar > Traer al frente/Enviar al fondo para determinar el orden *z* final. Para más información sobre los niveles de profundidad, consulte “[Trabajo con la lista de visualización](#)” en la página 26.

Se puede llamar al método `setFocus()` para seleccionar una instancia de un componente en una aplicación. Por ejemplo, en el siguiente ejemplo se crea una instancia de FocusManager para el contenedor actual (`this`) y se selecciona la instancia de Button `aButton`.

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

Para determinar el componente seleccionado, se puede llamar al método `getFocus()` y para determinar el componente del bucle de tabulación que se seleccionará a continuación, se puede llamar al método `getNextFocusManagerComponent()`. En el siguiente ejemplo, hay tres componentes en el escenario: un CheckBox, un RadioButton y un Button; cada uno tiene detectores de los eventos `MouseEvent.CLICK` y `FocusEvent.MOUSE_FOCUS_CHANGE`. Cuando se produce el evento `MouseEvent.CLICK`, porque el usuario ha hecho clic en el componente, la función `showFocus()` llama al método `getNextFocusManagerComponent()` para determinar el componente del bucle de tabulación que se seleccionará a continuación. A continuación, llama al método `setFocus()` para seleccionar dicho componente. Cuando se produce el evento `FocusEvent.MOUSE_FOCUS_CHANGE`, la función `fc()` muestra el nombre del componente en el que se produjo el evento. Este evento se activa cuando el usuario hace clic en un componente que no sea el componente siguiente en el bucle de tabulación.

```
// 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 crear un componente Button que reciba la selección cuando el usuario presiona Intro (Windows) o Retorno (Macintosh), debe establecerse la propiedad `FocusManager.defaultButton` en la instancia de Button que se prefiera como opción predeterminada, como en el código siguiente:

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

La clase `FocusManager` sustituye el rectángulo de selección de Flash Player predeterminado y dibuja un rectángulo de selección personalizado con esquinas redondeadas.

Para obtener más información sobre la creación de un esquema de selección en una aplicación de Flash, consulte la clase `FocusManager` en la [Referencia de ActionScript 3.0 para Flash Professional](#). Para crear un administrador de selección personalizado, se debe crear una clase que implemente la interfaz `IFocusManager`. Para obtener más información, consulte `IFocusManager` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Trabajo con componentes basados en List

Los componentes `List`, `DataGrid` y `TileList` heredan de la clase base `SelectableList`. Por este motivo, estos componentes se consideran componentes basados en `List`. Un componente `ComboBox` consta de un cuadro de texto y un componente `List`, de manera que también se considera un componente basado en `List`.

List consta de filas. DataGrid y TileList constan de filas que se pueden dividir en varias columnas. La intersección de una fila y una columna es una celda. En un componente List, que está formado por una sola columna de filas, cada fila es una celda. Una celda tiene los siguientes dos aspectos importantes:

- Los valores de datos que contienen las celdas se denominan elementos. Un *elemento* es un objeto de ActionScript que sirve para almacenar las unidades de información en un componente List. Un componente List puede concebirse como un conjunto; cada espacio indexado del conjunto es un elemento. En un componente List, un elemento es un objeto que, normalmente, tiene una propiedad `label` que se muestra y una propiedad `data` que se utiliza para almacenar datos. Un *proveedor de datos* es un modelo de datos de los elementos de un componente List. Un proveedor de datos permite llenar un componente basado en List asignándolo a la propiedad `dataProvider` del componente.
- Una celda puede contener diferentes tipos de datos, ya sean de texto o imágenes, clips de película o cualquier otra clase que se pueda crear. Por esta razón, una celda debe dibujarse o representarse de forma que resulte apropiada para su contenido. En consecuencia, los componentes basados en List tienen un *procesador de celdas* para mostrar sus celdas. En el caso de DataGrid, cada columna es un objeto `DataGridColumn`, que también tiene una propiedad `cellRenderer`; cada columna se puede representar de forma adecuada para su contenido.

Todos los componentes basados en List tienen propiedades `cellRenderer` y `dataProvider` que se pueden configurar para cargar y representar estos componentes. Para obtener información sobre la utilización de estas propiedades y trabajar con componentes basados en List, consulte “[Trabajo con DataProvider](#)” en la página 31 y “[Trabajo con CellRenderer](#)” en la página 39.

## Trabajo con DataProvider

`DataProvider` es un origen de datos que se puede utilizar para proporcionar datos a los componentes `ComboBox`, `DataGrid`, `List` y `TileList`. Cada una de estas clases de componente tiene una propiedad `dataProvider` a la que se puede asignar un objeto `DataProvider` para llenar con datos las celdas del componente. Normalmente, un proveedor de datos es una colección de datos como, por ejemplo, un objeto `Array` o `XML`.

### Creación de un objeto DataProvider

Para los componentes `ComboBox`, `List` y `TileList` se puede crear un objeto `DataProvider` mediante el parámetro `dataProvider` en el entorno de edición. El componente `DataGrid` no tiene un parámetro `dataProvider` en el inspector de propiedades, ya que puede tener varias columnas y, en consecuencia, su proveedor de datos es más complejo. Asimismo, se puede utilizar ActionScript para crear un objeto `DataProvider` para estos componentes y para `DataGrid`.

### Uso del parámetro dataProvider

Se puede crear un proveedor de datos sencillo para los componentes `ComboBox`, `List` y `TileList` haciendo clic en el parámetro `dataProvider` de la ficha `Parámetros`, en el inspector de propiedades o en el inspector de componentes.

Si se hace doble clic en la celda de valor, que inicialmente muestra un objeto `Array` vacío, se abrirá el cuadro de diálogo `Valores`, que permite introducir varios valores de etiqueta y datos para crear el proveedor de datos.



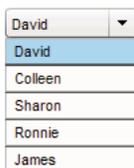
Cuadro de diálogo Valores para dataProvider

Haga clic en el signo más para añadir un elemento a dataProvider. Haga clic en el signo menos para eliminar un elemento. Haga clic en la tecla de flecha arriba para subir un elemento seleccionado por la lista, o bien haga clic en la tecla de flecha abajo para bajarlo. En la ilustración siguiente se muestra un cuadro de diálogo Valores que crea una lista de nombres y cumpleaños de niños.



Cuadro de diálogo Valores con datos

El objeto Array que se crea consta de pares de campos de etiqueta y valor. Los campos de etiqueta son `label` y `data` y los campos de valor son los nombres de los niños y sus cumpleaños. El campo de etiqueta identifica el contenido que aparece en List, que en este caso está formado por los nombres de los niños. El componente ComboBox resultante tiene el siguiente aspecto:



Componente ComboBox llenado por DataProvider

Cuando termine de añadir los datos, haga clic en Aceptar para cerrar el cuadro de diálogo. El objeto Array del parámetro dataProvider ya está lleno con los elementos que se han creado.

**Trabajo con componentes**

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

*Parámetro dataProvider con datos*

Se puede acceder a los valores de etiqueta y datos creados utilizando ActionScript para acceder a la propiedad `dataProvider` del componente.

**Creación de un objeto DataProvider con ActionScript**

Se puede crear un objeto `DataProvider` creando los datos en un objeto `Array` o `XML` y proporcionando el objeto como el parámetro `value` al constructor de `DataProvider`.

**Nota:** en ActionScript 3.0, no se puede asignar un objeto `Array` o `XML` directamente a una propiedad `dataProvider`, ya que la propiedad se define como objeto `DataProvider` y sólo puede recibir un objeto de tipo `DataProvider`.

En el ejemplo siguiente se llena un componente `List`, que es una columna de filas, con los nombres de varios niños y sus cumpleaños. En el ejemplo se define la lista en el objeto `Array` `items` y se proporciona como parámetro cuando se crea la instancia de `DataProvider` (`new DataProvider(items)`) y se asigna a la propiedad `dataProvider` del 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);
```

El objeto `Array` consta de pares de campos de etiqueta y valor. Los campos de etiqueta son `label` y `data` y los campos de valor son los nombres de los niños y sus cumpleaños. El campo de etiqueta identifica el contenido que aparece en `List`, que en este caso está formado por los nombres de los niños. El componente `List` resultante tiene el siguiente aspecto:

David
Colleen
Sharon
Ronnie
James

*Componente List llenado por DataProvider*

El valor del campo de datos está disponible cuando el usuario selecciona un elemento de la lista haciendo clic en él o desencadenando un evento `change`. En el ejemplo siguiente se añade un componente `TextArea` (`aTa`) y un controlador de eventos (`changeHandler`) al ejemplo anterior para mostrar el cumpleaños del niño cuando el usuario selecciona un nombre del componente `List`.

```
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);

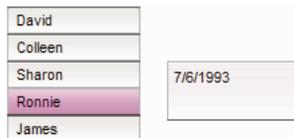
addChild(aList);
addChild(aTa);

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

aList.addEventListener(Event.CHANGE, changeHandler);

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

A continuación, cuando el usuario selecciona el nombre del niño en List, el cumpleaños de dicho niño se muestra en TextArea tal como se muestra en la ilustración siguiente. Esto se consigue con la función `changeHandler()` cuando establece la propiedad `text` de `TextArea` (`aTa.text`) en el valor del campo de datos del elemento seleccionado (`event.target.selectedItem.data`). La propiedad `event.target` es el objeto que ha activado el evento, que en este caso es el componente `List`.



Visualización del campo de datos de `DataProvider` de un componente `List`

Se pueden incluir datos que no sean texto en `DataProvider`. En el ejemplo siguiente se incluyen componentes `MovieClip` en un `DataProvider` que proporciona datos a un componente `TileList`. Se crea el objeto `DataProvider` llamando a `addItem()` para añadir todos los elementos una vez que se crea el componente `MovieClip`, un cuadro de color.

```

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();
}

```

También se pueden utilizar datos XML (en lugar de un conjunto) para llenar un objeto DataProvider. Por ejemplo, en el código siguiente se almacenan datos en un objeto XML denominado `employeesXML`, y luego se pasa dicho objeto como el parámetro de valor de la función constructora `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;

```

Se pueden proporcionar datos como atributos de los datos XML, como en el código anterior, o bien como propiedades de dichos datos, tal como se muestra en el código siguiente:

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

El objeto `DataProvider` también tiene un conjunto de métodos y propiedades que permiten acceder a él y manipularlo. Se puede utilizar la API de `DataProvider` para añadir, eliminar, reemplazar, ordenar y combinar elementos en un objeto `DataProvider`.

## Manipulación de un objeto `DataProvider`

Se pueden añadir elementos a un objeto `DataProvider` con los métodos `addItem()` y `addItemAt()`. En el ejemplo siguiente se añaden elementos que introduce el usuario en el campo de texto de un componente `ComboBox` editable. Se supone que el componente `ComboBox` se ha arrastrado al escenario y se le ha asignado el nombre de instancia `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);
}
```

Asimismo, se pueden reemplazar y eliminar elementos de un componente a través de su `DataProvider`. En el ejemplo siguiente se implementan dos componentes `List` diferentes, `listA` y `listB`, y se proporciona un componente `Button` denominado `Sync`. Cuando el usuario hace clic en `Button`, el ejemplo utiliza el método `replaceItemAt()` para reemplazar los elementos de `listB` con los elementos de `listA`. Si la longitud de `listA` es mayor que la de `listB`, el ejemplo llama al método `addItem()` para añadir elementos adicionales a `listB`. Si la longitud de `listB` es mayor que la de `listA`, el ejemplo llama al método `removeItemAt()` para eliminar los elementos adicionales 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++);
    }
}
}
```

También se pueden realizar combinaciones con un objeto `DataProvider` y ordenarlo mediante los métodos `merge()`, `sort()` y `sortOn()`. En el ejemplo siguiente se llenan dos instancias de `DataGrid` (`aDg` y `bDg`) con calendarios parciales de dos equipos de fútbol. Se añade un componente `Button` con la etiqueta `Combinar` y cuando el usuario hace clic en él, el controlador de eventos (`mrgHandler`) combina el calendario de `bDg` con el calendario de `aDg`, y ordena el componente `DataGrid` resultante en la columna `Nombre`.

```
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 obtener más información, consulte la clase `DataProvider` en la [Referencia de ActionScript 3.0](#).

## Trabajo con CellRenderer

`CellRenderer` es una clase que utilizan los componentes basados en `List` (por ejemplo, `List`, `DataGrid`, `TileList` y `ComboBox`) para manipular y mostrar contenido de celda personalizado para sus filas. Las celdas personalizadas pueden contener texto, un componente creado previamente (como un componente `CheckBox`) o cualquier clase de objeto de visualización que se cree. Para representar datos con una clase `CellRenderer` personalizada, se puede ampliar dicha clase o implementar la interfaz `ICellRenderer` para crear una clase `CellRenderer` propia.

Las clases `List`, `DataGrid`, `TileList` y `ComboBox` son subclases de la clase `SelectableList`. La clase `SelectableList` incluye un estilo `cellRenderer`. Este estilo define el objeto de visualización que el componente utiliza para representar celdas.

Se puede ajustar el formato de los estilos que utiliza `CellRenderer` llamando al método `setRendererStyle()` del objeto de `List` (consulte “[Formato de celdas](#)” en la página 39). O bien, se puede definir una clase personalizada para utilizar como `CellRenderer` (consulte “[Definición de una clase CellRenderer personalizada](#)” en la página 40).

### Formato de celdas

La clase `CellRenderer` incluye un número de estilos que permiten controlar el formato de la celda.

Los estilos siguientes permiten definir los aspectos utilizados para los diferentes estados de la celda (desactivado, presionado, sobre y arriba):

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

Se aplican los siguientes estilos al formato de texto:

- `disabledTextFormat`

- `textFormat`
- `textPadding`

Se pueden establecer estos estilos llamando al método `setRendererStyle()` del objeto `List`, o bien al método `setStyle()` del objeto `CellRenderer`. Se pueden obtener estos estilos llamando al método `getRendererStyle()` del objeto `List`, o bien al método `getStyle()` del objeto `CellRenderer`. Asimismo, se puede acceder a un objeto que define todos los estilos del procesador (como las propiedades con nombre del objeto) mediante la propiedad `rendererStyles` del objeto `List` o el método `getStyleDefinition()` del objeto `CellRenderer`.

Se puede llamar al método `clearRendererStyle()` para restablecer el valor predeterminado de un estilo.

Para obtener o establecer la altura de las filas de la lista, utilice la propiedad `rowHeight` del objeto `List`.

## Definición de una clase `CellRenderer` personalizada

### Creación de una clase que amplíe la clase `CellRenderer` para definir la personalización de dicha clase

Por ejemplo, el código siguiente incluye dos clases. La clase `ListSample` crea una instancia de un componente `List` y utiliza la otra clase, `CustomRenderer`, para definir el procesador de celdas que se usará para el componente `List`. La clase `CustomRenderer` amplía la clase `CellRenderer`.

- 1 Seleccione Archivo > Nuevo.
- 2 En el cuadro de diálogo Nuevo documento que se muestra, seleccione Archivo de Flash (ActionScript 3.0) y, a continuación, haga clic en Aceptar.
- 3 Seleccione Ventana > Componentes para mostrar el panel Componentes.
- 4 En el panel Componentes, arrastre un componente `List` al escenario.
- 5 Si Flash no muestra el inspector de propiedades, seleccione Ventana > Propiedades > Propiedades.
- 6 Con el componente `List` seleccionado, establezca las propiedades en el inspector de propiedades:
  - Nombre de instancia: `myList`
  - An. (anchura) 200
  - Al. (altura) 300
  - X: 20
  - Y: 20
- 7 Seleccione el fotograma 1 de la capa 1 en la línea de tiempo y elija Ventana > Acciones.
- 8 Introduzca el script siguiente en el panel Acciones:

```
myList.setStyle("cellRenderer", CustomCellRenderer);  
myList.addItem({label:"Burger -- $5.95"});  
myList.addItem({label:"Fries -- $1.95"});
```
- 9 Seleccione Archivo > Guardar. Asigne un nombre al archivo y haga clic en el botón Aceptar.
- 10 Seleccione Archivo > Nuevo.
- 11 En el cuadro de diálogo Nuevo documento que se muestra, seleccione Archivo ActionScript y, a continuación, haga clic en Aceptar.
- 12 En la ventana del script, introduzca el código siguiente para definir la clase `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 Seleccione Archivo > Guardar. Asigne el nombre CustomCellRenderer.as al archivo, colóquelo en el mismo directorio que el archivo FLA y haga clic en el botón Aceptar.

14 Seleccione Control > Probar película.

### Uso de una clase que implemente la interfaz ICellRenderer para definir una clase CellRenderer personalizada

También se puede definir una clase CellRenderer con cualquier clase que herede la clase DisplayObject e implemente la interfaz ICellRenderer. Por ejemplo, el código siguiente define dos clases. La clase ListSample2 añade un objeto List a la lista de visualización y define su CellRenderer para utilizar la clase CustomRenderer. La clase CustomRenderer amplía la clase CheckBox (que a su vez amplía la clase DisplayObject) e implementa la interfaz ICellRenderer. Debe tenerse en cuenta que la clase CustomRenderer define los métodos getter y setter para las propiedades data y listData, definidas en la interfaz ICellRenderer. Los otros métodos y propiedades definidos en la interfaz ICellRenderer (la propiedad selected y el método setSize()) ya están definidos en la clase CheckBox:

- 1 Seleccione Archivo > Nuevo.
- 2 En el cuadro de diálogo Nuevo documento que se muestra, seleccione Archivo de Flash (ActionScript 3.0) y, a continuación, haga clic en Aceptar.
- 3 Seleccione Ventana > Componentes para mostrar el panel Componentes.
- 4 En el panel Componentes, arrastre un componente List al escenario.
- 5 Si Flash no muestra el inspector de propiedades, seleccione Ventana > Propiedades > Propiedades.
- 6 Con el componente List seleccionado, establezca las propiedades en el inspector de propiedades:

- Nombre de instancia: myList
- An. (anchura) 100
- Al. (altura) 300
- X: 20
- Y: 20

7 Seleccione el fotograma 1 de la capa 1 en la línea de tiempo y elija Ventana > Acciones.

8 Introduzca el script siguiente en el panel Acciones:

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

9 Seleccione Archivo > Guardar. Asigne un nombre al archivo y haga clic en el botón Aceptar.

10 Seleccione Archivo > Nuevo.

11 En el cuadro de diálogo Nuevo documento que se muestra, seleccione Archivo ActionScript y, a continuación, haga clic en Aceptar.

12 En la ventana del script, introduzca el código siguiente para definir la clase 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 Seleccione Archivo > Guardar. Asigne el nombre CustomCellRenderer.as al archivo, colóquelo en el mismo directorio que el archivo FLA y haga clic en el botón Aceptar.

14 Seleccione Control > Probar película.

### Uso de un símbolo para definir una clase CellRenderer

Asimismo, se puede utilizar un símbolo en la biblioteca para definir una clase CellRenderer. El símbolo debe exportarse para ActionScript, y el nombre de clase del símbolo de la biblioteca debe tener un archivo de clase asociado que implemente la interfaz ICellRenderer o que amplíe la clase CellRenderer (o una de sus subclases).

En el ejemplo siguiente se define una clase CellRenderer personalizada con un símbolo de biblioteca.

- 1 Seleccione Archivo > Nuevo.
- 2 En el cuadro de diálogo Nuevo documento que se muestra, seleccione Archivo de Flash (ActionScript 3.0) y, a continuación, haga clic en Aceptar.
- 3 Seleccione Ventana > Componentes para mostrar el panel Componentes.
- 4 En el panel Componentes, arrastre un componente List al escenario.
- 5 Si Flash no muestra el inspector de propiedades, seleccione Ventana > Propiedades > Propiedades.
- 6 Con el componente List seleccionado, establezca las propiedades en el inspector de propiedades:
  - Nombre de instancia: myList
  - An. (anchura) 100
  - Al. (altura) 400

- X: 20
  - Y: 20
- 7 Haga clic en el panel Parámetros y, a continuación, haga doble clic en la fila dataProvider de la segunda columna.
  - 8 En el cuadro de diálogo Valores que se muestra, haga clic en el signo más dos veces para añadir dos elementos de datos (con etiquetas establecidas en label0 y label1) y después haga clic en el botón Aceptar.
  - 9 Con la herramienta Texto, dibuje un campo de texto en el escenario.
  - 10 Con el campo de texto seleccionado, establezca las propiedades en el inspector de propiedades:
    - Tipo de texto: texto dinámico
    - Nombre de instancia: textField
    - An. (anchura) 100
    - Tamaño de fuente: 24
    - X: 0
    - Y: 0
  - 11 Seleccione el campo de texto seleccionado y, a continuación, elija Modificar > Convertir en símbolo.
  - 12 En el cuadro de diálogo Convertir en símbolo, configure las siguientes opciones y, a continuación, haga clic en Aceptar.
    - Nombre: MyCellRenderer
    - Tipo: MovieClip
    - Exportar para ActionScript: Seleccionado
    - Exportar en primer fotograma: Seleccionado
    - Clase: MyCellRenderer
    - Clase base: flash.display.MovieClip

Si Flash muestra una advertencia de clase ActionScript, haga clic en el botón Aceptar en el cuadro de advertencia.
  - 13 Elimine del escenario la instancia del nuevo símbolo de clip de película.
  - 14 Seleccione el fotograma 1 de la capa 1 en la línea de tiempo y elija Ventana > Acciones.
  - 15 Introduzca el script siguiente en el panel Acciones:

```
myList.setStyle("cellRenderer", MyCellRenderer);
```
  - 16 Seleccione Archivo > Guardar. Asigne un nombre al archivo y haga clic en el botón Aceptar.
  - 17 Seleccione Archivo > Nuevo.
  - 18 En el cuadro de diálogo Nuevo documento que se muestra, seleccione Archivo ActionScript y, a continuación, haga clic en Aceptar.
  - 19 En la ventana del script, introduzca el código siguiente para definir la clase 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 Seleccione Archivo > Guardar. Asigne el nombre MyCellRenderer.as al archivo, colóquelo en el mismo directorio que el archivo FLA y haga clic en el botón Aceptar.

21 Seleccione Control > Probar película.

## Propiedades de CellRenderer

La propiedad `data` es un objeto que contiene todas las propiedades que se establecen para `CellRenderer`. Por ejemplo, en la clase siguiente, que define una clase `CellRenderer` personalizada que amplía la clase `CheckBox`, tenga en cuenta que la función setter de la propiedad `data` pasa el valor de `data.label` a la propiedad `label` (heredada de la clase `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;
    }
}
}
```

La propiedad `selected` define si una celda está seleccionada en la lista.

## Aplicación de CellRenderer en una columna del objeto DataGrid

Un objeto `DataGrid` puede tener varias columnas y se pueden especificar distintos procesadores de celdas para cada columna. Todas las columnas de un objeto `DataGrid` se representan con un objeto `DataGridColumn`, y la clase `DataGridColumn` incluye una propiedad `cellRenderer`, para la que se puede definir la clase `CellRenderer` para la columna.

## Definición de CellRenderer para una celda editable

La clase `DataGridCellEditor` define un procesador que se utiliza para celdas editables en un objeto `DataGrid`. Se convierte en el procesador de una celda cuando la propiedad `editable` del objeto `DataGrid` se establece en `true` y, a continuación, el usuario hace clic en la celda que se va a editar. Para definir una clase `CellRenderer` para la celda editable, establezca la propiedad `itemEditor` de cada elemento del conjunto `columns` del objeto `DataGrid`.

## Uso de una imagen, un archivo SWF o un clip de película como CellRenderer

La clase `ImageCell`, subclase de `CellRenderer`, define un objeto que se utiliza para representar celdas en las que el contenido principal de la celda es una imagen, un archivo SWF o un clip de película. La clase `ImageCell` incluye los estilos siguientes para definir el aspecto de la celda:

- `imagePadding`: relleno que separa el borde de la celda del borde de la imagen, expresado en píxeles.
- `selectedSkin`: aspecto que se utiliza para indicar el estado seleccionado.
- `textOverlayAlpha`: opacidad de la superposición detrás de la etiqueta de la celda.

- `textPadding`: relleno que separa el borde de la celda del borde del texto, expresado en píxeles.

La clase `ImageCell` es la clase `CellRenderer` predeterminada de la clase `TileList`.

## Hacer los componentes accesibles

El contenido visual de las aplicaciones Flash puede hacerse accesible para los usuarios con discapacidades visuales mediante un lector de pantalla, que proporciona una descripción de audio del contenido de la pantalla. Para obtener información sobre el acceso de un lector de pantalla a la aplicación Flash, consulte el Capítulo 18, "Creación de contenido accesible" en *Uso de Flash*.

Para que un componente ActionScript 3.0 sea accesible para un lector de pantalla, también se debe importar su clase de accesibilidad y llamar al método `enableAccessibility()` de dicha clase. Un lector de pantalla puede tener acceso a los siguientes componentes ActionScript 3.0:

Componente	Clase de accesibilidad
Button	ButtonAccImpl
CheckBox	CheckBoxAccImpl
ComboBox	ComboBoxAccImpl
List	ListAccImpl
RadioButton	RadioButtonAccImpl
TileList	TileListAccImpl

Las clases de accesibilidad de componente se encuentran en el paquete `fl.accessibility`. Para que un lector de pantalla pueda acceder a un componente `CheckBox`, por ejemplo, se deberán añadir las siguientes sentencias a la aplicación:

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

La accesibilidad de un componente sólo se activa una vez, sea cual sea su número de instancias.

**Nota:** si se permite la accesibilidad de forma marginal, se aumenta el tamaño de archivo, ya que se incluyen las clases requeridas durante la compilación.

También se puede navegar por la mayoría de componentes a través del teclado. Para obtener más información sobre la activación de componentes accesibles y la navegación con el teclado, consulte las secciones Interacción con el usuario de “Uso de los componentes de la interfaz de usuario” en la página 47 y las clases de accesibilidad en la [Referencia de ActionScript 3.0 para Flash Professional](#).

# Capítulo 4: Uso de los componentes de la interfaz de usuario

En este capítulo se explica cómo utilizar los componentes de interfaz de usuario de ActionScript 3.0 que se incluyen en Flash.

## Uso del componente Button

El componente Button es un botón rectangular y de tamaño modificable que los usuarios pueden presionar con el ratón o la barra espaciadora para iniciar una acción en la aplicación. También se le puede añadir un icono personalizado y cambiar su comportamiento de botón de comando a conmutador. Un componente Button conmutador permanece presionado cuando se hace clic en él y recupera su estado original al repetirse el clic.

Un componente Button es una parte fundamental de muchos formularios y aplicaciones web. Utilice botones siempre que necesite que el usuario inicie un evento. Por ejemplo, la mayoría de los formularios contienen un botón Enviar. También puede añadir botones Anterior y Siguiente a una presentación.

## Interacción del usuario con el componente Button

Es posible activar o desactivar los botones de una aplicación. Si está desactivado, el botón no recibe la entrada del ratón ni del teclado. Si está activado, el botón se selecciona cuando el usuario hace clic en él o usa el tabulador para llegar hasta él. Cuando una instancia de Button está seleccionada, es posible utilizar las siguientes teclas para controlarla:

Tecla	Descripción
Mayús+Tabulador	Desplaza la selección al objeto anterior.
Barra espaciadora	Presiona o suelta el botón y activa el evento <code>click</code> .
Tabulador	Desplaza la selección al objeto siguiente.
Intro/Retorno	Desplaza la selección al objeto siguiente si se define el botón como Button predeterminado de FocusManager.

Para obtener más información sobre el control de la selección, consulte la interfaz IFocusManager y la clase FocusManager en la [Referencia de ActionScript 3.0 para Flash Professional](#) y en “[Trabajo con FocusManager](#)” en la página 29.

La previsualización dinámica de cada instancia de Button refleja los cambios de parámetros realizados durante la edición en el inspector de propiedades o el inspector de componentes.

**Nota:** si el icono es más grande que el botón, sobresaldrá por los bordes de éste.

Para designar un botón como el botón de comando predeterminado de una aplicación (el botón que recibe el evento click cuando un usuario presiona Intro), debe establecerse `FocusManager.defaultButton`. Por ejemplo, el código siguiente establece que el botón predeterminado sea una instancia de Button denominada `submitButton`.

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

Si se añade el componente Button a una aplicación, se puede hacer accesible a un lector de pantalla añadiendo las líneas de código ActionScript siguientes:

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

La accesibilidad de un componente sólo se activa una vez, sea cual sea su número de instancias.

## Parámetros del componente Button

Puede establecer los siguientes parámetros de edición en el inspector de propiedades (Ventana > Propiedades > Propiedades) o en el inspector de componentes (Ventana > Inspector de componentes) para cada instancia de Button: `emphasized`, `label`, `labelPlacement`, `selected` y `toggle`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Al asignar un valor a estos parámetros, se establecerá el estado inicial de la propiedad en la aplicación. Si se establece la propiedad en ActionScript, se sustituye el valor establecido en el parámetro. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase Button en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente Button

El siguiente procedimiento explica cómo añadir un componente Button a una aplicación durante la edición. En este ejemplo, Button cambia el estado de un componente ColorPicker cuando se hace clic.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente Button del panel Componentes al escenario e introduzca los valores siguientes en el inspector de propiedades:
  - Introduzca el nombre de instancia **aButton**.
  - Introduzca **Show** para el parámetro de etiqueta.
- 3 Añada un componente ColorPicker al escenario y asígnele el nombre de instancia **aCp**.
- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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;
    }
}
```

La segunda línea de código registra la función `clickHandler()` como la función de controlador de eventos para el evento `MouseEvent.CLICK`. El evento se produce cuando el usuario hace clic en `Button`; la función `clickHandler()` adopta una de las acciones siguientes según el valor de `Button`:

- `Show` muestra el componente `ColorPicker` y cambia la etiqueta de `Button` a `Disable`.
- `Disable` desactiva el componente `ColorPicker` y cambia la etiqueta de `Button` a `Enable`.
- `Enable` activa el componente `ColorPicker` y cambia la etiqueta de `Button` a `Hide`.
- `Hide` oculta el componente `ColorPicker` y cambia la etiqueta de `Button` a `Show`.

5 Seleccione `Control > Probar película` para ejecutar la aplicación.

## Creación de una aplicación con el componente `Button`

En el procedimiento siguiente se crea un componente `Button` conmutador mediante `ActionScript` y se muestra el tipo de evento en el panel `Salida` al hacer clic en `Button`. En el ejemplo se crea la instancia de `Button` invocando el constructor de la clase y se añade dicha instancia al escenario llamando al método `addChild()`.

1 Cree un nuevo documento de `Flash (ActionScript 3.0)`.

2 Arrastre el componente `Button` del panel `Componentes` al panel `Biblioteca` del documento actual.

De esta manera, se añade el componente a la biblioteca, aunque éste no se vuelve visible en la aplicación.

3 Abra el panel `Acciones`, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código siguiente para crear una instancia de `Button`:

```
import fl.controls.Button;

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

El método `move()` coloca el botón en la ubicación 50 (coordenada x), 50 (coordenada y) en el escenario.

#### 4 Añada el siguiente código ActionScript para crear un detector de eventos y una función de controlador de eventos:

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

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

#### 5 Seleccione Control > Probar película.

Cuando hace clic en el botón, Flash muestra el mensaje, "Event type: click" en el panel Salida.

## Uso del componente CheckBox

Un elemento `CheckBox` es un recuadro que puede activarse o desactivarse. Cuando se selecciona, en el cuadro aparece una marca. Se puede añadir una etiqueta de texto a un componente `CheckBox` y colocarla a la izquierda, a la derecha o debajo del componente.

Se pueden utilizar los componentes `CheckBox` para recopilar un conjunto de valores `true` o `false` que no se excluyan entre sí. Por ejemplo, una aplicación que recopile información sobre el tipo de automóvil que desea comprar un usuario podría utilizar componentes `CheckBox` para permitir la selección de características.

### Interacción del usuario con CheckBox

Es posible activar o desactivar el componente `CheckBox` de una aplicación. Si `CheckBox` está activado y el usuario hace clic en él o en su etiqueta, dicho componente se selecciona y muestra su aspecto presionado. Si el usuario desplaza el puntero fuera del área de delimitación de un componente `CheckBox` o de su etiqueta mientras presiona el botón del ratón, el componente recupera su aspecto original y conserva la selección de entrada. El estado de `CheckBox` no cambia hasta que se deja de presionar el ratón sobre el componente. Asimismo, `CheckBox` tiene dos estados desactivados, seleccionado y no seleccionado, que utilizan los parámetros `selectedDisabledSkin` y `disabledSkin` respectivamente y que no permiten la interacción del ratón ni del teclado.

Si el componente `CheckBox` está desactivado, muestra su aspecto desactivado sea cual sea la acción del usuario. Si está desactivado, `CheckBox` no recibe la entrada del ratón ni del teclado.

La instancia de `CheckBox` recibe la selección cuando el usuario hace clic sobre ella o presiona el tabulador hasta su posición. Cuando una instancia de `CheckBox` está seleccionada, se pueden utilizar las siguientes teclas para controlarla:

Tecla	Descripción
Mayús+Tabulador	Desplaza la selección al elemento anterior.
Barra espaciadora	Selecciona el componente o anula su selección y activa el evento <code>change</code> .
Tabulador	Desplaza la selección al elemento siguiente.

Para obtener más información sobre el control de la selección, consulte “Trabajo con FocusManager” en la página 29 y la clase FocusManager en la *Referencia de ActionScript 3.0 para Flash Professional*.

La previsualización dinámica de cada instancia de CheckBox refleja los cambios de parámetros realizados durante la edición en el inspector de propiedades o el inspector de componentes.

Si se añade el componente CheckBox a una aplicación, se puede hacer accesible a un lector de pantalla añadiendo las líneas de código ActionScript siguientes:

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

La accesibilidad de un componente sólo se activa una vez, sea cual sea su número de instancias.

## Parámetros del componente CheckBox

A continuación se indican los parámetros de edición que se pueden definir para cada instancia del componente CheckBox en el inspector de propiedades o en el inspector de componentes: `label`, `labelPlacement` y `selected`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase CheckBox en la *Referencia de ActionScript 3.0 para Flash Professional*.

## Creación de una aplicación utilizando CheckBox

En el siguiente procedimiento se explica cómo añadir un componente CheckBox a una aplicación durante la edición, mediante el fragmento de un formulario de una aplicación para préstamos. El formulario pregunta si el solicitante es propietario de un inmueble y proporciona un componente CheckBox para contestar "sí". En caso afirmativo, el formulario presenta dos botones de opción para que el usuario indique el valor relativo del inmueble.

### Creación de una aplicación utilizando el componente CheckBox

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente CheckBox desde el panel Componentes al escenario.
- 3 En el inspector de propiedades, siga este procedimiento:
  - Introduzca **homeCh** como nombre de instancia.
  - Introduzca **140** en el valor de anchura.
  - Introduzca "**¿Es propietario de su vivienda?**" en el parámetro de la etiqueta.
- 4 Arrastre dos componentes RadioButton desde el panel Componentes al escenario y colóquelos debajo y a la derecha del elemento CheckBox. Introduzca los valores siguientes para dichos componentes en el inspector de propiedades:
  - Introduzca **underRb** y **overRb** en los nombres de instancia.
  - Introduzca **120** en el valor de anchura de los dos componentes RadioButton.
  - Introduzca **¿Menos de 500.000 EUR?** en el parámetro de la etiqueta de `underRb`.
  - Introduzca **¿Más de 500.000 EUR?** en el parámetro de la etiqueta de `overRb`.
  - Introduzca **valueGrp** en el parámetro `groupName` de los dos componentes RadioButton.
- 5 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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;
}
```

Este código crea un controlador de eventos para un evento `CLICK` que activa los componentes `RadioButton` `underRb` y `overRb` si está seleccionado el componente `CheckBox` `homeCh`, y los desactiva si no está seleccionado `homeCh`. Para obtener más información, consulte la clase `MouseEvent` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## 6 Seleccione Control > Probar película.

En el ejemplo siguiente se duplica la aplicación anterior, pero se crean componentes `CheckBox` y `RadioButton` con `ActionScript`.

### Creación de un componente `CheckBox` con `ActionScript`

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente `CheckBox` y el componente `RadioButton` desde el panel Componentes al panel Biblioteca del documento actual. Si el panel Biblioteca no está abierto, presione `Ctrl+L` o seleccione `Ventana > Biblioteca` para abrir el panel Biblioteca.

Esta acción permite que los componentes estén disponibles para la aplicación, pero sin colocarlos en el escenario.

- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código siguiente para crear y colocar instancias del componente:

```
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?";
```

Este código utiliza los constructores `CheckBox()` y `RadioButton()` para crear componentes, y el método `addChild()` para colocarlos en el escenario. Utiliza el método `move()` para colocar los componentes en el escenario.

- 4 Añada el siguiente código ActionScript para crear un detector de eventos y una función de controlador de eventos:

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

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

Este código crea un controlador de eventos para el evento `CLICK` que activa los componentes `RadioButton` `underRb` y `overRb` si está seleccionado el componente `CheckBox` `homeCh`, y los desactiva si no está seleccionado `homeCh`. Para obtener más información, consulte la clase `MouseEvent` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

- 5 Seleccione Control > Probar película.

## Uso del componente ColorPicker

El componente `ColorPicker` permite que el usuario seleccione un color en una lista de muestras. El modo predeterminado de `ColorPicker` muestra un solo color en un botón cuadrado. Cuando un usuario hace clic en el botón, aparece la lista de colores disponibles en un panel de muestras, además de un campo de texto que muestra el valor hexadecimal de la selección de color actual.

Se pueden definir los colores que aparecen en `ColorPicker` estableciendo su propiedad `colors` con los valores de color que se quieran mostrar.

### Interacción del usuario con el componente ColorPicker

Un componente `ColorPicker` permite que el usuario seleccione un color y lo aplique a otro objeto de la aplicación. Por ejemplo, si se desea que el usuario pueda personalizar los elementos de la aplicación, como el color de fondo o el color del texto, se puede incluir un componente `ColorPicker` y aplicar el color que selecciona el usuario.

El usuario elige un color haciendo clic en su lista de muestras en el panel o introduciendo su valor hexadecimal en el campo de texto. Una vez que el usuario selecciona un color, se puede utilizar la propiedad `selectedColor` de `ColorPicker` para aplicar el color al texto, o bien otro objeto de la aplicación.

Una instancia de `ColorPicker` se selecciona si el usuario mueve el puntero sobre ella o se desplaza hasta la misma mediante el tabulador. Si está abierto un panel de muestras de `ColorPicker`, se pueden utilizar las teclas siguientes para controlarlo:

Tecla	Descripción
Inicio	Mueve la selección al primer color del panel de muestras.
Flecha arriba	Mueve la selección una fila arriba en el panel de muestras.
Flecha abajo	Mueve la selección una fila abajo en el panel de muestras.
Flecha derecha	Mueve la selección del panel de muestras un color a la derecha.
Flecha izquierda	Mueve la selección del panel de muestras un color a la izquierda.
Fin	Mueve la selección al último color del panel de muestras.

## Parámetros del componente ColorPicker

A continuación se indican los parámetros de edición que se pueden definir para cada instancia del componente ColorPicker en el inspector de propiedades o en el inspector de componentes: `selectedColor` y `showTextField`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase ColorPicker en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente ColorPicker

En el ejemplo siguiente se añade un componente ColorPicker a una aplicación durante la edición. En este ejemplo, cada vez que cambia el color en ColorPicker, la función `changeHandler()` llama a la función `drawBox()` para dibujar un nuevo cuadro con el color seleccionado en ColorPicker.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente ColorPicker del panel Componentes al centro del escenario y asígnele el nombre de instancia `aCp`.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película.
- 5 Haga clic en ColorPicker y seleccione un color para el cuadro.

## Creación de un componente ColorPicker con ActionScript

Este ejemplo utiliza el constructor `ColorPicker()` y `addChild()` para crear un componente ColorPicker en el escenario. Establece la propiedad `colors` en los valores de color rojo (0xFF0000), verde (0x00FF00) y azul (0x0000FF) a fin de especificar los colores que mostrará ColorPicker. Asimismo, crea `TextArea` y, cada vez que se seleccione un color diferente en ColorPicker, el ejemplo cambia el color del texto en el componente `TextArea` con el que debe coincidir.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente ColorPicker del panel Componentes al panel Biblioteca.
- 3 Arrastre el componente `TextArea` del panel Componentes al panel Biblioteca.

- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película.

## Uso del componente ComboBox

Un componente ComboBox permite que el usuario seleccione elementos individuales en una lista desplegable. ComboBox puede ser estático o editable. Un componente ComboBox editable permite que el usuario introduzca texto directamente en el campo de texto, situado en la parte superior de la lista. Si la lista desplegable alcanza el final del documento, se abre arriba en vez de abajo. El componente ComboBox consta de tres subcomponentes: los componentes BaseButton, TextInput y List.

En un componente ComboBox editable, sólo el botón constituye el área activa, no el cuadro de texto. En el caso de un componente ComboBox estático, el botón y el cuadro de texto constituyen el área activa. El área activa responde abriendo o cerrando la lista desplegable.

Si el usuario efectúa una selección en la lista, ya sea con el ratón o con el teclado, la etiqueta de la selección se copia al campo de texto en la parte superior del componente ComboBox.

## Interacción del usuario con el componente ComboBox

Utilice el componente ComboBox en formularios o aplicaciones en los que haya que elegir una sola opción en una lista. Por ejemplo, puede emplear una lista desplegable de provincias en un formulario de dirección de cliente. Los componentes ComboBox editables también pueden utilizarse en escenarios más complejos. Por ejemplo, en una aplicación que proporcione itinerarios por carreteras, se podría utilizar un componente ComboBox editable para que el usuario introduzca sus direcciones de origen y destino. La lista desplegable contendría las direcciones introducidas con anterioridad.

Si el componente ComboBox se puede editar, es decir, si la propiedad `editable` es `true`, las teclas siguientes anulan la selección del cuadro de entrada de texto y dejan el valor anterior. La excepción es la tecla Intro, que aplica primero el nuevo valor si el usuario ha introducido texto.

Tecla	Descripción
Mayús + Tabulador	Desplaza la selección al elemento anterior. Si se selecciona un nuevo elemento, se distribuye un evento <code>change</code> .
Tabulador	Desplaza la selección al elemento siguiente. Si se selecciona un nuevo elemento, se distribuye un evento <code>change</code> .
Flecha abajo	Desplaza la selección un elemento hacia abajo.
Fin	Desplaza la selección a la parte inferior de la lista.
Esc	Cierra la lista desplegable y vuelve a seleccionar el componente ComboBox.
Intro	Cierra la lista desplegable y vuelve a seleccionar el componente ComboBox. Si el componente ComboBox se puede editar y el usuario introduce texto, Intro establece el valor en el texto introducido.
Inicio	Desplaza la selección a la parte superior de la lista.
Re Pág	La selección retrocede una página.
Av Pág	La selección avanza una página.

Si se añade el componente ComboBox a una aplicación, se puede hacer accesible a un lector de pantalla añadiendo las líneas de código ActionScript siguientes:

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

La accesibilidad de un componente sólo se activa una vez, sea cual sea su número de instancias.

## Parámetros del componente ComboBox

A continuación se indican los parámetros que se pueden definir para cada instancia de ComboBox en el inspector de propiedades o en el inspector de componentes: `showTextField`, `editable`, `prompt` y `rowCount`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase ComboBox en la [Referencia de ActionScript 3.0 para Flash Professional](#). Para información sobre la utilización del parámetro `dataProvider`, consulte “Uso del parámetro `dataProvider`” en la página 31.

## Creación de una aplicación con el componente ComboBox

En el procedimiento siguiente se describe cómo añadir un componente ComboBox a una aplicación durante la edición. El componente ComboBox es editable y, si se escribe **Add** en el campo de texto, el ejemplo añade un elemento a la lista desplegable.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente ComboBox al escenario y asígnele el nombre de instancia **aCb**. En la ficha Parámetros, establezca el parámetro `editable` en `true`.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código siguiente:

```
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 Seleccione Control > Probar película.

## Creación de un componente ComboBox con ActionScript

En el ejemplo siguiente se crea un componente ComboBox con ActionScript, que se llena con una lista de universidades de la zona de San Francisco (California). El ejemplo establece la propiedad `width` de ComboBox para admitir la anchura del texto del mensaje de solicitud. Asimismo, define la propiedad `dropdownWidth` para que sea un poco más ancha y se ajuste al nombre de universidad de mayor longitud.

Crea la lista de universidades en una instancia de Array, mediante la propiedad `label` para almacenar los nombres de facultad y la propiedad `data` para almacenar las direcciones URL del sitio web de cada facultad. Asigna Array a ComboBox estableciendo su propiedad `dataProvider`.

Si el usuario selecciona una universidad de la lista, activa un evento `Event.CHANGE` y una llamada a la función `changeHandler()`, que carga la propiedad `data` en una petición de URL para acceder al sitio web de la facultad.

Debe tenerse en cuenta que la última línea establece la propiedad `selectedIndex` de la instancia de ComboBox en -1 para volver a mostrar el mensaje de solicitud cuando se cierre la lista. De lo contrario, el nombre de la facultad seleccionada sustituirá el mensaje.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente ComboBox del panel Componentes al panel Biblioteca.

- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película.

Se puede implementar y ejecutar este ejemplo en el entorno de edición de Flash, pero aparecerán mensajes de advertencia si se intenta acceder a los sitios web de facultad al hacer clic en elementos de ComboBox. Para poder disfrutar de la funcionalidad completa de ComboBox en Internet, acceda a la ubicación siguiente:

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

## Uso del componente DataGrid

El componente DataGrid permite mostrar datos en una cuadrícula de filas y columnas, dibujando los datos de un conjunto o un archivo XML externo que se puede analizar en un conjunto de DataProvider. El componente DataGrid incluye desplazamiento horizontal y vertical, compatibilidad con eventos (incluida la compatibilidad con celdas editables) y capacidades de ordenación.

Es posible personalizar y cambiar el tamaño de las características, como la fuente, el color y los bordes de columnas en una cuadrícula. Puede utilizar un clip de película personalizado como procesador de celdas para cualquier columna de una cuadrícula. Los procesadores de celdas muestran el contenido de una celda. Se pueden desactivar las barras de desplazamiento y utilizar los métodos de DataGrid para crear una visualización del estilo de vista de página. Para obtener más información sobre la personalización, consulte la clase DataGridColumn en la [Referencia de ActionScript 3.0 para Flash Professional](#).

### Más temas de ayuda

[Creating, populating, and resizing the DataGrid component \(Creación, relleno y cambio de tamaño del componente DataGrid\)](#)

[Customizing and sorting the DataGrid component \(Personalización y ordenación del componente DataGrid\)](#)

[Filtering and formatting data in the DataGrid component \(Filtro y formato de datos del componente DataGrid\)](#)

## Interacción del usuario con el componente DataGrid

Puede utilizar el ratón y el teclado para interactuar con un componente DataGrid.

Si la propiedad `sortableColumns` y la propiedad `sortable` de la columna tienen el valor `true`, al hacer clic en un encabezado de columna, se ordenan los datos según los valores de dicha columna. Para desactivar la ordenación en una columna determinada, hay que establecer su propiedad `sortable` en `false`.

Si la propiedad `resizableColumns` es `true`, se puede cambiar el tamaño de las columnas arrastrando los divisores de columna en la fila de encabezado.

Al hacer clic en una celda editable, se selecciona dicha celda; al hacer clic en una celda no editable no se altera el estado de la selección. Una celda es editable si sus propiedades `DataGrid.editable` y `DataGridColumn.editable` son `true`.

Para obtener más información, consulte las clases `DataGrid` y `DataGridColumn` en la [Referencia de ActionScript 3.0](#).

Si se selecciona una instancia `DataGrid` haciendo clic en la misma o utilizando la tabulación, podrá utilizar las teclas siguientes para controlar dicha instancia:

Tecla	Descripción
Flecha abajo	Cuando se edita una celda, el punto de inserción se desplaza al final del texto de la celda. Si una celda no puede editarse, la tecla de flecha abajo permite gestionar la selección tal como lo hace el componente <code>List</code> .
Flecha arriba	Cuando se edita una celda, el punto de inserción se desplaza al principio del texto de la celda. Si una celda no puede editarse, la tecla de flecha arriba permite gestionar la selección tal como lo hace el componente <code>List</code> .
Mayús+Flecha arriba/abajo	Si el componente <code>DataGrid</code> no es editable y <code>allowMultipleSelection</code> es <code>true</code> , selecciona las filas contiguas. Si se invierte la dirección con la flecha opuesta, anula la selección de las filas seleccionadas hasta que pase la fila inicial; a partir de ese momento, se seleccionan las filas en dicha dirección.
Mayús+Clic	Si <code>allowMultipleSelection</code> es <code>true</code> , selecciona todas las filas entre la fila seleccionada y la posición del símbolo actual (celda resaltada).
Ctrl+Clic	Si <code>allowMultipleSelection</code> es <code>true</code> , selecciona filas adicionales, que no necesitan ser contiguas.
Flecha derecha	Cuando se edita una celda, el punto de inserción se desplaza un carácter hacia la derecha. Si una celda no puede editarse, la tecla de flecha derecha no hará nada.
Flecha izquierda	Cuando se edita una celda, el punto de inserción se desplaza un carácter hacia la izquierda. Si una celda no puede editarse, la tecla de flecha izquierda no hará nada.

Tecla	Descripción
Inicio	Selecciona la primera fila en DataGrid.
Fin	Selecciona la última fila en DataGrid.
Re Pág	Selecciona la primera fila de una página de DataGrid. Una página consta del número de filas que DataGrid puede mostrar sin desplazamiento.
Av Pág	Selecciona la última fila de una página de DataGrid. Una página consta del número de filas que DataGrid puede mostrar sin desplazamiento.
Retorno/Intro/Mayús+Intro	Si una celda puede editarse, se lleva a cabo el cambio y el punto de inserción se desplaza a la celda de la misma columna, en la fila siguiente (arriba o abajo, en función del sentido de desplazamiento activado).
Mayús+Tabulador/Tabulador	Si DataGrid es editable, selecciona el elemento anterior/siguiente hasta que se alcance el final de la columna; a continuación, se selecciona la fila anterior/siguiente hasta que se alcance la primera o la última celda. Si se selecciona la primera celda, Mayús+Tabulador selecciona el control anterior. Si se selecciona la última celda, Tabulador selecciona el control siguiente.  Si DataGrid no es editable, selecciona el control anterior/siguiente.

Puede utilizar el componente DataGrid como base para los numerosos tipos de aplicaciones basadas en datos. Se puede mostrar fácilmente una vista de datos tabular con formato, aunque también se pueden utilizar las funciones del procesador de celdas para crear interfaces de usuario más sofisticadas y editables. A continuación se presentan usos prácticos del componente DataGrid:

- Cliente de correo web
- Páginas de resultados de búsqueda
- Aplicaciones de hojas de cálculo, como las calculadoras de préstamos y las aplicaciones de impresos de declaración de la renta.

Al diseñar una aplicación con el componente DataGrid, resulta útil comprender el diseño del componente List, ya que la clase DataGrid amplía la clase SelectableList. Para obtener más información sobre la clase SelectableList y el componente List, consulte las clases SelectableList y List en la [Referencia de ActionScript 3.0](#).

Si se añade el componente DataGrid a una aplicación, se puede hacer accesible a un lector de pantalla añadiendo las líneas de código ActionScript siguientes:

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

La accesibilidad de un componente sólo se activa una vez, sea cual sea su número de instancias. Para más información, consulte el Capítulo 18, "Creación de contenido accesible" de *Uso de Flash*.

## Parámetros del componente DataGrid

A continuación se indican los parámetros de edición que se pueden definir para cada instancia del componente DataGrid en el inspector de propiedades o en el inspector de componentes: allowMultipleSelection, editable, headerHeight, horizontalLineScrollSize, horizontalPageScrollSize, horizontalScrollPolicy, resizableColumns, rowHeight, showHeaders, verticalLineScrollSize, verticalPageScrollSize y verticalScrollPolicy. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase DataGrid en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente DataGrid

Para crear una aplicación con el componente DataGrid, primero debe determinar el origen de los datos. Normalmente, los datos proceden de Array, que puede insertarse en la cuadrícula estableciendo la propiedad `dataProvider`. También se pueden utilizar los métodos de las clases DataGrid y DataGridColumn para añadir datos a la cuadrícula.

### Uso de un proveedor de datos local con un componente DataGrid

En este ejemplo se crea un componente DataGrid para mostrar el calendario de un equipo de fútbol. El ejemplo define el calendario en Array (`aRoster`) y lo asigna a la propiedad `dataProvider` de DataGrid.

- 1 En Flash, seleccione Archivo > Nuevo y, a continuación, seleccione Archivo de Flash (ActionScript 3.0).
- 2 Arrastre el componente DataGrid del panel Componentes al escenario.
- 3 En el inspector de propiedades, introduzca el nombre de instancia **aDg**.
- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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);
};
```

La función `bldRosterGrid()` establece el tamaño de DataGrid y define el orden de las columnas y sus respectivos tamaños.

- 5 Seleccione Control > Probar película.

### Especificar columnas y añadir la ordenación para el componente DataGrid de una aplicación

Debe tenerse en cuenta que se puede hacer clic en cualquier encabezado de columna para ordenar el contenido de DataGrid en orden descendente, según los valores de dicha columna.

En el siguiente ejemplo se utiliza el método `addColumn()` para añadir instancias de `DataGridColumn` a un componente `DataGrid`. Las columnas representan nombres de jugadores y sus puntuaciones. En el ejemplo también se define la propiedad `sortOptions` para especificar las opciones de orden para cada columna: `Array.CASEINSENSITIVE` para la columna de nombres y `Array.NUMERIC` para la columna de puntuaciones. Se da el tamaño apropiado a `DataGrid` estableciendo la longitud en el número de filas y la anchura en 200.

- 1 En Flash, seleccione Archivo > Nuevo y, a continuación, seleccione Archivo de Flash (ActionScript 3.0).
- 2 Arrastre el componente `DataGrid` del panel Componentes al escenario.
- 3 En el inspector de propiedades, introduzca el nombre de instancia **aDg**.
- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película.

### Creación de una instancia del componente DataGrid mediante código ActionScript

En este ejemplo se crea un componente `DataGrid` mediante ActionScript, que se llena con una instancia de `Array` de nombres y puntuaciones de jugadores.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente `DataGrid` del panel Componentes al panel Biblioteca del documento actual.  
De este modo, se añade el componente a la biblioteca, aunque éste no se vuelve visible en la aplicación.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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);
```

Este código crea la instancia de DataGrid y, a continuación, ajusta el tamaño de la cuadrícula y la coloca en su posición correspondiente.

- 4 Cree un conjunto, añádale datos e identifíquela como el proveedor de datos de 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 Seleccione Control > Probar película.

### Cargar un componente DataGrid con un archivo XML

En el ejemplo siguiente se utiliza la clase DataGridColumn para crear las columnas de DataGrid. Se llena DataGrid pasando un objeto XML como el parámetro value del constructor DataProvider().

- 1 Con un editor de texto, cree un archivo XML con los siguientes datos y guárdelo como team.xml en la misma carpeta en la que guardará el archivo 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 Cree un nuevo documento de Flash (ActionScript 3.0).
- 3 En el panel Componentes, haga doble clic en el componente DataGrid para añadirlo al escenario.
- 4 En el inspector de propiedades, introduzca el nombre de instancia **aDg**.
- 5 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película.

## Uso del componente Label

El componente Label muestra una sola línea de texto, normalmente para identificar otros elementos o actividades en una página web. Se puede especificar que Label tenga formato HTML para sacar partido de sus etiquetas de formato de texto. También se puede controlar la alineación y el tamaño de una etiqueta. Los componentes Label no tienen bordes, no se pueden seleccionar y no difunden eventos.

La previsualización dinámica de cada instancia de Label refleja los cambios de parámetros realizados durante la edición en el inspector de propiedades o el inspector de componentes. La etiqueta no tiene bordes, por lo que el único modo de obtener una previsualización dinámica es definir su parámetro text.

### Interacción del usuario con el componente Label

Se utiliza un componente Label para crear una etiqueta de texto para otro componente en un formularios, como la etiqueta "Nombre:" a la izquierda de un campo TextInput que acepta el nombre de un usuario. Resulta conveniente utilizar un componente Label en lugar de un campo de texto normal, ya que así podrá utilizar estilos para lograr una apariencia uniforme.

Para girar un componente Label, hay que incorporar las fuentes; de lo contrario, no se mostrarán al probar la película.

## Parámetros del componente Label

A continuación se indican los parámetros de edición que se pueden definir para cada instancia del componente Label en el inspector de propiedades o en el inspector de componentes: `autoSize`, `condenseWhite`, `selectable`, `text` y `wordWrap`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase Label en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente Label

El siguiente procedimiento explica cómo añadir un componente Label a una aplicación durante la edición. En este ejemplo, la etiqueta sólo muestra el texto "Fecha de caducidad".

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente Label del panel Componentes al escenario y asígnele los valores siguientes en el inspector de propiedades:
  - Introduzca **aLabel** como nombre de instancia.
  - Introduzca **80** en el valor de anchura.
  - Introduzca **100** en el valor X.
  - Introduzca **100** en el valor Y.
  - Introduzca **Fecha de caducidad** en el parámetro `text`.
- 3 Arrastre un componente TextArea al escenario y asígnele los valores siguientes en el inspector de propiedades:
  - Introduzca **aTa** como nombre de instancia.
  - Introduzca **22** en el valor de altura.
  - Introduzca **200** en el valor X.
  - Introduzca **100** en el valor Y.
- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película.

### Creación de una instancia del componente Label mediante código ActionScript

En el ejemplo siguiente se crea un parámetro Label con ActionScript. Se utiliza un componente Label para identificar la función de un componente ColorPicker y se utiliza la propiedad `htmlText` para aplicar el formato al texto de Label.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente Label del panel Componentes al panel Biblioteca del documento actual.
- 3 Arrastre el componente ColorPicker del panel Componentes al panel Biblioteca del documento actual.
- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película.

## Uso del componente List

El componente List es un cuadro de lista desplazable de selección única o múltiple. Una lista también puede mostrar gráficos, incluidos otros componentes. Los elementos que se visualizan en la lista se añaden con el cuadro de diálogo Valores que aparece al hacer clic en los campos de parámetros data o labels. Para añadir elementos a la lista también pueden utilizarse los métodos `List.addItem()` y `List.addItemAt()`.

El componente List utiliza un índice basado en cero, en el que el elemento con el índice 0 aparece en primer lugar. Si añade, elimina o reemplaza elementos de lista mediante los métodos y propiedades de la clase List, es posible que tenga que especificar el índice del elemento de lista.

### Interacción del usuario con el componente List

Se puede configurar una lista para que los usuarios efectúen selecciones únicas o múltiples. Por ejemplo, un usuario visita un sitio web de comercio electrónico y debe elegir el artículo que va a comprar. Hay 30 artículos en una lista que el usuario recorre y hace clic en el que desea seleccionar.

También se puede diseñar un componente List que utilice clips de película personalizados como filas para poder mostrar más información al usuario. Por ejemplo, en una aplicación de correo electrónico, cada buzón podría ser un componente List y cada fila podría tener iconos para indicar la prioridad y el estado.

El componente List se selecciona al hacer clic o desplazarse a él mediante el tabulador; a continuación, se pueden utilizar las teclas siguientes para controlarlo:

Tecla	Descripción
Teclas alfanuméricas	Salta al siguiente elemento que tiene <code>Key.getAscii()</code> como el primer carácter de su etiqueta.
Control	Tecla conmutadora que permite seleccionar y anular la selección de varios elementos no contiguos.
Flecha abajo	La selección desciende un elemento.
Inicio	La selección se desplaza al principio de la lista.
Av Pág	La selección avanza una página.
Re Pág	La selección retrocede una página.
Mayús	Permite seleccionar elementos contiguos.
Flecha arriba	La selección asciende un elemento.

**Nota:** debe tenerse en cuenta que los tamaños de desplazamiento se expresan en píxeles y no en filas.

**Nota:** para las teclas Av Pág y Re Pág, una página está formada por los elementos que caben en pantalla menos uno. Por ejemplo, para avanzar por una lista desplegable que presenta las líneas de diez en diez, la pantalla muestra los elementos 0-9, 9-18, 18-27 y así sucesivamente, solapando un elemento por página.

Para obtener más información sobre el control de la selección, consulte la interfaz IFocusManager y la clase FocusManager en la [Referencia de ActionScript 3.0 para Flash Professional](#), y en “Trabajo con FocusManager” en la página 29.

La previsualización dinámica de cada instancia de List del escenario refleja los cambios de parámetros realizados durante la edición en el inspector de propiedades o en el inspector de componentes.

Si se añade el componente List a una aplicación, se puede hacer accesible a un lector de pantalla añadiendo las líneas de código ActionScript siguientes:

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

La accesibilidad de un componente sólo se activa una vez, sea cual sea su número de instancias. Para más información, consulte el Capítulo 18, "Creación de contenido accesible" de *Uso de Flash*.

## Parámetros del componente List

A continuación se indican los parámetros de edición que se pueden definir para cada instancia del componente List en el inspector de propiedades o en el inspector de componentes: `allowMultipleSelection`, `dataProvider`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `multipleSelection`, `verticalLineScrollSize`, `verticalPageScrollSize` y `verticalScrollPolicy`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase List en la [Referencia de ActionScript 3.0 para Flash Professional](#). Para información sobre la utilización del parámetro `dataProvider`, consulte “Uso del parámetro `dataProvider`” en la página 31.

## Creación de una aplicación con el componente List

Los siguientes ejemplos explican cómo añadir un componente List a una aplicación durante la edición.

### Añadir un componente List a una aplicación

En este ejemplo, el componente List consta de etiquetas que identifican modelos de automóvil y campos de datos que contienen precios.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente List desde el panel Componentes al escenario.
- 3 En el inspector de propiedades, siga este procedimiento:
  - Introduzca el nombre de instancia **aList**.
  - Asigne un valor de **200** a la anchura.
- 4 Utilice la herramienta Texto para crear un campo de texto debajo de **aList** y asígnele el nombre de instancia **aTf**.
- 5 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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;
}
```

Este código utiliza el método `addItem()` para llenar `aList` con tres elementos; asigna un valor `label` (que aparece en la lista) a cada elemento, así como un valor `data`. Al seleccionar un elemento en el componente List, el detector de eventos llama a la función `showData()`, que muestra el valor `data` para el elemento seleccionado.

- 6 Seleccione Control > Probar película para compilar y ejecutar esta aplicación.

### Llenar una instancia de List con un proveedor de datos

En este ejemplo se crea una lista de modelos de automóvil y sus precios. Para llenar el componente List, se utiliza un proveedor de datos en lugar del método `addItem()`.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente List desde el panel Componentes al escenario.
- 3 En el inspector de propiedades, siga este procedimiento:
  - Introduzca el nombre de instancia **aList**.
  - Asigne un valor de **200** a la anchura.
- 4 Utilice la herramienta Texto para crear un campo de texto debajo de **aList** y asígnele el nombre de instancia **aTf**.
- 5 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película para ver el componente List con sus elementos.

### Uso de un componente List para controlar una instancia de MovieClip

En el ejemplo siguiente se crea una componente List de nombres de colores y, al seleccionar uno, se aplica el color a un componente MovieClip.

- 1 Cree un documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente List del panel Componentes al escenario y asígnele los valores siguientes en el inspector de propiedades:
  - Introduzca **aList** como nombre de instancia.
  - Introduzca **60** en el valor de altura.
  - Introduzca **100** en el valor X.
  - Introduzca **150** en el valor Y.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película para ejecutar la aplicación.
- 5 Haga clic en los colores de List para ver cómo se muestran en MovieClip.

#### Creación de una instancia del componente List con ActionScript:

En este ejemplo se crea una lista sencilla utilizando ActionScript y se llena con el método `addItem()`.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente List del panel Componentes al panel Biblioteca.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película para ejecutar la aplicación.

## Uso del componente NumericStepper

El componente NumericStepper permite que el usuario se desplace por un conjunto ordenado de números. El componente consta de un número en un cuadro de texto que se muestra junto a pequeños botones de flecha arriba y flecha abajo. Cuando el usuario presiona los botones, el número aumenta o disminuye de forma gradual según la unidad especificada en el parámetro `stepSize`, hasta que el usuario suelta los botones o hasta que se alcanza el valor máximo o mínimo. El texto del cuadro de texto del componente NumericStepper también se puede editar.

La previsualización dinámica de cada instancia de NumericStepper refleja la configuración del parámetro `value` en el inspector de propiedades o el inspector de componentes. Sin embargo, en la previsualización dinámica ni el teclado ni el ratón pueden interactuar con los botones de flecha de NumericStepper.

### Interacción del usuario con el componente NumericStepper

Se puede utilizar el componente NumericStepper siempre que se desee que el usuario seleccione un valor numérico. Por ejemplo, se podría utilizar un componente NumericStepper en un formulario para definir el mes, el día y el año de la fecha de caducidad de una tarjeta de crédito. Asimismo, puede utilizar un componente NumericStepper para que el usuario aumente o disminuya el tamaño de la fuente.

El componente NumericStepper sólo gestiona datos numéricos. Además, para ver más de dos lugares numéricos (por ejemplo, los números 5246 o 1,34), es preciso cambiar el tamaño del componente durante la edición.

Es posible activar o desactivar el componente NumericStepper de una aplicación. Si está desactivado, NumericStepper no recibe la entrada del ratón ni del teclado. Si está activado, NumericStepper se selecciona al hacer clic en él o desplazarse al mismo mediante el tabulador, y su selección interna se establece en el cuadro de texto. Cuando una instancia de NumericStepper está seleccionada, se pueden utilizar las teclas siguientes para controlarla:

Tecla	Descripción
Flecha abajo	El valor cambia en una unidad.
Flecha izquierda	Desplaza el punto de inserción a la izquierda dentro del cuadro de texto.
Flecha derecha	Desplaza el punto de inserción a la derecha dentro del cuadro de texto.
Mayús+Tabulador	Desplaza la selección al objeto anterior.
Tabulador	Desplaza la selección al objeto siguiente.
Flecha arriba	El valor cambia en una unidad.

Para obtener más información sobre el control de la selección, consulte la clase FocusManager en la [Referencia de ActionScript 3.0 para Flash Professional](#) y en “Trabajo con FocusManager” en la página 29.

### Parámetros del componente NumericStepper

A continuación se indican los parámetros que se pueden definir para cada instancia de NumericStepper en el inspector de propiedades o en el inspector de componentes: `maximum`, `minimum`, `stepSize` y `value`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase NumericStepper en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente NumericStepper

El siguiente procedimiento explica cómo añadir un componente NumericStepper a una aplicación durante la edición. En el ejemplo se colocan un componente NumericStepper y un componente Label en el escenario y se crea un detector de eventos para un evento `Event.CHANGE` en la instancia de NumericStepper. Si el valor del componente NumericStepper cambia, el ejemplo muestra el nuevo valor en la propiedad `text` de la instancia de Label.

- 1 Arrastre un componente NumericStepper desde el panel Componentes al escenario.
- 2 En el inspector de propiedades, introduzca el nombre de instancia **aNs**.
- 3 Arrastre un componente Label desde el panel Componentes al escenario.
- 4 En el inspector de propiedades, introduzca el nombre de instancia **aLabel**.
- 5 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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;
};
```

En este ejemplo, se establece la propiedad `text` de la etiqueta en el valor de NumericStepper. La función `changeHandler()` actualiza la propiedad `text` de la etiqueta cuando cambia el valor de la instancia de NumericStepper.

- 6 Seleccione Control > Probar película.

### Creación de un componente NumericStepper con ActionScript:

En este ejemplo se crean tres componentes NumericStepper con código ActionScript, para introducir el mes, el día y el año de la fecha de nacimiento del usuario. También se añaden componentes Label para un mensaje de solicitud y para identificadores de cada uno de los componentes NumericStepper.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente Label al panel Biblioteca.
- 3 Arrastre un componente NumericStepper al panel Biblioteca.
- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película para ejecutar la aplicación.

## Uso del componente ProgressBar

El componente ProgressBar muestra el progreso de carga de contenido, que resulta tranquilizador para el usuario cuando el contenido es de gran tamaño y puede retrasar la ejecución de la aplicación. El componente ProgressBar es útil para mostrar el progreso de las imágenes y partes que se cargan de una aplicación. El proceso de carga puede ser determinado o indeterminado. Una barra de progreso *determinada* es la representación lineal del progreso de una tarea en el tiempo y se utiliza cuando la cantidad de contenido que va a cargarse es conocida. Una barra de progreso *indeterminada* se utiliza cuando la cantidad de contenido que va a cargarse es desconocida. Asimismo, se puede añadir un componente Label para mostrar el progreso de carga como porcentaje.

El componente ProgressBar utiliza la escala de 9 divisiones y tiene un aspecto de barra, un aspecto de seguimiento y un aspecto indeterminado.

### Interacción del usuario con el componente ProgressBar

Existen tres modos en los que se puede utilizar el componente ProgressBar. Los valores más frecuentes de este parámetro son event y polled. Estos modos especifican un proceso de carga que emite los eventos `progress` y `complete` (modos event y polled), o bien, expone las propiedades `bytesLoaded` y `bytesTotal` (modo polled). También se puede utilizar el componente ProgressBar en modo manual estableciendo las propiedades `maximum`, `minimum` y `value` junto con las llamadas al método `ProgressBar.setProgress()`. Se puede establecer la propiedad `indeterminated` para indicar si ProgressBar tiene un relleno discontinuo y un origen de tamaño desconocido (`true`) o un relleno continuo y un origen de tamaño conocido (`false`).

El modo de ProgressBar se define estableciendo su propiedad `mode`, mediante el parámetro `mode` del inspector de propiedades o del inspector de componentes, o bien, mediante `ActionScript`.

Si se utiliza ProgressBar para mostrar el estado de procesamiento, como el análisis de 100.000 elementos, y se encuentra en un solo bucle de fotograma, no se podrán ver actualizaciones para ProgressBar porque la pantalla no se actualiza.

### Parámetros del componente ProgressBar

A continuación se indican los parámetros de edición que se pueden definir para cada instancia del componente ProgressBar en el inspector de propiedades o en el inspector de componentes: `direction`, `mode` y `source`. Cada uno de ellos tiene una propiedad de `ActionScript` del mismo nombre.

Puede escribir código `ActionScript` para controlar éstas y otras opciones del componente ProgressBar utilizando sus propiedades, métodos y eventos. Para obtener más información, consulte la clase ProgressBar en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente ProgressBar

En el procedimiento siguiente se muestra cómo añadir un componente ProgressBar a una aplicación durante la edición. En este ejemplo, ProgressBar utiliza el modo event. En el modo event, el contenido de carga emite los eventos `progress` y `complete` que ProgressBar distribuye para indicar el progreso. Cuando se produce el evento `progress`, el ejemplo actualiza una etiqueta para mostrar el porcentaje del contenido que se ha cargado. Cuando se produce el evento `complete`, el ejemplo muestra "Carga completa" y el valor de la propiedad `bytesTotal`, que es el tamaño del archivo.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente ProgressBar del panel Componentes al escenario.
  - En el inspector de propiedades, introduzca el nombre de instancia **aPb**.
  - En la sección Parámetros, introduzca **200** en el valor X.
  - Introduzca **260** en el valor Y.
  - Seleccione `event` para el parámetro `mode`.
- 3 Arrastre el componente Button desde el panel Componentes al escenario.
  - En el inspector de propiedades, introduzca **loadButton** como el nombre de instancia.
  - Introduzca **220** en el parámetro X.
  - Introduzca **290** en el parámetro Y.
  - Introduzca **Load Sound** para el parámetro `label`.
- 4 Arrastre un componente Label al escenario y asígnele el nombre de instancia **progLabel**.
  - Introduzca **150** en el valor de anchura.
  - Introduzca **200** en el parámetro X.
  - Introduzca **230** en el parámetro Y.
  - En la sección Parámetros, borre el valor del parámetro `text`.
- 5 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente, que carga un archivo de audio 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 Seleccione Control > Probar película.

### Creación de una aplicación con el componente ProgressBar en modo polled

En el ejemplo siguiente, se establece ProgressBar en el modo polled. En el modo polled, el progreso se determina detectando eventos `progress` en el contenido que se carga y utilizando sus propiedades `bytesLoaded` y `bytesTotal` para calcular el progreso. En este ejemplo se carga un objeto `Sound`, se detectan sus eventos `progress` y se calcula el porcentaje cargado mediante sus propiedades `bytesLoaded` y `bytesTotal`. Se muestra el porcentaje cargado tanto en una etiqueta como en el panel Salida.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente ProgressBar del panel Componentes al escenario e introduzca los valores siguientes en el inspector de propiedades:
  - Introduzca **aPb** como nombre de instancia.
  - Introduzca **185** en el valor X.
  - Introduzca **225** en el valor Y.
- 3 Arrastre el componente Label al escenario e introduzca los valores siguientes en el inspector de propiedades:
  - Introduzca **progLabel** como nombre de instancia.
  - Introduzca **180** en el valor X.

- Introduzca **180** en el valor Y.
- En la sección Parámetros, borre el valor del parámetro de texto.

- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente, que crea un objeto Sound (`aSound`) y llama a `loadSound()` para cargar un sonido en el 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 Seleccione Control > Probar película para ejecutar la aplicación.

### Creación de una aplicación con el componente **ProgressBar** en modo manual

En el ejemplo siguiente, se establece **ProgressBar** en el modo manual. En el modo manual, se debe establecer el progreso manualmente llamando al método `setProgress()` y llenándolo con los valores actuales y máximos a fin de determinar el alcance del progreso. Pero no se necesita definir la propiedad `source` en modo manual. El ejemplo utiliza un componente **NumericStepper**, con un valor máximo de 250, para incrementar **ProgressBar**. Cuando el valor de **NumericStepper** cambia y activa un evento `CHANGE`, el controlador de eventos (`nsChangeHandler`) llama al método `setProgress()` para avanzar el componente **ProgressBar**. También muestra el porcentaje del progreso completado, según el valor máximo.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente **ProgressBar** del panel Componentes al escenario y asígnele los valores siguientes en el inspector de propiedades:
  - Introduzca **aPb** como nombre de instancia.
  - Introduzca **180** en el valor X.
  - Introduzca **175** en el valor Y.
- 3 Arrastre un componente **NumericStepper** al escenario e introduzca los valores siguientes en el inspector de propiedades:
  - Introduzca **aNs** como nombre de instancia.
  - Introduzca **220** en el valor X.
  - Introduzca **215** en el valor Y.

- En la sección Parámetros, introduzca **250** para el parámetro máximo, **0** para el valor mínimo, **1** para el parámetro `stepSize` y **0** para el parámetro `value`.
- 4 Arrastre un componente Label al escenario e introduzca los valores siguientes en el inspector de propiedades:
- Introduzca **progLabel** como nombre de instancia.
  - Introduzca **150** en el valor de anchura.
  - Introduzca **180** en el valor X.
  - Introduzca **120** en el valor Y.
  - En la ficha Parámetros, borre el componente Label como valor del parámetro de texto.
- 5 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código siguiente:
- ```
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 Seleccione Control > Probar película para ejecutar la aplicación.
- 7 Haga clic en la flecha arriba de NumericStepper para avanzar ProgressBar.

### Creación de un componente ProgressBar con ActionScript

En este ejemplo se crea un componente ProgressBar utilizando ActionScript. Al margen de ello, se duplica la funcionalidad del ejemplo anterior, que crea un componente ProgressBar en modo manual.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente ProgressBar al panel Biblioteca.
- 3 Arrastre el componente NumericStepper al panel Biblioteca.
- 4 Arrastre el componente Label al panel Biblioteca.
- 5 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código siguiente:

```
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 Seleccione Control > Probar película para ejecutar la aplicación.
- 7 Haga clic en la flecha arriba de NumericStepper para avanzar ProgressBar.

## Uso del componente RadioButton

El componente RadioButton permite obligar al usuario a seleccionar una opción de un conjunto de opciones. Este componente debe utilizarse en un grupo formado al menos por dos instancias de RadioButton. Sólo es posible seleccionar un miembro del grupo cada vez. Al seleccionar un botón de opción de un grupo, se anula la selección del botón de opción seleccionado en dicho grupo. Puede definir el parámetro `groupName` para indicar el grupo al que pertenece el botón de opción.

Un botón de opción es una parte fundamental de muchos formularios y aplicaciones en Internet. Puede utilizar los botones de opción siempre que necesite que un usuario seleccione un elemento de un grupo de opciones. Por ejemplo, podría emplear botones de opción en un formulario para preguntar qué tarjeta de crédito desea utilizar el cliente.

## Interacción del usuario con el componente RadioButton

Los botones de opción pueden estar activados o desactivados. Un botón de opción en estado desactivado no recibe la entrada del ratón ni del teclado. Si el usuario hace clic en un grupo de componentes RadioButton o presiona el tabulador hasta su posición, la selección recae sólo en el botón de opción elegido. A continuación, el usuario puede utilizar las teclas siguientes para controlarlo.

| Tecla                          | Descripción                                                                              |
|--------------------------------|------------------------------------------------------------------------------------------|
| Flecha arriba/flecha izquierda | La selección se desplaza al botón anterior del grupo de botones de opción.               |
| Flecha abajo/flecha derecha    | La selección se desplaza al botón siguiente del grupo de botones de opción.              |
| Tabulador                      | Desplaza la selección desde el grupo de botones de opción hasta el componente siguiente. |

Para obtener más información sobre el control de la selección, consulte la interfaz IFocusManager y la clase FocusManager en la [Referencia de ActionScript 3.0 para Flash Professional](#) y en “[Trabajo con FocusManager](#)” en la página 29.

La previsualización dinámica de cada instancia de RadioButton del escenario refleja los cambios de parámetros realizados durante la edición en el inspector de propiedades o en el inspector de componentes. Sin embargo, la previsualización dinámica no muestra la exclusión mutua de la selección. Si se define el parámetro seleccionado en `true` para dos botones de opción del mismo grupo, ambos botones aparecerán seleccionados, si bien sólo la última instancia creada estará seleccionada en tiempo de ejecución. Para más información, consulte “[Parámetros del componente RadioButton](#)” en la página 80.

Si se añade el componente RadioButton a una aplicación, se puede hacer accesible a un lector de pantalla añadiendo las líneas de código siguientes:

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

La accesibilidad de un componente sólo se activa una vez, sea cual sea su número de instancias. Para más información, consulte el Capítulo 18, "Creación de contenido accesible" de Uso de Flash.

## Parámetros del componente RadioButton

A continuación se indican los parámetros de edición que se pueden definir para cada instancia del componente RadioButton en el inspector de propiedades o en el inspector de componentes: `groupName`, `label`, `labelPlacement`, `selected` y `value`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase RadioButton en la [Referencia de ActionScript 3.0 para Flash Professional](#).

Puede escribir código ActionScript para definir otras opciones para las instancias de RadioButton con los métodos, propiedades y eventos de la clase RadioButton.

## Creación de una aplicación con el componente RadioButton

El procedimiento siguiente explica cómo añadir componentes RadioButton a una aplicación durante la edición. En este ejemplo, los componentes RadioButton se utilizan para presentar una pregunta que requiera una respuesta de tipo "sí" o "no". Los datos de RadioButton se muestran en un componente TextArea.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre dos componentes RadioButton desde el panel Componentes al escenario.

- 3 Seleccione el primer botón de opción. En el inspector de propiedades, asígnele el nombre de instancia **yesRb** y el nombre de grupo **rbGroup**.
- 4 Seleccione el segundo botón de opción. En el inspector de propiedades, asígnele el nombre de instancia **noRb** y el nombre de grupo **rbGroup**.
- 5 Arrastre un componente TextArea del panel Componentes al escenario y asígnele el nombre de instancia **aTa**.
- 6 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película para ejecutar la aplicación.

### Creación de un componente RadioButton con ActionScript

En este ejemplo se utiliza ActionScript para crear tres componentes RadioButton para los colores rojo, azul y verde, y se dibuja un cuadro gris. La propiedad `value` de cada RadioButton especifica el valor hexadecimal del color asociado al botón. Cuando el usuario hace clic en uno de los componentes RadioButton, la función `clickHandler()` llama a `drawBox()` y pasa el color de la propiedad `value` de RadioButton para aplicar color al cuadro.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente RadioButton al panel Biblioteca.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película para ejecutar la aplicación.

Para obtener más información, consulte la clase `RadioButton` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Uso del componente ScrollPane

Puede utilizar el componente `ScrollPane` para visualizar contenido demasiado grande para el área en que se esté cargando. Por ejemplo, si se tiene una imagen de gran tamaño pero poco espacio para ella en la aplicación, se podría cargar en un componente `ScrollPane`. `ScrollPane` puede aceptar clips de película, así como archivos JPEG, PNG, GIF y SWF.

Los componentes como ScrollPane y UILoader tienen eventos `complete` que permiten determinar el momento en que termina la carga de contenido. Para definir propiedades en el contenido de un componente ScrollPane o UILoader, es necesario detectar el evento `complete` y establecer la propiedad en el controlador de eventos. Por ejemplo, el código siguiente crea un detector para el evento `Event.COMPLETE` y un controlador de eventos que establece la propiedad `alpha` del contenido de ScrollPane en `.5`:

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

Si se especifica una ubicación al cargar contenido en ScrollPane, se debe especificar la ubicación (coordenadas X e Y) como 0,0. Por ejemplo, el código siguiente carga ScrollPane correctamente porque el cuadro se dibuja en la ubicación 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 obtener más información, consulte la clase ScrollPane en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Interacción del usuario con el componente ScrollPane

Un componente ScrollPane puede estar activado o desactivado. Si está desactivado, no recibe la entrada del ratón ni del teclado. El usuario puede utilizar las teclas siguientes para controlar un componente ScrollPane cuando esté seleccionado:

| Tecla            | Descripción                                                           |
|------------------|-----------------------------------------------------------------------|
| Flecha abajo     | Mueve el contenido una línea de desplazamiento vertical hacia arriba. |
| Flecha arriba    | Mueve el contenido una línea de desplazamiento vertical hacia abajo.  |
| Fin              | Mueve el contenido hasta la parte inferior del componente ScrollPane. |
| Flecha izquierda | Desplaza el contenido horizontalmente una línea hacia la derecha.     |
| Flecha derecha   | Desplaza el contenido horizontalmente una línea hacia la izquierda.   |
| Inicio           | Mueve el contenido hasta la parte superior del componente ScrollPane. |
| Fin              | Mueve el contenido hasta la parte inferior del componente ScrollPane. |
| Av Pág           | Desplaza el contenido verticalmente una página hacia arriba.          |
| Re Pág           | Desplaza el contenido verticalmente una página hacia abajo.           |

El usuario puede utilizar el ratón para interactuar con ScrollPane, tanto en su contenido como en las barras de desplazamiento verticales y horizontales. El usuario puede arrastrar el contenido mediante el ratón si la propiedad `scrollDrag` está establecida en `true`. La aparición de un puntero con forma de mano en el contenido indica que el usuario puede arrastrarlo. A diferencia de la mayoría de controles, se producen acciones desde que se presiona el botón del ratón hasta que se suelta. Si el contenido dispone de tabulaciones válidas, debe establecerse `scrollDrag` en `false`. De lo contrario, todas las acciones del ratón sobre el contenido invocarán el arrastre de desplazamiento.

## Parámetros del componente ScrollPane

Puede establecer los siguientes parámetros para cada instancia de ScrollPane en el inspector de propiedades o en el inspector de componentes: `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `scrollDrag`, `source`, `verticalLineScrollSize`, `verticalPageScrollSize` y `verticalScrollPolicy`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase ScrollPane en la [Referencia de ActionScript 3.0 para Flash Professional](#).

Puede escribir código ActionScript para controlar éstas y otras opciones adicionales para los componentes ScrollPane utilizando sus propiedades, métodos y eventos.

## Creación de una aplicación con el componente ScrollPane

En el procedimiento siguiente se explica cómo añadir un componente ScrollPane a una aplicación durante la edición. En este ejemplo, ScrollPane carga una imagen de una ruta de acceso especificada por la propiedad `source`.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente ScrollPane del panel Componentes al escenario y asígnele el nombre de instancia **aSp**.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película para ejecutar la aplicación.

## Creación de una instancia del componente ScrollPane con ActionScript

En el ejemplo se crea un componente ScrollPane, se establece su tamaño y se carga una imagen de dicho componente con la propiedad `source`. También se crean dos detectores. El primero detecta un evento `scroll` y muestra la posición de la imagen a medida que el usuario se desplaza vertical o horizontalmente. El segundo detecta un evento `complete` y muestra un mensaje en el panel Salida que indica que la imagen ha terminado de cargarse.

Este ejemplo crea un componente ScrollPane mediante ActionScript y coloca un componente MovieClip (cuadro rojo) en él, cuyo tamaño es de 150 píxeles (ancho) por 300 píxeles (alto).

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente ScrollPane del panel Componentes al panel Biblioteca.
- 3 Arrastre el componente DataGrid del panel Componentes al panel Biblioteca.

- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película para ejecutar la aplicación.

## Uso del componente Slider

El componente Slider permite que el usuario seleccione un valor deslizando un *pulgar* gráfico entre los extremos de una pista que corresponde a un rango de valores. Se puede utilizar un deslizador para permitir que el usuario seleccione un valor como, por ejemplo, un número o un porcentaje. También se puede utilizar código ActionScript para que el valor del deslizador influya en el comportamiento de un segundo objeto. Por ejemplo, se podría asociar el deslizador con una imagen y reducirla o ampliarla según la posición relativa, o el valor, del deslizador.

El valor actual del componente Slider se determina mediante la ubicación relativa del deslizador entre los extremos de la pista o los valores mínimos o máximos de dicho componente.

Slider admite un rango continuo de valores entre sus valores mínimos y máximos, pero también se puede establecer el parámetro `snapInterval` para especificar intervalos entre el valor mínimo y máximo. Un componente Slider puede mostrar marcas de selección independientes de los valores asignados del deslizador, a intervalos especificados a lo largo de la pista.

De forma predeterminada, el deslizador tiene una orientación horizontal, pero se le puede dar una orientación vertical estableciendo el valor del parámetro `direction` en vertical. La pista del deslizador se expande de extremo a extremo y las marcas de selección se colocan de izquierda a derecha, encima de la pista.

## Interacción del usuario con Slider

Cuando una instancia de Slider está seleccionada, se pueden utilizar las teclas siguientes para controlarla:

| Tecla            | Descripción                                              |
|------------------|----------------------------------------------------------|
| Flecha derecha   | Aumenta el valor asociado de un deslizador horizontal.   |
| Flecha arriba    | Aumenta el valor asociado de un deslizador vertical.     |
| Flecha izquierda | Disminuye el valor asociado de un deslizador horizontal. |
| Flecha abajo     | Disminuye el valor asociado de un deslizador vertical.   |
| Mayús+Tabulador  | Desplaza la selección al objeto anterior.                |
| Tabulador        | Desplaza la selección al objeto siguiente.               |

Para obtener más información sobre el control de la selección, consulte la interfaz IFocusManager y la clase FocusManager en la [Referencia de ActionScript 3.0 para Flash Professional](#) y en “Trabajo con FocusManager” en la página 29.

La previsualización dinámica de cada instancia de Slider refleja los cambios de parámetros realizados durante la edición en el inspector de propiedades o el inspector de componentes.

## Parámetros del componente Slider

Puede establecer los siguientes parámetros de edición para cada instancia del componente Slider en el inspector de propiedades o en el inspector de componentes: `direction`, `liveDragging`, `maximum`, `minimum`, `snapInterval`, `tickInterval` y `value`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase Slider en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente Slider

En el ejemplo siguiente se crea una instancia de Slider para permitir que el usuario exprese su nivel de satisfacción con algún evento hipotético. El usuario mueve el componente Slider a derecha o izquierda para indicar un mayor o menor nivel de satisfacción.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente Label del panel Componentes al centro del escenario.
  - Asigne el nombre de instancia **valueLabel**.
  - Asigne el valor **0por ciento** al parámetro `text`.
- 3 Arrastre un componente Slider del panel Componentes y céntrelo debajo de `value_label`.
  - Asigne el nombre de instancia **aSlider**.
  - Asigne una anchura de **200**
  - Asigne una altura de **10**
  - Asigne un valor de **100** al parámetro `maximum`.
  - Asigne un valor de **10** a los parámetros `snapInterval` y `tickInterval`.
- 4 Arrastre otra instancia de Label del panel Biblioteca y céntrela debajo de `aSlider`.
  - Asigne el nombre de instancia **promptLabel**.
  - Asigne una anchura de 250.
  - Asigne una altura de 22.

- Introduzca **Indique su nivel de satisfacción** en el parámetro `text`.

5 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película.

En este ejemplo, a medida que mueve el deslizador de un intervalo a otro, un detector del evento `SliderEvent.CHANGE` actualiza la propiedad `text` de `valueLabel` para mostrar el porcentaje correspondiente a la posición del deslizador.

## Creación de una aplicación con el componente Slider mediante ActionScript

En el ejemplo siguiente se crea un componente Slider con ActionScript. Se descarga la imagen de una flor y se utiliza el componente Slider para permitir que el usuario pueda aplicar más o menos intensidad de luz a la imagen cambiando su propiedad `alpha` para crear una correspondencia con el valor de Slider.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre los componentes Label y Slider del panel Componentes al panel Biblioteca del documento actual.  
Esta acción añade los componentes a la biblioteca, pero no los muestra en la aplicación.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código siguiente para crear y colocar instancias del componente:

```
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 Seleccione Control > Probar película para ejecutar la aplicación.
- 5 Mueva el deslizador de Slider a la izquierda para disminuir la intensidad de luz de la imagen y a la derecha para aumentarla.

## Uso del componente TextArea

El componente TextArea es un contenedor del objeto nativo TextField de ActionScript. Se puede utilizar el componente TextArea para mostrar texto, así como para editar y recibir entradas de texto si la propiedad `editable` es `true`. El componente puede mostrar o recibir varias líneas de texto y contener líneas largas de texto si la propiedad `wordWrap` se establece en `true`. La propiedad `restrict` permite restringir los caracteres que el usuario puede introducir. `maxChars` permite especificar el número máximo de caracteres que puede introducir el usuario. Si el texto sobrepasa los límites horizontales o verticales del área de texto, aparecen automáticamente barras de desplazamiento horizontales y verticales, a menos que sus propiedades asociadas, `horizontalScrollPolicy` y `verticalScrollPolicy`, se establezcan en `off`.

Puede utilizar un componente `TextArea` siempre que necesite un campo de texto de varias líneas. Por ejemplo, suponga que precisa un componente `TextArea` como campo de comentarios de un formulario. En tal caso, se puede definir un detector que compruebe si el campo está vacío cuando un usuario emplea el tabulador fuera del campo. El detector podría mostrar un mensaje de error indicando que debe introducirse un comentario en el campo.

Si necesita un campo de texto de una única línea, utilice el componente `TextInput`.

Se puede establecer el estilo `textFormat` con el método `setStyle()` para cambiar el estilo de texto que aparece en una instancia de `TextArea`. Asimismo, se puede aplicar el formato HTML a un componente `TextArea` mediante la propiedad `htmlText` de `ActionScript` y se puede establecer la propiedad `displayAsPassword` en `true` para enmascarar texto con asteriscos. Si se establece la propiedad `condenseWhite` en `true`, Flash elimina los espacios en blanco adicionales del nuevo texto (originados por espacios, saltos de línea, etc.). Esto no tiene ningún efecto sobre el texto que ya se encuentra en el control.

## Interacción del usuario con el componente `TextArea`

Es posible activar o desactivar el componente `TextArea` en una aplicación. En el estado desactivado, no puede recibir entradas del ratón ni del teclado. Cuando está activado, sigue las mismas reglas de selección y navegación que un objeto `TextField` de `ActionScript`. Cuando una instancia de `TextArea` está seleccionada, se puede controlar mediante las teclas siguientes:

| Tecla            | Descripción                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Teclas de flecha | Mueven el punto de inserción arriba, abajo, a la izquierda o a la derecha dentro del texto, siempre y cuando éste sea editable. |
| Av Pág           | Mueve el punto de inserción al final del texto, si éste es editable.                                                            |
| Re Pág           | Mueve el punto de inserción al principio del texto, si éste es editable.                                                        |
| Mayús+Tabulador  | Desplaza la selección al objeto anterior en el bucle de tabulación.                                                             |
| Tabulador        | Desplaza la selección al objeto siguiente en el bucle de tabulación.                                                            |

Para obtener más información sobre el control de la selección, consulte la clase `FocusManager` en la [Referencia de ActionScript 3.0 para Flash Professional](#) y en “Trabajo con `FocusManager`” en la página 29.

## Parámetros del componente `TextArea`

Puede establecer los siguientes parámetros de edición para cada instancia del componente `TextArea` en el inspector de propiedades o en el inspector de componentes: `condenseWhite`, `editable`, `horizontalScrollPolicy`, `maxChars`, `restrict`, `text`, `verticalScrollPolicy` y `wordwrap`. Cada uno de estos parámetros tiene una propiedad de `ActionScript` del mismo nombre. Para obtener información los valores posibles de estos parámetros, consulte la clase `TextArea` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

La previsualización dinámica de cada instancia de `TextArea` refleja los cambios de parámetros realizados durante la edición en el inspector de propiedades o el inspector de componentes. Si se necesita una barra de desplazamiento, ésta aparece en la previsualización dinámica, aunque no funciona. No es posible seleccionar texto en la previsualización dinámica, de igual modo que tampoco se puede introducir texto en la instancia del componente en el escenario.

Es posible escribir código `ActionScript` para controlar éstas y otras opciones adicionales para el componente `TextArea` mediante sus propiedades, métodos y eventos. Para obtener más información, consulte la clase `TextArea` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente TextArea

El procedimiento siguiente explica cómo añadir un componente TextArea a una aplicación durante la edición. En el ejemplo, se configura un controlador de eventos `FocusOut` en la instancia de TextArea que verifica que el usuario ha escrito algo en el área de texto antes de seleccionar una parte diferente de la interfaz.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente TextArea del panel Componentes al escenario y asígnele el nombre de instancia **aTa**. Deje los parámetros establecidos en las opciones predeterminadas.
- 3 Arrastre un segundo componente TextArea del panel Componentes al escenario, colóquelo debajo del primero y asígnele el nombre de instancia **bTa**. Deje los parámetros establecidos en las opciones predeterminadas.
- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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();
}
```

En este ejemplo se limitan los caracteres que se pueden introducir en el área de texto `aTa` a caracteres en minúscula, comas, apóstrofes y espacios. También se configuran los controladores de eventos para los eventos `change`, `KEY_FOCUS_CHANGE` y `MOUSE_FOCUS_CHANGE` en el área de texto `aTa`. La función `changeHandler()` hace que el texto introducido en el área de texto `aTa` aparezca automáticamente en el área de texto `bTa`, mediante la asignación de `aTa.text` a `bTa.text` en cada evento `change`. La función `k_m_fHandler()` de los eventos `KEY_FOCUS_CHANGE` y `MOUSE_FOCUS_CHANGE` impide presionar la tecla Tabulador para desplazarse al siguiente campo sin introducir texto. Para ello, se impide el comportamiento predeterminado.

- 5 Seleccione Control > Probar película.

Si se presiona la tecla Tabulador para seleccionar la segunda área de texto sin introducir texto, aparecerá un mensaje de error y la selección deberá volver a la primera área de texto. A medida que se introduce texto en la primera área de texto, se verá duplicado en la segunda área de texto.

## Creación de una instancia del componente TextArea con ActionScript

En el ejemplo siguiente se crea un componente TextArea con ActionScript. El código establece la propiedad `condenseWhite` en `true` para condensar el espacio en blanco y asigna texto a la propiedad `htmlText` para sacar partido de los atributos de formato de texto HTML.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente TextArea al panel Biblioteca.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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);
```

En este ejemplo se utiliza la propiedad `htmlText` para aplicar atributos de negrita y subrayado de HTML a un bloque de texto y mostrarlo en el área de texto `a_ta`. En el ejemplo también se establece la propiedad `condenseWhite` en `true` a fin de condensar el espacio en blanco del bloque de texto. El método `setSize()` establece la altura y la anchura del área de texto, y el método `move()` define su posición. El método `addChild()` añade la instancia de `TextArea` al escenario.

#### 4 Seleccione Control > Probar película.

## Uso del componente TextInput

El componente `TextInput` es un componente de una sola línea de texto que ajusta el objeto `TextField` nativo de `ActionScript`. Si necesita un campo de texto de varias líneas, utilice el componente `TextArea`. Por ejemplo, si desea utilizar un componente `TextInput` como campo de contraseña en un formulario, se puede definir un detector que compruebe si el campo tiene caracteres suficientes cuando el usuario utiliza el tabulador para salir del campo. El detector podría mostrar un mensaje de error indicando que se debe introducir el número de caracteres correcto.

Se puede establecer la propiedad `textFormat` con el método `setStyle()` para cambiar el estilo de texto que aparece en una instancia de `TextInput`. El componente `TextInput` puede definirse con el formato HTML o como un campo de contraseña que disfraza el texto.

## Interacción del usuario con el componente TextInput

Es posible activar o desactivar el componente `TextInput` en una aplicación. Si está desactivado, no recibe la entrada del ratón ni del teclado. Cuando está activado, sigue las mismas reglas de selección y navegación que un objeto `TextField` de `ActionScript`. Cuando la instancia de `TextInput` está seleccionada, puede controlarla con las teclas siguientes:

| Tecla            | Descripción                                                                 |
|------------------|-----------------------------------------------------------------------------|
| Teclas de flecha | Desplaza el punto de inserción un carácter hacia la izquierda o la derecha. |
| Mayús+Tabulador  | Desplaza la selección al objeto anterior.                                   |
| Tabulador        | Desplaza la selección al objeto siguiente.                                  |

Para obtener más información sobre el control de la selección, consulte la clase `FocusManager` en la [Referencia de ActionScript 3.0 para Flash Professional](#) y en “Trabajo con `FocusManager`” en la página 29.

La previsualización dinámica de cada instancia de `TextInput` refleja los cambios de parámetros realizados durante la edición en el inspector de propiedades o el inspector de componentes. No es posible seleccionar texto en la previsualización dinámica, de igual modo que tampoco se puede introducir texto en la instancia del componente en el escenario.

Cuando se añade el componente `TextInput` a una aplicación, es posible utilizar el panel Accesibilidad para que los lectores de pantalla puedan acceder al mismo.

## Parámetros del componente `TextInput`

Puede establecer los siguientes parámetros de edición para cada instancia del componente `TextInput` en el inspector de propiedades o en el inspector de componentes: `editable`, `displayAsPassword`, `maxChars`, `restrict` y `text`. Cada uno de estos parámetros tiene una propiedad de `ActionScript` del mismo nombre. Para obtener información sobre los valores posibles de estos parámetros, consulte la clase `TextInput` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

Es posible escribir código `ActionScript` para controlar éstas y otras opciones adicionales para el componente `TextInput` mediante sus propiedades, métodos y eventos. Para obtener más información, consulte la clase `TextInput` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente `TextInput`

En el procedimiento siguiente se explica cómo añadir un componente `TextInput` a una aplicación. En el ejemplo se utilizan dos campos de `TextInput` para recibir y confirmar una contraseña. Se utiliza un detector de eventos para comprobar que se ha introducido un mínimo de ocho caracteres y que coincide el texto de los dos campos.

- 1 Cree un nuevo documento de Flash (`ActionScript 3.0`).
- 2 Arrastre un componente `Label` del panel Componentes al escenario y asígnele los valores siguientes en el inspector de propiedades:
  - Introduzca el nombre de instancia **pwdLabel**.
  - Introduzca un valor de anchura de **100**.
  - Introduzca un valor X de **50**.
  - Introduzca un valor Y de **150**.
  - En la sección Parámetros, introduzca el valor de **Password** del parámetro de texto.
- 3 Arrastre un segundo componente `Label` del panel Componentes al escenario y asígnele los valores siguientes:
  - Introduzca el nombre de instancia **confirmLabel**.
  - Introduzca un valor de anchura de **100**.
  - Introduzca un valor X de **50**.
  - Introduzca un valor Y de **200**.
  - En la sección Parámetros, introduzca el valor de **ConfirmPassword** del parámetro de texto.
- 4 Arrastre un componente `TextInput` del panel Componentes al escenario y asígnele los valores siguientes:
  - Introduzca el nombre de instancia **pwdTi**.
  - Introduzca un valor de anchura de **150**.
  - Introduzca un valor X de **190**.
  - Introduzca un valor Y de **150**.

- En la sección Parámetros, haga doble clic en el valor del parámetro `displayAsPassword` y seleccione **true**. Esta acción hace que el valor introducido en el campo de texto se enmascare con asteriscos.
- 5 Arrastre un segundo componente `TextInput` del panel Componentes al escenario y asígnele los siguientes valores:
- Introduzca el nombre de instancia **confirmTi**.
  - Introduzca un valor de anchura de **150**.
  - Introduzca un valor X de **190**.
  - Introduzca un valor Y de **200**.
  - En la sección Parámetros, haga doble clic en el valor del parámetro `displayAsPassword` y seleccione **true**. Esta acción hace que el valor introducido en el campo de texto se enmascare con asteriscos.
- 6 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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);
```

Este código configura un controlador de eventos `enter` en la instancia de `TextInput` denominada `confirmTi`. Si no coinciden las dos contraseñas, o bien si el usuario escribe menos de ocho caracteres, el ejemplo muestra el mensaje "Contraseña incorrecta. Vuelva a introducirla". Si las contraseñas constan de ocho caracteres o más y coinciden, el ejemplo muestra el valor introducido en el panel Salida.

- 7 Seleccione Control > Probar película.

## Creación de una instancia del componente `TextInput` con ActionScript

En el ejemplo siguiente se crea un componente `TextInput` con ActionScript. En el ejemplo también se crea un componente `Label` que se utiliza para solicitar al usuario que introduzca su nombre. En el ejemplo se establece la propiedad `restrict` del componente para permitir únicamente letras mayúsculas y minúsculas, un punto y un espacio. También se crea un objeto `TextFormat` que se utiliza para formatear el texto en los componentes `Label` y `TextInput`.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente `TextInput` del panel Componentes al panel Biblioteca.
- 3 Arrastre un componente `Label` del panel Componentes al panel Biblioteca.
- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película para ejecutar la aplicación.

## Uso del componente TileList

El componente `TileList` consta de una lista formada por filas y columnas que obtienen datos de un proveedor de datos. Un *elemento* hace referencia a una unidad de datos almacenados en una celda de `TileList`. Dicho elemento, que se origina en el proveedor de datos, suele tener una propiedad `label` y una propiedad `source`. La propiedad `label` identifica el contenido que se mostrará en una celda y la propiedad `source` le proporciona un valor.

Se puede crear una instancia de `Array`, o bien recuperarla de un servidor. El componente `TileList` tiene métodos que establecen una conexión proxy con su proveedor de datos, por ejemplo, los métodos `addItem()` y `removeItem()`. Si no se proporciona ningún proveedor de datos externo a la lista, estos métodos crean automáticamente una instancia de proveedor de datos que se presenta mediante `List.dataProvider`.

### Interacción del usuario con el componente TileList

Un componente `TileList` muestra cada celda con un objeto `Sprite` que implementa la interfaz `ICellRenderer`. Se puede especificar este elemento de representación con la propiedad `cellRenderer` de `TileList`. El elemento `CellRenderer` predeterminado del componente `TileList` es `ImageCell`, que muestra una imagen (clase, mapa de bits, instancia o URL) y una etiqueta opcional. La etiqueta es una línea única que siempre se alinea con la parte inferior de la celda. Sólo es posible desplazarse por `TileList` en una sola dirección.

Cuando una instancia de `TileList` está seleccionada, también se pueden utilizar las teclas siguientes para acceder a sus elementos:

| Tecla                           | Descripción                                                                                                                                                                                                                                                                                           |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Flecha arriba/flecha abajo      | Permiten moverse arriba y abajo en una columna. Si la propiedad <code>allowMultipleSelection</code> es <code>true</code> , se pueden utilizar estas teclas combinadas con la tecla Mayús para seleccionar varias celdas.                                                                              |
| Flecha izquierda/flecha derecha | Permiten moverse a la izquierda y a la derecha de una fila. Si la propiedad <code>allowMultipleSelection</code> es <code>true</code> , se pueden utilizar estas teclas combinadas con la tecla Mayús para seleccionar varias celdas.                                                                  |
| Inicio                          | Selecciona la primera celda de un componente <code>TileList</code> . Si la propiedad <code>allowMultipleSelection</code> es <code>true</code> , al mantener presionada la tecla Mayús mientras se presiona Inicio se seleccionarán todas las celdas desde la selección actual hasta la primera celda. |
| Fin                             | Selecciona la última celda de un componente <code>TileList</code> . Si la propiedad <code>allowMultipleSelection</code> es <code>true</code> , al mantener presionada la tecla Mayús mientras se presiona Fin se seleccionarán todas las celdas desde la selección actual hasta la última celda.      |
| Ctrl                            | Si la propiedad <code>allowMultipleSelection</code> se establece en <code>true</code> , permite seleccionar varias celdas, sin un orden específico.                                                                                                                                                   |

Si se añade el componente `TileList` a una aplicación, se puede hacer accesible a un lector de pantalla añadiendo las líneas de código `ActionScript` siguientes:

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

La accesibilidad de un componente sólo se activa una vez, sea cual sea su número de instancias. Para más información, consulte el Capítulo 18, "Creación de contenido accesible" de *Uso de Flash*.

## Parámetros del componente `TileList`

Puede establecer los siguientes parámetros de edición para cada instancia del componente `TileList` en el inspector de propiedades o en el inspector de componentes: `allowMultipleSelection`, `columnCount`, `columnWidth`, `dataProvider`, `direction`, `horizontalScrollLineSize`, `horizontalScrollPageSize`, `labels`, `rowCount`, `rowHeight`, `ScrollPolicy`, `verticalScrollLineSize` y `verticalScrollPageSize`. Cada uno de estos parámetros tiene una propiedad de `ActionScript` del mismo nombre. Para información sobre la utilización del parámetro `dataProvider`, consulte "Uso del parámetro `dataProvider`" en la página 31.

Se puede escribir código `ActionScript` para definir opciones adicionales de instancias de `TileList` mediante sus métodos, propiedades y eventos. Para obtener más información, consulte la clase `TileList` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente `TileList`

En este ejemplo se utilizan componentes `MovieClip` para rellenar un componente `TileList` con un conjunto de colores.

- 1 Cree un nuevo documento de Flash (`ActionScript 3.0`).
- 2 Arrastre un componente `TileList` al escenario y asígnele el nombre de instancia `aTL`.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código `ActionScript` siguiente:

```
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 Seleccione Control > Probar película para probar la aplicación.

### Creación de un componente TileList con ActionScript

Este ejemplo crea dinámicamente una instancia de TileList y le añade instancias de los componentes ColorPicker, ComboBox, NumericStepper y CheckBox. Se crea una instancia de Array que contiene etiquetas y nombres del componente que se mostrará, y se asigna dicha instancia (dp) a la propiedad dataProvider de TileList. Se utilizan las propiedades columnWidth y rowHeight y el método setSize() para disponer el componente TileList, el método move() para colocarlo en el escenario, el estilo contentPadding para insertar un espacio entre los bordes de la instancia de TileList y su contenido, y el método sortItemsOn() para ordenar el contenido mediante sus etiquetas.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre los siguientes componentes del panel Componentes al panel Biblioteca: ColorPicker, ComboBox, NumericStepper, CheckBox y TileList.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película para probar la aplicación.

## Uso del componente UI Loader

El componente UI Loader es un contenedor que puede mostrar archivos SWF, JPEG, JPEG progresivo, PNG y GIF. Se puede utilizar el componente UI Loader cuando sea necesario recuperar contenido de una ubicación remota e insertarla en una aplicación Flash. Por ejemplo, se puede utilizar el componente UI Loader para añadir el logotipo de una empresa (archivo JPEG) a un formulario. Asimismo, se podría utilizar un componente UI Loader en una aplicación que muestre fotografías. El método `load()` se utiliza para cargar contenido, la propiedad `percentLoaded` para determinar la cantidad de contenido que se ha cargado y el evento `complete` para determinar cuándo ha terminado la carga.

Es posible ajustar la escala del contenido de UI Loader, o bien cambiar el tamaño de UI Loader para que se ajuste al tamaño del contenido. De forma predeterminada, se cambia la escala del contenido para que se ajuste a UI Loader. También se puede cargar el contenido en tiempo de ejecución y controlar el progreso de carga (aunque después de que el contenido se haya cargado, se almacena en la caché y el progreso salta al 100% rápidamente). Si se especifica una ubicación al cargar contenido en UI Loader, se debe especificar dicha ubicación (coordenadas X e Y) como 0,0.

### Interacción del usuario con el componente UI Loader

El componente UI Loader no puede seleccionarse. Sin embargo, sí se puede seleccionar el contenido cargado en el componente UI Loader e interactuar con él. Para obtener más información sobre el control de la selección, consulte la clase `FocusManager` en la [Referencia de ActionScript 3.0 para Flash Professional](#) y en “[Trabajo con FocusManager](#)” en la página 29.

## Parámetros del componente UI Loader

Puede establecer los siguientes parámetros de edición para cada instancia del componente UI Loader en el inspector de propiedades o en el inspector de componentes: `autoLoad`, `maintainAspectRatio`, `source` y `scaleContent`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre.

La previsualización dinámica de cada instancia de UI Loader refleja los cambios de parámetros realizados durante la edición en el inspector de propiedades o el inspector de componentes.

Se puede escribir código ActionScript para definir opciones adicionales para instancias de UI Loader con sus métodos, propiedades y eventos. Para obtener más información, consulte la clase UI Loader en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente UI Loader

En el procedimiento siguiente se explica cómo añadir un componente UI Loader a una aplicación durante la edición. En este ejemplo, se carga una imagen GIF de un logotipo.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente UI Loader del panel Componentes al escenario.
- 3 En el inspector de propiedades, introduzca el nombre de instancia **aUI**.
- 4 Seleccione el cargador en el escenario y en el inspector de componentes, e introduzca <http://www.helpexamples.com/images/logo.gif> para el parámetro `source`.

## Creación de una instancia del componente UI Loader con ActionScript

En este ejemplo, se crea un componente UI Loader mediante ActionScript y se carga una imagen JPEG de una flor. Cuando se produce el evento `complete`, muestra el número de bytes cargados en el panel Salida.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente UI Loader del panel Componentes al panel Biblioteca.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
import fl.containers.UILoader;

var aLoader:UILoader = new UILoader();
aLoader.source = "http://www.flash-mx.com/images/imagen1.jpg";
aLoader.scaleContent = false;
addChild(aLoader);

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

- 4 Seleccione Control > Probar película.

## Uso del componente UIScrollBar

El componente UIScrollBar permite añadir una barra de desplazamiento a un campo de texto. Se puede añadir una barra de desplazamiento a un campo de texto durante la edición o en tiempo de ejecución con ActionScript. Para utilizar el componente UIScrollBar, es necesario crear un campo de texto en el escenario y arrastrar el componente UIScrollBar del panel Componentes a cualquier cuadrante del recuadro de delimitación del campo de texto.

Si la longitud de la barra de desplazamiento es inferior al tamaño combinado de sus flechas de desplazamiento, no se mostrará correctamente. Uno de los botones de flecha quedará oculto detrás del otro. Flash no proporciona funciones de comprobación de errores para estos casos. Es aconsejable ocultar la barra de desplazamiento con ActionScript. Si se cambia el tamaño de la barra de desplazamiento de forma que no haya espacio suficiente para el cuadro de desplazamiento (deslizador), Flash oculta el cuadro de desplazamiento.

El componente UIScrollBar funciona como cualquier otra barra de desplazamiento. Contiene botones de flecha en ambos extremos y una guía de desplazamiento y un cuadro de desplazamiento (deslizador) en medio. Puede asociarse con cualquier borde de un campo de texto y utilizarse tanto vertical como horizontalmente.

Para obtener información sobre TextField, consulte la clase TextField en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Interacción del usuario con el componente UIScrollBar

A diferencia de muchos otros componentes, el componente UIScrollBar puede recibir entradas continuas del ratón como, por ejemplo, cuando el usuario mantiene presionado el botón del ratón en lugar de tener que hacer clic repetidas veces.

El componente UIScrollBar no permite la interacción con el teclado.

## Parámetros del componente UIScrollBar

Puede establecer los siguientes parámetros de edición para cada instancia del componente UIScrollBar en el inspector de propiedades o en el inspector de componentes: `direction` y `scrollTargetName`. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre.

Se puede escribir código ActionScript para definir opciones adicionales para instancias de UIScrollBar mediante métodos, propiedades y eventos de clase. Para obtener más información, consulte la clase UIScrollBar en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Creación de una aplicación con el componente UIScrollBar

En el procedimiento siguiente se describe cómo añadir un componente UIScrollBar a una aplicación durante la edición.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Cree un campo de texto dinámico con la altura suficiente para admitir una o dos líneas de texto y asígnele el nombre **myText** en el inspector de propiedades.
- 3 En el inspector de propiedades, establezca el Tipo de línea del campo de introducción de texto en Multilínea o en Multilínea sin ajuste si tiene previsto utilizar la barra de desplazamiento horizontalmente.
- 4 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente para llenar la propiedad `text`; de este modo, el usuario deberá desplazarse por el texto para verlo completamente:

```
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:** asegúrese de que el campo de texto en el escenario es lo suficientemente pequeño como para tener que desplazarse por él para poder ver todo el texto. Si no lo es, la barra de desplazamiento no aparece o puede aparecer simplemente como dos líneas sin control deslizador, que es la parte que se arrastra para desplazarse por el contenido.*

- 5 Compruebe si está activado el ajuste a objetos (Ver > Ajustar > Ajustar a objetos).
- 6 Arrastre una instancia de UIScrollBar del panel Componentes al campo de introducción de texto junto al lado donde desea asociarlo. El componente debe solaparse con el campo de texto al soltar el ratón, para que quede correctamente vinculado al campo. Asígnele el nombre de instancia **mySb**.

La propiedad `scrollTargetName` del componente se llena automáticamente con el nombre de instancia del campo de texto en el inspector de propiedades y el inspector de componentes. Si no aparece en la ficha Parámetros, es posible que no haya solapado suficientemente la instancia de UIScrollBar.

- 7 Seleccione Control > Probar película.

### Creación de una instancia del componente UIScrollBar con ActionScript

Se puede crear una instancia de UIScrollBar con ActionScript y asociarla con un campo de texto en tiempo de ejecución. En el ejemplo siguiente se crea una instancia de UIScrollBar orientada horizontalmente, que se adjunta a la parte inferior de la instancia del campo de texto denominada **myTxt** (cargada con texto desde una dirección URL). En el ejemplo también se define el tamaño de la barra de desplazamiento para que coincida con el tamaño del campo de texto:

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente ScrollBar al panel Biblioteca.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código ActionScript siguiente:

```
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 Seleccione Control > Probar película.

# Capítulo 5: Personalización de los componentes de la interfaz de usuario

## Personalización de componentes de interfaz de usuario

Para personalizar la apariencia de los componentes de las aplicaciones, es necesario modificar al menos uno de los siguientes elementos:

**Estilos** Cada componente tiene un conjunto de estilos que pueden definirse para especificar los valores que utiliza Flash para representar la apariencia del componente. Los estilos normalmente especifican los aspectos e iconos que se utilizan en un componente en cada uno de sus estados, además de los valores de relleno y formato de texto.

**Aspectos** Un *aspecto* contiene una colección de símbolos que forman la apariencia gráfica de los componentes en un estado determinado. Mientras que un estilo especifica el aspecto que se debe utilizar, el aspecto es un elemento gráfico que Flash utiliza para dibujar el componente. La *aplicación de aspectos* es el proceso mediante el cual se cambia la apariencia de un componente, modificando o sustituyendo sus gráficos.

*Nota: la apariencia predeterminada de los componentes ActionScript 3.0 podría considerarse un tema (Aeon Halo) pero estos aspectos se generan en los componentes. Los componentes ActionScript 3.0 no admiten los archivos de tema externos compatibles con los componentes ActionScript 2.0.*

## Definición de estilos

Los estilos de un componente normalmente especifican valores para sus aspectos, iconos, formato de texto y relleno cuando Flash dibuja el componente en sus diversos estados. Por ejemplo, Flash dibuja un componente Button con un aspecto distinto para mostrar su estado presionado, que se produce al hacer clic con el ratón sobre él, que para mostrar su estado normal o cuando no está presionado. También utiliza un aspecto distinto cuando está en estado desactivado, lo que se produce al establecer la propiedad `enabled` en `false`.

Se pueden definir estilos para los componentes en los niveles de documento, clase e instancia. Además, algunas propiedades de estilo pueden heredarse de un componente principal. Por ejemplo, el componente List hereda los estilos de ScrollBar cuando hereda de BaseScrollPane.

Es posible definir estilos para personalizar un componente de las siguientes formas:

- Definir los estilos en una instancia de componente. Se pueden cambiar las propiedades de texto y de color de una única instancia de componente. Este recurso es eficaz en determinadas situaciones, aunque puede tardar mucho tiempo si necesita definir propiedades individuales en todos los componentes de un documento.
- Definir los estilos de todos los componentes de un tipo de documento determinado. Si se desea aplicar un aspecto uniforme a todos los componentes de un determinado tipo, por ejemplo, todos los componentes CheckBox o Button de un documento, se pueden definir estilos en el nivel de componente.

Los componentes añadidos a un contenedor heredan los valores de las propiedades de estilo establecidas para el contenedor.

Flash no muestra los cambios realizados en las propiedades de estilo al visualizar componentes en el escenario mediante la función Previsualización dinámica.

## Aspectos básicos de la configuración de estilos

A continuación se destacan algunos puntos clave de la utilización de estilos:

**Herencia** Un componente secundario se ha diseñado para heredar un estilo del componente principal. No es posible definir la herencia de estilos en ActionScript.

**Precedencia** Si un estilo de un componente se define de varias formas, Flash utiliza el primer estilo que encuentra según su orden de precedencia. Flash busca estilos en el orden siguiente hasta que encuentra un valor:

- 1 Busca una propiedad de estilo en la instancia de componente.
- 2 Si el estilo es uno de los estilos heredados, Flash busca en la jerarquía de elementos principales un valor heredado.
- 3 Busca el estilo en el componente.
- 4 Busca una configuración global en StyleManager.
- 5 Si la propiedad aún no está definida, tiene el valor `undefined`.

## Acceso a los estilos predeterminados de un componente

Para acceder a los estilos predeterminados de un componente, se puede utilizar el método estático `getDefaultStyles()` en la clase de componente. Por ejemplo, el código siguiente obtiene los estilos predeterminados del componente `ComboBox` y muestra los valores predeterminados de las propiedades `buttonWidth` y `downArrowDownSkin`:

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

## Definición y obtención de los estilos en una instancia de un componente

Una instancia de un componente de interfaz de usuario puede llamar a los métodos `setStyle()` y `getStyle()` directamente para definir u obtener un estilo. La sintaxis siguiente define un estilo y un valor para una instancia de componente:

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

Esta sintaxis obtiene un estilo de una instancia de componente:

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

Debe tenerse en cuenta que el método `getStyle()` devuelve el tipo `Object` porque puede devolver varios estilos que tengan distintos tipos de datos. Por ejemplo, el código siguiente define el estilo de fuente de una instancia de `TextArea` (`aTa`) y, a continuación, lo obtiene a través del método `getStyle()`. En el ejemplo, el valor devuelto se convierte en un objeto `TextFormat` para asignarlo a una variable `TextFormat`. Si no se realizara la conversión, el compilador emitiría un error por intentar forzar una variable `Object` en una variable `TextFormat`.

```
import flash.text.TextFormat;

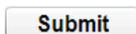
var tf:TextFormat = new TextFormat();
tf.font = "Georgia";
aTa.setStyle("textFormat", tf);
aTa.text = "Hello World!";
var aStyle:TextFormat = aTa.getStyle("textFormat") as TextFormat;
trace(aStyle.font);
```

## Uso de TextFormat para definir las propiedades de texto

El objeto TextFormat se utiliza para aplicar formato al texto de una instancia de componente. El objeto TextFormat tiene propiedades que permiten especificar características de texto como `bold`, `bullet`, `color`, `font`, `italic` y `size`, entre otras. Se pueden definir estas propiedades en el objeto TextFormat y luego llamar al método `setStyle()` para aplicarlas a una instancia de componente. Por ejemplo, el código siguiente define las propiedades `font`, `size` y `bold` de un objeto TextFormat y las aplica a una instancia 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);
```

En la siguiente ilustración se muestra el efecto de esta configuración en un botón con una etiqueta Submit (Enviar):



Las propiedades de estilo establecidas en una instancia de componente mediante `setStyle()` tienen la prioridad más alta y reemplazan las demás configuraciones de estilo. Sin embargo, cuanto más propiedades se definan mediante `setStyle()` en una sola instancia de componente, más lenta será la representación del componente en tiempo de ejecución.

## Definición de un estilo en todas las instancias de un componente

Para definir un estilo en todas las instancias de una clase de componente, se debe utilizar el método estático `setComponentStyle()` de la clase StyleManager. Por ejemplo, para establecer que el color del texto sea rojo en todos los botones, primero arrastre un componente Button al escenario y luego añada el siguiente código ActionScript al panel Acciones del fotograma 1 de la línea de tiempo:

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

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

Todos los componentes Button que añada posteriormente al escenario tendrán etiquetas rojas.

## Definición de un estilo en todos los componentes

Para definir un estilo en todos los componentes, se utiliza el método estático `setStyle()` de la clase StyleManager.

- 1 Arrastre un componente List al escenario y asígnele el nombre de instancia **aList**.
- 2 Arrastre un componente Button al escenario y asígnele el nombre de instancia **aButton**.
- 3 Presione **F9** o seleccione Acciones en el menú Ventana para abrir el panel Acciones, si aún no está abierto, e introduzca el siguiente código en el fotograma 1 de la línea de tiempo para establecer que el color del texto sea rojo en todos los componentes:

```
import fl.managers.StyleManager;

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

- 4 Añada el siguiente código al panel Acciones para llenar de texto el componente List.

```
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 Seleccione Control > Probar película o presione Ctrl+Intro para compilar el código y probar el contenido. El texto de la etiqueta del botón y de la lista debería ser de color rojo.

## Aspectos

La apariencia de un componente consta de elementos gráficos como un contorno, un color de relleno, iconos e incluso otros componentes. Un componente ComboBox, por ejemplo, contiene un componente List y éste, a su vez, contiene un componente ScrollBar. En conjunto, los elementos gráficos forman la apariencia del componente ComboBox. Sin embargo, la apariencia de un componente cambia en función de su estado actual. Por ejemplo, un componente CheckBox sin su etiqueta tiene una apariencia similar a la siguiente cuando aparece en la aplicación:



*Un componente CheckBox en su estado normal sin presionar*

Si se hace clic en el botón del ratón y se mantiene presionado sobre el componente CheckBox, pasa a tener esta apariencia:



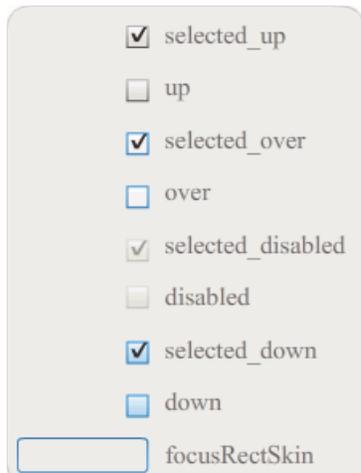
*Un componente CheckBox en su estado presionado*

Cuando se suelta el botón del ratón, el componente CheckBox recupera su apariencia original pero ahora contiene una marca de verificación que indica que se ha seleccionado.



*Un componente CheckBox en su estado seleccionado*

En conjunto, los iconos que representan el componente en sus diversos estados se denominan *aspectos del componente*. Para cambiar la apariencia de un componente en cualquiera de sus estados o en todos ellos, es necesario editar sus aspectos en Flash, como si se tratara de cualquier otro símbolo de Flash. Hay dos formas de acceder a los aspectos de un componente. La forma más sencilla es arrastrar el componente al escenario y hacer doble clic en él. Se abre una paleta de aspectos del componente similar a la siguiente para un componente CheckBox.



Aspectos de un componente CheckBox

También se puede acceder a los aspectos de un componente de forma individual en el panel Biblioteca. Cuando se arrastra un componente al escenario, también se copia a la biblioteca junto con una carpeta de sus activos y cualquier otro componente que contenga. Por ejemplo, si arrastra un componente ComboBox al escenario, el panel Biblioteca también tendrá los componentes List, ScrollBar y TextInput, incorporados en ComboBox, junto con una carpeta de aspectos de cada uno de estos componentes y una carpeta Activos compartidos que contiene elementos que comparten estos componentes. Para editar los aspectos de cualquiera de estos componentes, debe abrirse la carpeta de aspectos correspondiente (ComboBoxSkins, ListSkins, ScrollBarSkins o TextInputSkins) y hacer doble clic en el icono del aspecto que se desea editar. Al hacer doble clic en ComboBox\_downSkin, por ejemplo, se abre el aspecto en el modo de edición de símbolos, como se muestra en la siguiente ilustración:



ComboBox\_downSkin

## Creación de un aspecto nuevo

Si se desea crear un nuevo aspecto para un componente del documento, deben editarse los aspectos del componente para cambiar su apariencia. Para acceder a los aspectos de un componente, simplemente haga doble clic en el componente en el escenario para abrir una paleta de sus aspectos. A continuación, haga doble clic en el aspecto que desee editar para abrirlo en el modo de edición de símbolos. Por ejemplo, haga doble clic en el componente TextArea en el escenario para abrir sus activos en el modo de edición de símbolos. Establezca el control de zoom en 400%, o incluso en un valor superior, y edite el símbolo para cambiar su aspecto. Cuando haya terminado, el cambio afectará a todas las instancias del componente en el documento. Como alternativa, puede hacer doble clic en un aspecto concreto del panel Biblioteca para abrirlo en el escenario en el modo de edición de símbolos.

Es posible modificar los aspectos de un componente de las siguientes formas:

- Creación de un nuevo aspecto en todas las instancias
- Creación de nuevos aspectos en algunas instancias

## Creación de un aspecto en todas las instancias

Cuando se edita un aspecto de un componente, de forma predeterminada se cambia la apariencia del componente en todas sus instancias que aparecen en el documento. Si se desean crear distintas apariencias para un mismo componente, es necesario duplicar los aspectos que se quieren cambiar y asignarles nombres distintos, editarlos y luego definir los estilos apropiados para aplicarlos. Para más información, consulte “[Creación de aspectos en algunas instancias](#)” en la página 107.

En este capítulo se describe cómo modificar uno o varios aspectos en cada uno de los componentes de interfaz de usuario. Si se sigue uno de estos procedimientos para cambiar uno o varios aspectos de un componente de interfaz de usuario, se cambiará en todas las instancias que aparezcan en el documento.

## Creación de aspectos en algunas instancias

Para crear un aspecto en algunas instancias de un componente, utilice el siguiente procedimiento general:

- Seleccione el aspecto en la carpeta Activos del componente en el panel Biblioteca.
- Duplique el aspecto y asígnele un nombre de clase exclusivo.
- Edite el aspecto para darle la apariencia que desee.
- Llame al método `setStyle()` en la instancia del componente para asignar el nuevo aspecto al estilo de aspecto.

En el siguiente procedimiento se crea un nuevo aspecto `selectedDownSkin` en una de las dos instancias de `Button`.

- 1 Cree un documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre dos instancias de `Button` del panel Componentes al escenario y asígneles los nombres de instancia **aButton** y **bButton**.
- 3 Abra el panel Biblioteca y luego las carpetas `Component Assets` y `ButtonSkins` que contiene.
- 4 Haga clic en el aspecto `selectedDownSkin` para seleccionarlo.
- 5 Haga clic con el botón derecho del ratón para abrir el menú contextual y seleccione `Duplicar`.
- 6 En el cuadro de diálogo `Duplicar símbolo`, asigne al nuevo aspecto un nombre exclusivo como, por ejemplo, **Button\_mySelectedDownSkin**. Haga clic en `Aceptar`.
- 7 En la carpeta `Biblioteca > Component Assets > ButtonSkins`, seleccione `Button_mySelectedDownSkin` y haga clic con el botón derecho de ratón para abrir el menú contextual. Seleccione `Vinculación` para abrir el cuadro de diálogo `Propiedades de vinculación`.
- 8 Active la casilla de verificación `Exportar para ActionScript`. Deje activada la casilla de verificación `Exportar` en primer fotograma y asegúrese de que el nombre de clase es exclusivo. Haga clic en `Aceptar` y de nuevo en `Aceptar` en el mensaje de advertencia que indica que no se pudo encontrar una definición de clase y que se creará una.
- 9 Haga doble clic en el aspecto `Button_mySelectedDownSkin` en el panel Biblioteca para abrirlo en el modo de edición de símbolos.
- 10 Haga clic en el relleno azul en el centro del aspecto hasta que el color aparezca en el selector de color de relleno en el inspector de propiedades. Haga clic en el selector de color y seleccione el color `#00CC00` para rellenar el aspecto.
- 11 Haga clic en el botón `Atrás` en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 12 En el inspector de propiedades, haga clic en la ficha `Parámetros` en cada uno de los botones y establezca el parámetro `toggle` en `true`.
- 13 Añada el siguiente código al panel `Acciones`, en el fotograma 1 de la línea de tiempo:

```
bButton.setStyle("selectedDownSkin", Button_mySelectedDownSkin);  
bButton.setStyle("downSkin", Button_mySelectedDownSkin);
```

14 Seleccione Control > Probar película.

15 Haga clic en cada uno de los botones. Tenga en cuenta que el aspecto de estado presionado (seleccionado y no seleccionado) del objeto `bButton` utiliza el nuevo símbolo de aspecto.

## Personalización del componente Button

El componente `Button` puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable de la clase `Button`, como `height`, `width`, `scaleX` y `scaleY`.

Cuando se modifica el tamaño del botón, no cambia el tamaño del icono ni de la etiqueta. En recuadro de delimitación de un componente `Button` se corresponde con el borde de `Button` y también designa el área activa de la instancia. Si se aumenta el tamaño de la instancia, también aumenta el tamaño del área activa. Si el recuadro de delimitación es demasiado pequeño para que encaje la etiqueta, ésta se recorta para ajustarse al espacio disponible.

Si el componente `Button` tiene un icono que es más grande que `Button`, sobresaldrá por los bordes de éste.

### Uso de estilos con el componente Button

Los estilos de un componente `Button` normalmente especifican valores para sus aspectos, iconos, formato de texto y relleno cuando se dibuja el componente en sus diversos estados.

En el siguiente procedimiento se colocan dos instancias de `Button` en el escenario y se establece la propiedad `emphasized` en `true` en ambos componentes cuando el usuario hace clic en uno de ellos. También se establece el estilo `emphasizedSkin` del segundo componente `Button` en el estilo `selectedOverSkin` cuando el usuario hace clic en él, de modo que los dos componentes `Button` muestran distintos aspectos para un mismo estado.

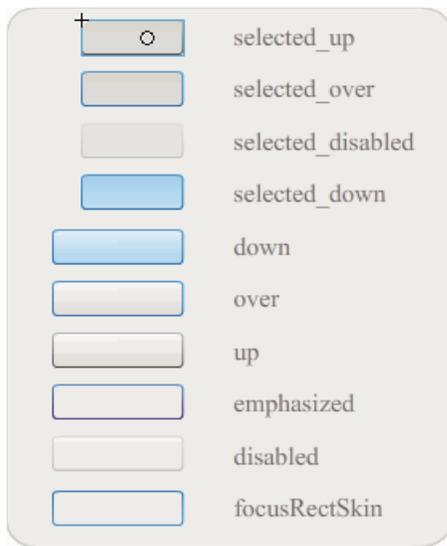
- 1 Cree un archivo de Flash (ActionScript 3.0).
- 2 Arrastre dos instancias de `Button` al escenario de forma consecutiva y asígneles los nombres de instancia **aBtn** y **bBtn**. En la ficha Parámetros del inspector de propiedades, asígneles las etiquetas `Button A` y `Button B`.
- 3 Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo:

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

- 4 Seleccione Control > Probar película.
- 5 Haga clic en uno de los botones para ver el efecto del estilo `emphasizedSkin` en cada uno de los botones.

### Uso de aspectos con el componente Button

El componente `Button` utiliza los siguientes aspectos, que se corresponden con sus diversos estados. Para editar uno o varios aspectos con el fin de cambiar la apariencia de `Button`, haga doble clic en la instancia de `Button` en el escenario para abrir su paleta de aspectos, como se muestra en la siguiente ilustración:

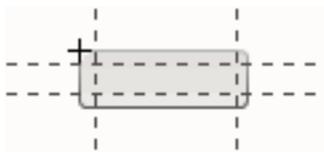


Aspectos de Button

Si un botón está activado, muestra el estado correspondiente al desplazamiento del puntero sobre él. Cuando se presiona el botón, recibe la selección de entrada y muestra el estado presionado. Cuando se suelta el botón del ratón, el botón recupera el estado correspondiente al desplazamiento del puntero sobre él. Si el puntero se desplaza fuera del botón estando el ratón presionado, el botón recupera su estado original. Si el parámetro `toggle` está establecido en `true`, se muestra el estado presionado con el aspecto `selectedDownSkin`, el estado sin presionar con el aspecto `selectedUpSkin` y el estado cuando se desplaza el puntero sobre él con el aspecto `selectedOverSkin`.

Si el componente Button está desactivado, muestra el estado desactivado sea cual sea la acción del usuario.

Para editar uno de los aspectos, haga doble clic en él para abrirlo en el modo de edición de símbolos, como se muestra en la siguiente ilustración:



Button en modo de edición de símbolos

En este punto, puede utilizar las herramientas de edición de Flash para editar el aspecto según sus preferencias.

El siguiente procedimiento cambia el color del aspecto `selected_over` de Button.

- 1 Cree un archivo de Flash (ActionScript 3.0).
- 2 Arrastre un componente Button del panel Componentes al escenario. En la ficha Parámetros, establezca el parámetro `toggle` en `true`.
- 3 Haga doble clic en el componente Button para abrir su paleta de aspectos.
- 4 Haga doble clic en el aspecto `selected_over` para abrirlo en el modo de edición de símbolos.
- 5 Establezca el control de zoom en 400% para ampliar el icono y poder editarlo.
- 6 Haga doble clic en el fondo hasta que aparezca su color en el selector de color de relleno en el inspector de propiedades.
- 7 En el selector de color de relleno, seleccione el color `#CC0099` para aplicarlo al fondo del aspecto `selected_over`.

- 8 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 9 Seleccione Control > Probar película.
- 10 Haga clic en el botón para que pase al modo seleccionado.

Cuando mueva el puntero del ratón sobre el componente Button, el estado `selected_over` debería aparecer del modo que se muestra en la siguiente ilustración.



*Button con el aspecto `selected_over` con el color modificado*

## Personalización del componente CheckBox

El componente CheckBox puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable de la clase CheckBox. Por ejemplo, para cambiar el tamaño de un componente CheckBox, se pueden definir sus propiedades `height`, `width`, `scaleX` y `scaleY`. Cuando se modifica el tamaño del componente CheckBox, no cambia el tamaño de la etiqueta ni del icono de la casilla de verificación; sólo cambia el tamaño del recuadro de delimitación.

El recuadro de delimitación de una instancia de CheckBox es invisible y designa el área activa de la instancia. Si se aumenta el tamaño de la instancia, también aumenta el tamaño del área activa. Si el recuadro de delimitación es demasiado pequeño para que encaje la etiqueta, ésta se recorta para ajustarse al espacio disponible.

### Uso de estilos con CheckBox

Es posible definir propiedades de estilo para cambiar el aspecto de una instancia de CheckBox. Por ejemplo, en el siguiente procedimiento se cambia el tamaño y el color de una etiqueta de CheckBox.

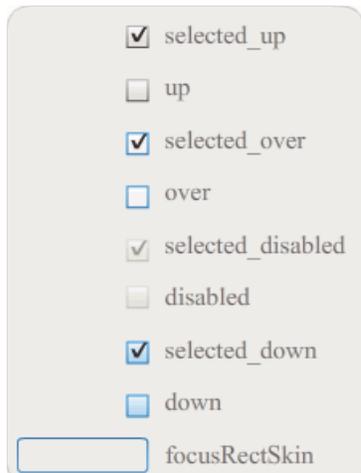
- 1 Arrastre un componente CheckBox del panel Componentes al escenario y asígnele el nombre de instancia **myCb**.
- 2 Haga clic en la ficha Parámetros en el inspector de propiedades e introduzca el siguiente valor en el parámetro de etiqueta: **¿Menos de 500 EUR?**
- 3 Haga clic en el fotograma 1 de la línea de tiempo principal e introduzca el siguiente código en el panel Acciones:

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

Para más información, consulte “Definición de estilos” en la página 102. Para obtener información sobre la definición de las propiedades de estilo para cambiar los iconos y aspectos del componente, consulte “Creación de un aspecto nuevo” en la página 106 y “Uso de aspectos con CheckBox” en la página 110.

### Uso de aspectos con CheckBox

El componente CheckBox tiene los siguientes aspectos, que pueden editarse para cambiar su apariencia.



Aspectos de CheckBox

En este ejemplo se cambia el color de contorno y el color de fondo del componente en sus estados `up` y `selectedUp`. Para cambiar los aspectos de otros estados hay que seguir pasos similares.

- 1 Cree un nuevo documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre el componente CheckBox al escenario, de forma que también se coloca en la biblioteca con una carpeta de sus activos.
- 3 Haga doble clic en el componente CheckBox situado en el escenario para abrir su panel de iconos de aspecto.
- 4 Haga doble clic en el icono `selected_up` para abrirlo en el modo de edición de símbolos.
- 5 Establezca el control de zoom en 800% para ampliar el icono y poder editarlo.
- 6 Haga clic en el borde de CheckBox para seleccionarlo. Con el selector de color de relleno en el inspector de propiedades, seleccione el color #0033FF para aplicarlo al borde.
- 7 Haga doble clic en el fondo del componente CheckBox para seleccionarlo y utilice de nuevo el selector de color de relleno para establecer que el color de fondo sea #00CCFF.
- 8 Repita los pasos 4 a 8 para el aspecto `up` de CheckBox.
- 9 Seleccione Control > Probar película.

## Personalización del componente ColorPicker

La única modificación de tamaño que se puede hacer en ColorPicker es a través de sus estilos: `swatchWidth`, `swatchHeight`, `backgroundPadding`, `textFieldWidth` y `textFieldHeight`. Si se intenta cambiar el tamaño de ColorPicker con la herramienta Transformación o con ActionScript, utilizando el método `setSize()` o las propiedades `width`, `height`, `scaleX` o `scaleY`, estos valores se omiten al crear el archivo SWF y el componente ColorPicker se muestra con su tamaño predeterminado. El fondo de la paleta cambiará de tamaño para ajustarse al número de columnas definidas mediante `setStyle()` en el estilo `columnCount`. El número predeterminado de columnas es 18. Se puede establecer que haya 1024 colores personalizados y la paleta cambiará de tamaño verticalmente para acomodar el número de muestras de color.

## Uso de estilos con el componente ColorPicker

Es posible definir varios estilos para cambiar la apariencia del componente ColorPicker. Por ejemplo, en el siguiente procedimiento se cambia el número de columnas (`columnCount`) de ColorPicker a 12, se cambia la altura (`swatchHeight`) y anchura (`swatchWidth`) de las muestras de color y se cambia el relleno del campo de texto (`textPadding`) y del fondo (`backgroundPadding`).

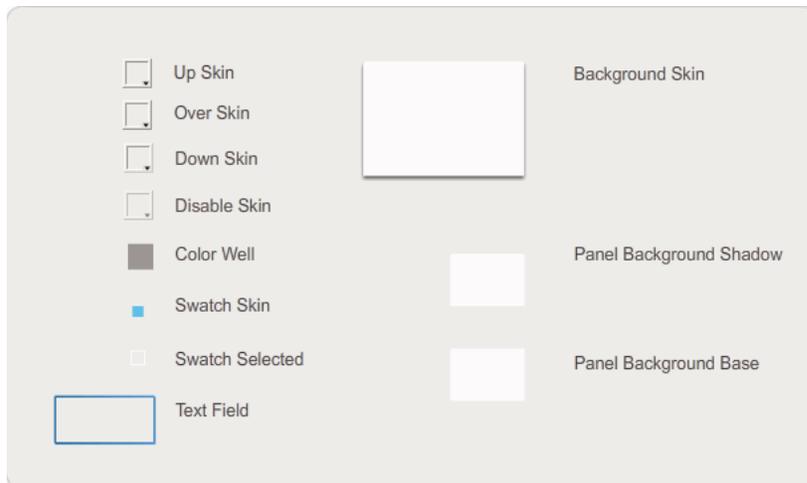
- 1 Cree un nuevo documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre el componente ColorPicker al escenario y asígnele el nombre de instancia **aCp**.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código siguiente:

```
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 Seleccione Control > Probar película.
- 5 Haga clic en el componente ColorPicker para abrirlo y ver el modo en que esta configuración ha modificado su apariencia.

## Uso de aspectos con el componente ColorPicker

El componente ColorPicker utiliza los siguientes aspectos para representar sus estados visuales.



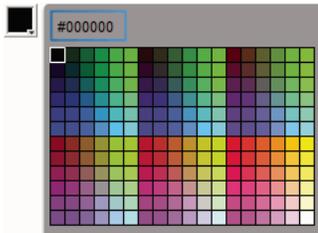
*Aspectos de ColorPicker*

Se puede cambiar el color de un aspecto Background para cambiar el color del fondo de la paleta.

- 1 Cree un nuevo documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre el componente ColorPicker al escenario.
- 3 Haga doble clic en él para abrir su paleta de aspectos.
- 4 Haga doble clic en el aspecto Background hasta seleccionarlo; el selector de color de relleno aparecerá en el inspector de propiedades.
- 5 En el selector de color de relleno, seleccione el color #999999 para aplicarlo al aspecto Background.

- 6 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 7 Seleccione Control > Probar película.

Al hacer clic en ColorPicker, el fondo de la paleta debería ser gris, tal y como se muestra en la siguiente ilustración.



*ColorPicker con el aspecto Background gris oscuro*

## Personalización del componente ComboBox

El componente ComboBox puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable de la clase ComboBox, como `height`, `width`, `scaleX` y `scaleY`.

El componente ComboBox cambiará de tamaño para ajustarse a la anchura y altura especificadas. La lista cambiará de tamaño para ajustarse a la anchura del componente, a menos que se haya definido la propiedad `dropdownWidth`.

Si el texto es demasiado largo para encajar en el componente ComboBox, se recorta para ajustarse al espacio disponible. Se debe cambiar el tamaño del componente ComboBox y definir la propiedad `dropdownWidth` para que quepa el texto.

### Uso de estilos con el componente ComboBox

Es posible definir propiedades de estilo para cambiar el aspecto de un componente ComboBox. Los estilos especifican los valores de aspectos, procesador de celdas, relleno y anchura del botón del componente. En el siguiente ejemplo se definen los estilos `buttonWidth` y `textPadding`. El estilo `buttonWidth` establece la anchura del área activa del botón y se aplica cuando el componente ComboBox puede editarse y sólo se puede presionar el botón para abrir la lista desplegable. El estilo `textPadding` determina la cantidad de espacio entre el borde exterior del campo de texto y el texto. Si se aumenta la altura del componente ComboBox, resulta útil centrar el texto verticalmente en el campo de texto. De lo contrario, podría parecer que el texto está en la parte superior del campo de texto.

- 1 Cree un nuevo documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre el componente ComboBox al escenario y asígnele el nombre de instancia **aCb**.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código siguiente:

```
import fl.data.DataProvider;

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

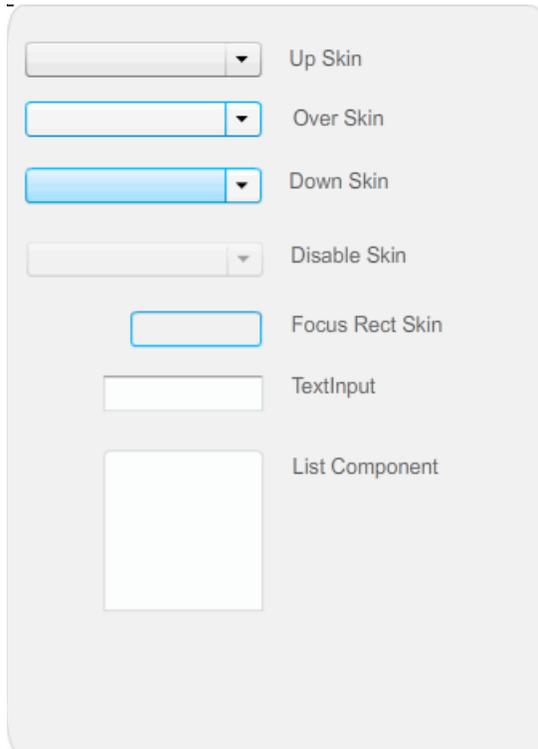
var items:Array = [
    {label:"San Francisco", data:"601 Townsend St."},
    {label:"San Jose", data:"345 Park Ave."},
    {label:"San Diego", data:"10590 West Ocean Air Drive, Suite 100"},
    {label:"Santa Rosa", data:"2235 Mercury Way, Suite 105"},
    {label:"San Luis Obispo", data:"3220 South Higuera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

#### 4 Seleccione Control > Probar película.

Se debe tener en cuenta que el área del botón donde se puede hacer clic para abrir la lista desplegable es únicamente un área estrecha en el lado derecho. También hay que tener en cuenta que el texto aparece centrado verticalmente en el campo de texto. Se puede intentar ejecutar el ejemplo sin las dos sentencias `setStyle()` para comprobar su efecto.

## Uso de aspectos con ComboBox

El componente ComboBox utiliza los siguientes aspectos para representar sus estados visuales:



Aspectos de ComboBox

Se puede cambiar el color del aspecto Up para cambiar el color del componente en su estado inactivo en el escenario.

- 1 Cree un nuevo documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre un componente ComboBox al escenario.
- 3 Haga doble clic en él para abrir su paleta de aspectos.
- 4 Haga doble clic en el aspecto Up hasta que se seleccione y se abra para editarlo.
- 5 Establezca el control de zoom en 400%.
- 6 Haga clic en el área central del aspecto hasta que aparezca su color en el selector de color de relleno en el inspector de propiedades.
- 7 En el selector de color de relleno, seleccione el color #33FF99 para aplicarlo al aspecto Up.
- 8 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 9 Seleccione Control > Probar película.

El componente ComboBox debería aparecer en el escenario tal y como se muestra en la siguiente ilustración.



*ComboBox con color personalizado en el aspecto de fondo*

## Personalización del componente DataGrid

El componente DataGrid puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable, como `width`, `height`, `scaleX` y `scaleY`. Si no dispone de ninguna barra de desplazamiento horizontal, las anchuras de las columnas se ajustarán de forma proporcional. Si se ajusta el tamaño de una columna, y por lo tanto de una celda, es posible que el texto aparezca recortado.

### Uso de estilos con el componente DataGrid

Es posible definir propiedades de estilo para cambiar el aspecto de un componente DataGrid. El componente DataGrid hereda estilos del componente List. (Consulte “[Uso de estilos con el componente List](#)” en la página 121).

### Definición de estilos para una columna individual

Un objeto DataGrid puede tener varias columnas y se pueden especificar distintos procesadores de celdas para cada columna. Todas las columnas de un objeto DataGrid se representan con un objeto DataGridColumn, y la clase DataGridColumn incluye una propiedad `cellRenderer`, para la que se puede definir la clase CellRenderer para la columna.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente DataGrid al panel Biblioteca.
- 3 Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo. Este código crea un componente DataGrid con una cadena larga de texto en la tercera columna. Al final, establece la propiedad `cellRenderer` de la columna en el nombre de un procesador de celdas que represente una celda de varias líneas.

```

/* 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 Guarde el archivo FLA con el nombre MultiLineGrid fla.
- 5 Cree un archivo de ActionScript.
- 6 Copie el siguiente código ActionScript en la ventana 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 Guarde el archivo de ActionScript como MultiLineCell.as en la misma carpeta donde guardó el archivo MultiLineGrid fla.

8 Vuelva a la aplicación MultiLineGrid fla y seleccione Control > Probar película.

El componente DataGrid debería tener un aspecto similar al siguiente:

| 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 la aplicación MultiLineGrid fla

### Definición de estilos de encabezado

Para definir el estilo de texto de una fila de encabezado, se puede utilizar el estilo `headerTextFormat`. En el siguiente ejemplo se utiliza el objeto `TextFormat` para establecer que el estilo `headerTextFormat` utilice la fuente Arial, el color rojo, un tamaño de fuente de 14 y la cursiva.

- 1 Cree un nuevo documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre el componente DataGrid al escenario y asígnele el nombre de instancia **aDg**.
- 3 Abra el panel Acciones, seleccione el fotograma 1 en la línea de tiempo principal e introduzca el código siguiente:

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

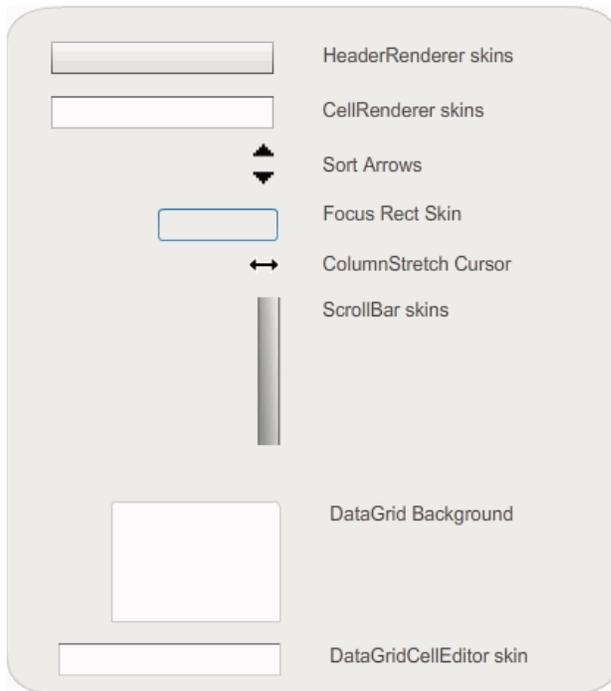
var myDP:Array = new Array();
myDP = [{FirstName:"Winston", LastName:"Elstad"},
        {FirstName:"Ric", LastName:"Dietrich"},
        {FirstName:"Ewing", LastName:"Canepa"},
        {FirstName:"Kevin", LastName:"Wade"},
        {FirstName:"Kimberly", LastName:"Dietrich"},
        {FirstName:"AJ", LastName:"Bilow"},
        {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 Control > Probar película para ejecutar la aplicación.

## Uso de aspectos con el componente DataGrid

El componente DataGrid utiliza los siguientes aspectos para representar sus estados visuales:



*Aspectos de DataGrid*

El aspecto CellRenderer se utiliza en las celdas del cuerpo del componente DataGrid y el aspecto HeaderRenderer se utiliza en la fila de encabezado. El siguiente procedimiento cambia el color de fondo de la fila de encabezado pero se podría utilizar el mismo proceso para cambiar el color de fondo de las celdas del cuerpo del componente DataGrid si se editara el aspecto CellRenderer.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente DataGrid al escenario y asígnele el nombre de instancia **aDg**.
- 3 Haga doble clic en el componente para abrir su paleta de aspectos.
- 4 Establezca el control de zoom en 400% para ampliar los iconos y poder editarlos.
- 5 Haga doble clic en el aspecto HeaderRenderer para abrir su paleta de aspectos.
- 6 Haga doble clic en Up\_Skin para abrirlo en el modo de edición de símbolos y haga clic en el fondo hasta que se seleccione y aparezca el selector de color de relleno en el inspector de propiedades.
- 7 En el selector de color de relleno, seleccione el color #00CC00 para aplicarlo al fondo del aspecto Up\_Skin HeaderRenderer.
- 8 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 9 Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo, para añadir datos al componente 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 Seleccione Control > Probar película para probar la aplicación.

El componente DataGrid debería aparecer como se muestra en la siguiente ilustración, con el fondo de la fila de encabezado de color verde.

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

*DataGrid con fondo de fila de encabezado personalizado*

## Personalización del componente Label

El componente Label puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. También es posible definir el parámetro de edición `autoSize`; en tal caso, el recuadro de delimitación de la previsualización dinámica no varía, pero el componente Label sí cambia de tamaño. El componente Label cambia de tamaño en función del parámetro `wordwrap`. Si el parámetro es `true`, el componente Label cambia de tamaño verticalmente para que quepa el texto. Si el parámetro es `false`, el componente Label cambia de tamaño horizontalmente. En tiempo de ejecución, utilice el método `setSize()`. Para obtener más información, consulte el método `Label.setSize()` y la propiedad `Label.autoSize` en la *Referencia de ActionScript 3.0 para Flash Professional*. También consulte “[Creación de una aplicación con el componente Label](#)” en la página 65.

## Uso de estilos con el componente Label

Es posible definir propiedades de estilo para cambiar el aspecto de una instancia de Label. Todo el texto de la instancia del componente Label debe compartir el mismo estilo. El componente Label tiene un estilo `textFormat`, que tiene los mismos atributos que el objeto `TextFormat` y que permite definir las mismas propiedades para el contenido de `Label.text` que un componente `TextField` normal de Flash. En el siguiente ejemplo se establece que el color de texto de una etiqueta sea rojo.

- 1 Arrastre un componente Label del panel Componentes al escenario y asígnele el nombre de instancia `a_label`.
- 2 Haga clic en la ficha Parámetros y reemplace el valor de la propiedad `text` con el texto:

### Aplicar rojo

- 3 Seleccione el fotograma 1 de la línea de tiempo principal, abra el panel Acciones e introduzca el código siguiente:

```
/* 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 Seleccione Control > Probar película.

Para obtener más información sobre los estilos de Label, consulte la clase Label en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Aspectos y el componente Label

El componente Label no tiene elementos visuales a los que aplicar aspectos.

## Personalización del componente List

El componente List puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable de la clase List, como `height`, `width`, `scaleX` y `scaleY`.

Cuando se cambia el tamaño de una lista, las filas se contraen horizontalmente y recortan el texto que contienen. Verticalmente, la lista añade o elimina las filas necesarias. Las barras de desplazamiento se colocan automáticamente según convenga.

## Uso de estilos con el componente List

Es posible definir propiedades de estilo para cambiar el aspecto de un componente List. Los estilos especifican los valores de los aspectos y el relleno del componente cuando éste se dibuja.

Los diversos estilos de aspecto permiten especificar distintas clases que se utilizan en el aspecto. Para obtener más información sobre cómo utilizar estilos de aspecto, consulte “Aspectos” en la página 105.

El siguiente procedimiento define el valor del estilo `contentPadding` en el componente `List`. Debe tenerse en cuenta que el valor de esta configuración se resta del tamaño del componente `List` para obtener el relleno que rodea el contenido, de modo que quizás sea necesario aumentar el tamaño del componente `List` para evitar que el texto de `List` quede recortado.

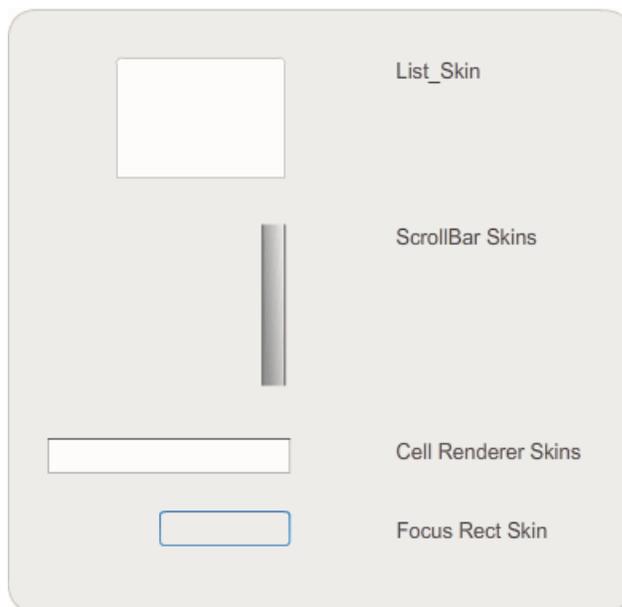
- 1 Cree un nuevo documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre un componente `List` del panel Componentes al escenario y asígnele el nombre de instancia `aList`.
- 3 Seleccione el fotograma 1 en la línea de tiempo principal, abra el panel Acciones e introduzca el siguiente código, que define el estilo `contentPadding` y añade datos al componente `List`:

```
aList.setStyle("contentPadding", 5);  
aList.setSize(145, 200);  
aList.addItem({label:"1956 Chevy (Cherry Red)", data:35000});  
aList.addItem({label:"1966 Mustang (Classic)", data:27000});  
aList.addItem({label:"1976 Volvo (Xcllnt Cond)", data:17000});  
aList.rowCount = aList.length;
```

- 4 Seleccione Control > Probar película.

## Uso de aspectos con el componente List

El componente `List` utiliza los siguientes aspectos para representar sus estados visuales:

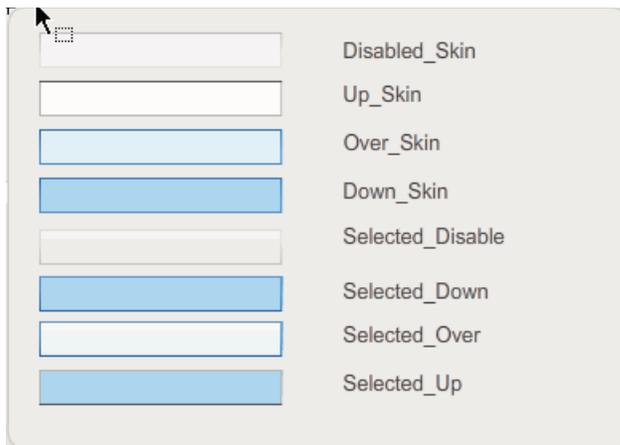


Aspectos de List

Para más información sobre la aplicación de aspectos en `ScrollBar`, consulte “[Personalización del componente UIScrollBar](#)” en la página 137. Para más información sobre la aplicación de aspectos en el aspecto `Focus Rect`, consulte “[Personalización del componente TextArea](#)” en la página 131.

**Nota:** si se cambia el aspecto `ScrollBar` en un componente, también cambiará en todos los demás componentes que lo utilicen.

Haga doble clic en el aspecto `Cell Renderer` para abrir una segunda paleta de aspectos para los distintos estados de una celda del componente `List`.



Aspectos de procesador de celdas de List

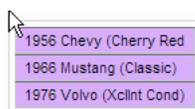
Se pueden editar estos aspectos para cambiar la apariencia de las celdas del componente List. El siguiente procedimiento cambia el color del aspecto Up para cambiar la apariencia de List en su estado normal inactivo.

- 1 Cree un nuevo documento de archivo de Flash (ActionScript 3.0).
- 2 Arrastre un componente List del panel Componentes al escenario y asígnele el nombre de instancia **aList**.
- 3 Haga doble clic en el componente List para abrir su paleta de aspectos.
- 4 Haga doble clic en el aspecto Cell Renderer para abrir la paleta de aspectos Cell Renderer.
- 5 Haga doble clic en el aspecto Up\_Skin para abrirlo y poder editarlo.
- 6 Haga clic en el área de relleno del aspecto para seleccionarla. Debería aparecer un selector de color de relleno en el inspector de propiedades, con el color de relleno actual del aspecto.
- 7 En el selector de color de relleno, seleccione el color #CC66FF para aplicarlo al relleno del aspecto Up\_Skin.
- 8 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 9 Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo, para añadir datos al componente 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 Seleccione Control > Probar película.

El componente List debería aparecer como se muestra en la siguiente ilustración:



Celdas de List sin color de Up\_Skin personalizado

El encuadre se obtiene de la configuración del estilo `contentPadding`.

## Personalización del componente NumericStepper

El componente NumericStepper puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad o método aplicable de la clase NumericStepper, como `width`, `height`, `scaleX` y `scaleY`.

Si se cambia el tamaño del componente NumericStepper, no cambia la anchura de los botones de flecha arriba y flecha abajo. Si al componente NumericStepper se le asigna un tamaño superior a la altura predeterminada, el comportamiento predeterminado fija los botones de flecha en la parte superior e inferior del componente. De lo contrario, la escala de 9 divisiones determina cómo se dibujan los botones. Los botones de flecha siempre aparecen a la derecha del cuadro de texto.

### Estilos y el componente NumericStepper

Es posible definir propiedades de estilo para cambiar la apariencia del componente NumericStepper. Los estilos especifican los valores de los aspectos, el relleno y el formato de texto del componente cuando éste se dibuja. El estilo `textFormat` permite cambiar el tamaño y la apariencia del valor de NumericStepper. Los diversos estilos de aspecto permiten especificar distintas clases que se utilizan en el aspecto. Para obtener más información sobre cómo utilizar estilos de aspecto, consulte “Aspectos” en la página 105.

Este procedimiento utiliza el estilo `textFormat` para cambiar la apariencia del valor que muestra NumericStepper.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente NumericStepper del panel Componentes al escenario y asígnele el nombre de instancia `myNs`.
- 3 Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo 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 Seleccione Control > Probar película.

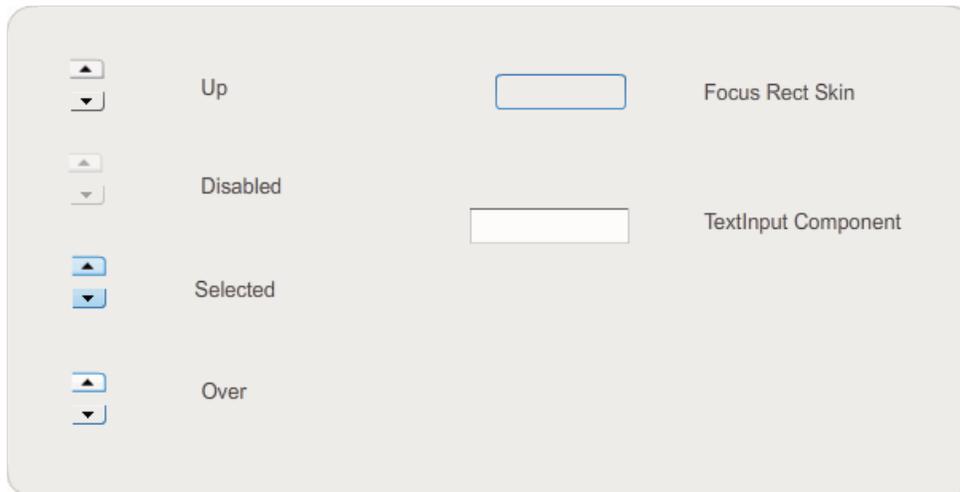
### Aspectos y el componente NumericStepper

El componente NumericStepper tiene aspectos que representan los estados presionado, sin presionar, desactivado y seleccionado de sus botones.

Si un componente NumericStepper está activado, los botones de flecha arriba y flecha abajo muestran el estado correspondiente al desplazamiento del puntero sobre ellos. Los botones muestran su estado presionado cuando se presionan. Cuando se suelta el ratón, recuperan el estado correspondiente al desplazamiento del puntero sobre ellos. Si el puntero se desplaza fuera de los botones estando el ratón presionado, los botones recuperan su estado original.

Si el componente está desactivado, muestra el estado desactivado sea cual sea la acción del usuario.

Un componente NumericStepper tiene los siguientes aspectos:



Aspectos de NumericStepper

- 1 Cree un archivo FLA.
- 2 Arrastre el componente NumericStepper al escenario.
- 3 Establezca el control de zoom en 400% para ampliar la imagen y poder editarla.
- 4 Haga doble clic en el fondo del aspecto TextInput en el panel de aspectos hasta que llegue al nivel Grupo y aparezca el color de fondo en el selector de color de relleno, en el inspector de propiedades.
- 5 Utilice el selector de color de relleno en el inspector de propiedades para seleccionar el color #9999FF y aplicarlo al fondo del aspecto TextInput.
- 6 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 7 Vuelva a hacer doble clic en NumericStepper para abrir de nuevo el panel de aspectos.
- 8 Haga doble clic en el fondo del botón de flecha arriba en el grupo Up hasta que se seleccione el fondo y aparezca su color en el selector de color de relleno, en el inspector de propiedades.
- 9 Seleccione el color #9966FF para aplicarlo al fondo del botón de flecha arriba.
- 10 Repita los pasos 8 y 9 en la flecha abajo del grupo Up.
- 11 Seleccione Control > Probar película.

La instancia de NumericStepper debe aparecer tal y como se muestra en la siguiente ilustración:



## Personalización del componente ProgressBar

El componente ProgressBar puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable de la clase ProgressBar, como `height`, `width`, `scaleX` y `scaleY`.

El componente `ProgressBar` tiene tres aspectos: un aspecto de guía, un aspecto de barra y un aspecto indeterminado. Utiliza la escala de 9 divisiones para ajustar la escala de los activos.

## Estilos y el componente `ProgressBar`

Es posible definir propiedades de estilo para cambiar la apariencia de una instancia de `ProgressBar`. Los estilos de `ProgressBar` especifican los valores de los aspectos y el relleno del componente cuando éste se dibuja. En el siguiente ejemplo se amplía el tamaño de una instancia de `ProgressBar` y se define su estilo `barPadding`.

- 1 Cree un archivo FLA.
- 2 Arrastre un componente `ProgressBar` del panel Componentes al escenario y asígnele el nombre de instancia **myPb**.
- 3 Haga clic en el fotograma 1 de la línea de tiempo principal e introduzca el siguiente código en el panel Acciones:

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

- 4 Seleccione Control > Probar película.

Para obtener información sobre cómo definir estilos de aspecto, consulte [“Aspectos”](#) en la página 105.

## Aspectos y el componente `ProgressBar`

El componente `ProgressBar` utiliza aspectos para representar la guía de la barra de progreso, la barra completada, así como una barra indeterminada, como se muestra en la siguiente ilustración.



*Aspectos de `ProgressBar`*

La barra se coloca sobre el aspecto de guía y se utiliza `barPadding` para determinar la posición. Se ajusta la escala de los activos con la escala de 9 divisiones.

La barra indeterminada se utiliza cuando la propiedad `indeterminate` de la instancia de `ProgressBar` se establece en `true`. El aspecto cambia de tamaño vertical y horizontalmente para ajustarse al tamaño de `ProgressBar`.

Es posible editar estos aspectos para cambiar la apariencia del componente `ProgressBar`. En el siguiente ejemplo, se cambia el color de la barra indeterminada.

- 1 Cree un archivo FLA.
- 2 Arrastre un componente `ProgressBar` al escenario y haga doble clic en él para abrir su panel de iconos de aspecto.
- 3 Haga doble clic en el aspecto de barra indeterminada.
- 4 Establezca el control de zoom en 400% para ampliar el icono y poder editarlo.
- 5 Haga doble clic en una de las barras diagonales y mantenga presionada la tecla Mayús mientras hace clic en cada una de las demás barras. El color actual aparece en el selector de color de relleno en el inspector de propiedades.

- Haga clic en el selector de color de relleno en el inspector de propiedades para abrirlo y seleccionar el color #00CC00 para aplicárselo a las barras diagonales seleccionadas.
- Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- Seleccione Control > Probar película.

El componente ProgressBar debería aparecer como se muestra en la siguiente ilustración.



## Personalización del componente RadioButton

El componente RadioButton puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()`.

El recuadro de delimitación de un componente RadioButton es invisible y designa el área activa del componente. Si aumenta el tamaño del componente, aumenta también el tamaño del área activa.

Si el tamaño de la etiqueta supera el del recuadro de delimitación del componente, la etiqueta se recorta para ajustarse al espacio disponible.

## Uso de estilos con el componente RadioButton

Puede definir propiedades de estilo para cambiar el aspecto de una instancia de RadioButton. Las propiedades de estilo de RadioButton especifican los valores de los aspectos, los iconos, el formato de texto y el relleno del componente cuando éste se dibuja. Los estilos de RadioButton especifican los valores de los aspectos y el relleno del diseño del componente cuando éste se dibuja.

En el ejemplo siguiente se obtiene el estilo `textFormat` de un componente CheckBox y se aplica a un componente RadioButton para que el estilo de sus etiquetas sea idéntico.

- Cree un nuevo documento de Flash (ActionScript 3.0).
- Arrastre un componente CheckBox hasta el escenario e introduzca **myCh** como nombre de instancia en el inspector de propiedades.
- Arrastre un componente RadioButton hasta el escenario e introduzca **myRb** como nombre de instancia en el inspector de propiedades.
- Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo.

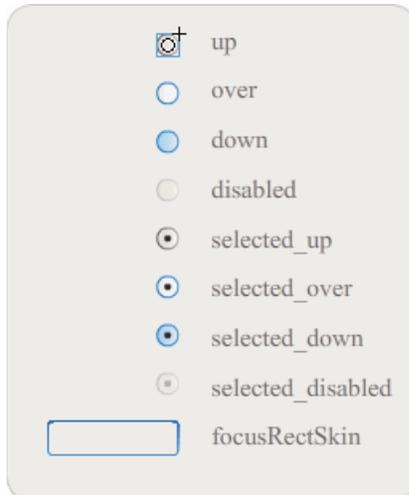
```
var tf:TextFormat = new TextFormat();  
tf.color = 0x00FF00;  
tf.font = "Georgia";  
tf.size = 18;  
myCh.setStyle("textFormat", tf);  
myRb.setStyle("textFormat", myCh.getStyle("textFormat"));
```

Este código define el estilo `textFormat` en el componente CheckBox y, a continuación, lo aplica al componente RadioButton a través de una llamada al método `getStyle()` en el componente CheckBox.

- Seleccione Control > Probar película.

## Aspectos y el componente RadioButton

El componente RadioButton tiene los siguientes aspectos, que pueden editarse para cambiar su apariencia:



Aspectos de RadioButton

Si hay un componente RadioButton activado y no seleccionado, muestra el estado correspondiente al desplazamiento del puntero sobre él. Cuando un usuario hace clic en un RadioButton, recibe la selección de entrada y muestra su aspecto selected\_down. Cuando un usuario suelta el botón del ratón, RadioButton muestra su aspecto selected\_up. Si un usuario mueve el puntero fuera de la zona activa de RadioButton mientras presiona el botón del ratón, RadioButton vuelve a mostrar su aspecto up.

Si RadioButton está desactivado, muestra el estado desactivado sea cual sea la acción del usuario.

En el siguiente ejemplo se reemplaza el aspecto selected\_up que indica el botón está seleccionado.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente RadioButton al escenario y haga doble clic en él para abrir su paleta de aspectos.
- 3 Establezca el control de zoom en 800% para ampliar el icono y poder editarlo.
- 4 Haga doble clic en el aspecto selected\_up para seleccionarlo y presione la tecla Supr para eliminarlo.
- 5 Seleccione la herramienta Rectángulo en el panel Herramientas.
- 6 En el inspector de propiedades, establezca que el color de línea sea rojo (#FF0000) y el color de relleno sea negro (#000000).
- 7 Tomando como punto de partida las cruces que marcan el punto de registro (también denominado *punto de origen* o *punto cero*) del símbolo, haga clic y arrastre el puntero para dibujar un rectángulo.
- 8 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 9 Seleccione Control > Probar película.
- 10 Haga clic en el componente RadioButton para seleccionarlo.

El componente `RadioButton` en su estado seleccionado debería tener un aspecto similar al que se muestra en la siguiente ilustración.



## Personalización del componente `ScrollPane`

El componente `ScrollPane` puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad o método aplicable de la clase `ScrollPane`, como `height`, `width`, `scaleX` y `scaleY`.

El componente `ScrollPane` presenta las siguientes características gráficas:

- El punto de registro (o *punto de origen* o *punto cero*) de su contenido se encuentra en la esquina superior izquierda del panel.
- Cuando se desactiva la barra de desplazamiento horizontal, la barra vertical se muestra de arriba abajo sobre el lado derecho del panel de desplazamiento. Si se desactiva la barra de desplazamiento vertical, la barra horizontal se muestra de izquierda a derecha sobre el borde inferior del panel de desplazamiento. También se pueden desactivar las dos barras de desplazamiento.
- Si el panel de desplazamiento es demasiado pequeño, es posible que el contenido no se muestre correctamente.
- Cuando se cambia el tamaño del panel de desplazamiento, la guía y el cuadro de desplazamiento (deslizador) se expanden o se contraen, y cambia el tamaño de sus áreas activas. Los botones no cambian de tamaño.

### Uso de estilos con el componente `ScrollPane`

Las propiedades de estilo del componente `ScrollPane` especifican los valores de los aspectos y el relleno del diseño del componente cuando éste se dibuja. Los diversos estilos de aspecto permiten especificar distintas clases que se utilizan en los aspectos del componente. Para obtener más información sobre cómo utilizar estilos de aspecto, consulte “Aspectos” en la página 105.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente `ScrollPane` al escenario y asígnele el nombre de instancia **mySp**.
- 3 Haga clic en la ficha Parámetros en el inspector de propiedades e introduzca el siguiente valor en el parámetro `source`: **`http://www.helpexamples.com/flash/images/image1.jpg`**
- 4 Haga clic en el fotograma 1 de la línea de tiempo principal y añada el siguiente código al panel Acciones.

```
mySp.setStyle("contentPadding", 5);
```

Debe tenerse en cuenta que el relleno se aplica entre el borde del componente y su contenido, fuera de las barras de desplazamiento.

- 5 Seleccione Control > Probar película.

## Aspectos y el componente ScrollPane

El componente ScrollPane utiliza un borde y barras de desplazamiento para los activos de desplazamiento. Para más información sobre la aplicación de aspectos en barras de desplazamiento consulte “[Uso de aspectos con el componente UI ScrollBar](#)” en la página 137.

## Personalización del componente Slider

El componente Slider puede transformarse horizontalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable de la clase Slider, como `width` y `scaleX`.

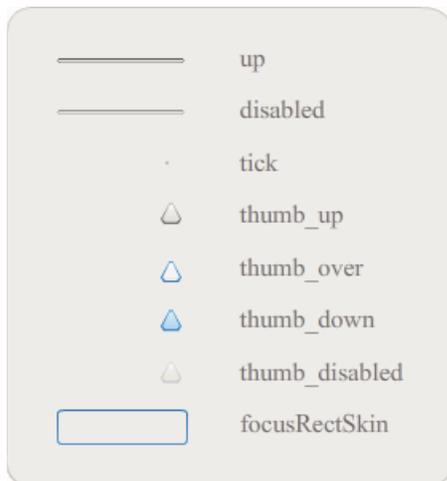
Sólo es posible aumentar la longitud de un deslizador. No se puede aumentar su altura. Flash omite la propiedad `height` y el parámetro `height` del método `setSize()`. Sin embargo, sí es posible crear un deslizador vertical y aumentar su longitud verticalmente.

## Estilos y el componente Slider

Los estilos del componente Slider sólo especifican las clases de sus aspectos y un valor de `FocusRectPadding`, que especifica el número de píxeles que se utilizan como relleno entre el recuadro de delimitación del componente y su límite exterior. Para más información sobre cómo utilizar estilos de aspecto, consulte “[Aspectos](#)” en la página 105.

## Aspectos y el componente Slider

El componente Slider tiene los siguientes aspectos, que pueden editarse para cambiar su apariencia.



*Aspectos de Slider*

En el siguiente ejemplo se edita la guía up para cambiar el color a azul.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente Slider del panel Componentes al escenario.
- 3 Haga doble clic en el componente Slider para abrir su panel de aspectos.

- 4 Haga doble clic en la marca de registro de la guía up para abrirla en el modo de edición de símbolos.
- 5 Establezca el control de zoom en 800% para ampliar el icono y poder editarlo. Tenga en cuenta que la guía de Slider se compone de tres barras.
- 6 Haga clic en la barra superior para seleccionarla. Una vez seleccionada, aparecerá su color en el selector de color de relleno en el inspector de propiedades.
- 7 Utilice el selector de color de relleno en el inspector de propiedades para seleccionar el color #000066 y aplicarlo a la barra superior de la guía de Slider.
- 8 Haga clic en la barra central de la guía de Slider para seleccionarla. Una vez seleccionada, aparecerá su color en el selector de color de relleno en el inspector de propiedades.
- 9 Utilice el selector de color de relleno en el inspector de propiedades para seleccionar el color #0066FF y aplicarlo a la barra central de la guía de Slider.
- 10 Haga clic en la barra inferior de la guía de Slider para seleccionarla. Una vez seleccionada, aparecerá su color en el selector de color de relleno en el inspector de propiedades.
- 11 Utilice el selector de color de relleno en el inspector de propiedades para seleccionar el color #00CCFF y aplicarlo a la barra inferior de la guía de Slider.
- 12 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 13 Seleccione Control > Probar película.

El componente Slider debería aparecer como se muestra en la siguiente ilustración.



## Personalización del componente TextArea

El componente TextArea puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable de la clase TextArea, como `height`, `width`, `scaleX` y `scaleY`.

Cuando se cambia el tamaño de un componente TextArea, el borde se ajusta al nuevo recuadro de delimitación. Si es necesario, las barras de desplazamiento se sitúan en los bordes inferior y derecho. A continuación, se ajusta el tamaño del área de texto al interior del área restante; el componente TextArea carece de elementos con tamaño fijo. Si el componente TextArea es demasiado estrecho para mostrar el tamaño del texto, éste se recorta.

### Estilos y el componente TextArea

Los estilos del componente TextArea especifican los valores de los aspectos, el relleno y el formato de texto del componente cuando éste se dibuja. Los estilos `textFormat` y `disabledTextFormat` definen el estilo del texto que muestra el componente TextArea. Para obtener más información sobre las propiedades de estilo de aspectos, consulte ["Uso de aspectos con el componente TextArea"](#) en la página 132.

En el siguiente ejemplo se define el estilo `disabledTextFormat` para cambiar la apariencia del texto cuando se desactiva el componente `TextArea`; se aplica el mismo proceso para definir el estilo `textFormat` de un componente `TextArea` activado.

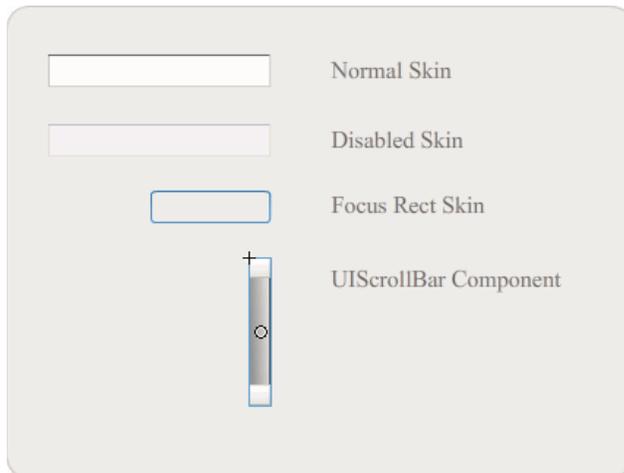
- 1 Cree un archivo de Flash.
- 2 Arrastre un componente `TextArea` al escenario y asígnele el nombre de instancia **myTa**.
- 3 Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo principal.

```
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 Seleccione Control > Probar película.

## Uso de aspectos con el componente `TextArea`

El componente `TextArea` utiliza los siguientes aspectos, que pueden editarse para cambiar su apariencia.



Aspectos de `TextArea`

**Nota:** si se cambia el aspecto `ScrollBar` en un componente, también cambiará en todos los demás componentes que lo utilicen.

El siguiente procedimiento cambia los colores del borde de `Focus Rect Skin`, que aparece cuando se selecciona el componente `TextArea` y el aspecto `Normal`.

- 1 Cree un archivo de Flash.
- 2 Arrastre un componente `TextArea` al escenario y haga doble clic en él para abrir su panel de iconos de aspecto.
- 3 Haga doble clic en `Focus Rect Skin`.
- 4 Haga clic en el borde de `Focus Rect Skin` para seleccionarlo. Una vez seleccionado, aparecerá su color en el selector de color de relleno en el inspector de propiedades.

- 5 Haga clic en el selector de color de relleno en el inspector de propiedades para abrirlo y seleccionar el color #CC0000 para aplicarlo al borde.
- 6 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 7 Haga doble clic en el componente TextArea para abrir su panel de iconos de aspecto.
- 8 Haga doble clic en el aspecto Normal.
- 9 Seleccione cada uno de los extremos del borde del aspecto Normal, de uno en uno, y asígneles el color #990099.
- 10 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- 11 Seleccione Control > Probar película.

Al seleccionar el componente TextArea para comenzar a introducir texto, su borde debería aparecer como se muestra en la siguiente ilustración:



El borde exterior es el aspecto Focus Rect y el borde interior es el borde del aspecto Normal.

Para más información sobre la aplicación de aspectos en UIScrollBar, consulte [“Personalización del componente UIScrollBar”](#) en la página 137.

## Personalización del componente TextInput

Es posible cambiar el tamaño de una instancia de TextInput durante la edición o en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable de la clase TextInput, como `height`, `width`, `scaleX` y `scaleY`.

Cuando se cambia el tamaño de un componente TextInput, el borde se ajusta al nuevo recuadro de delimitación. El componente TextInput no utiliza barras de desplazamiento, si bien el punto de inserción se desplaza automáticamente a medida que el usuario interactúa con el texto. Así, el tamaño del campo de texto se ajusta al interior del área restante; el componente TextInput carece de elementos con tamaño fijo. Si el componente TextInput es demasiado pequeño para mostrar el texto, éste se recorta.

### Estilos y el componente TextInput

Los estilos del componente TextInput especifican los valores de los aspectos, el relleno y el formato de texto del componente cuando éste se dibuja. Los estilos `textFormat` y `disabledTextFormat` definen el estilo del texto que se muestra en el componente. Para obtener más información sobre las propiedades de estilo de aspectos, consulte [“Aspectos y el componente TextInput”](#) en la página 134.

En el siguiente ejemplo se define el estilo `textFormat` para establecer la fuente, el tamaño y el color del texto que se muestra en el componente TextInput. El mismo proceso se aplica para definir el estilo `disabledTextFormat` que se aplica cuando el componente está desactivado.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente TextInput al escenario y asígnele el nombre de instancia **myTi**.

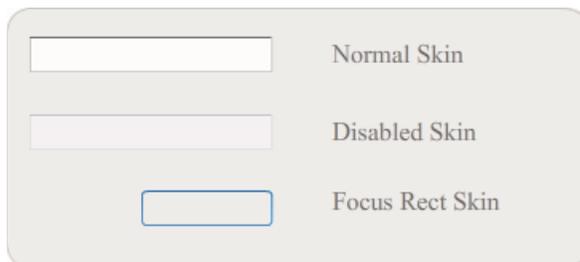
- Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo principal.

```
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);
```

- Selecione Control > Probar película.

## Aspectos y el componente TextInput

El componente TextInput utiliza los siguientes aspectos, que pueden editarse para cambiar su apariencia:



*Texto de TextInput*

El siguiente procedimiento cambia los colores del borde y del fondo de un componente TextInput:

- Cree un archivo de Flash.
- Arrastre un componente TextInput al escenario y haga doble clic en él para abrir su panel de aspectos.
- Haga doble clic en el aspecto Normal.
- Establezca el control de zoom en 800% para ampliar el icono y poder editarlo.
- Selecione cada uno de los extremos del borde del aspecto Normal, de uno en uno, y aplíqueles el color #993399.
- Haga doble clic en el fondo hasta que aparezca su color en el selector de color de relleno en el inspector de propiedades. Seleccione el color #99CCCC para aplicarlo al fondo.
- Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario para volver al modo de edición de documentos.
- Selecione Control > Probar película.

El componente TextInput debería aparecer como se muestra en la siguiente ilustración:



## Personalización del componente TileList

El componente `TileList` puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable, como `width`, `height`, `columnCount`, `rowCount`, `scaleX` y `scaleY`. El componente `ScrollBar`, que contiene `TileList`, realiza un ajuste de escala con el cuadro de lista.

### Estilos y el componente TileList

Los estilos del componente `TileList` especifican los valores de los aspectos, el relleno y el formato de texto del componente cuando éste se dibuja. Los estilos `textFormat` y `disabledTextFormat` definen el estilo del texto que se muestra en el componente. Para más información sobre los estilos de aspectos, consulte ["Uso de aspectos con el componente TileList"](#) en la página 135.

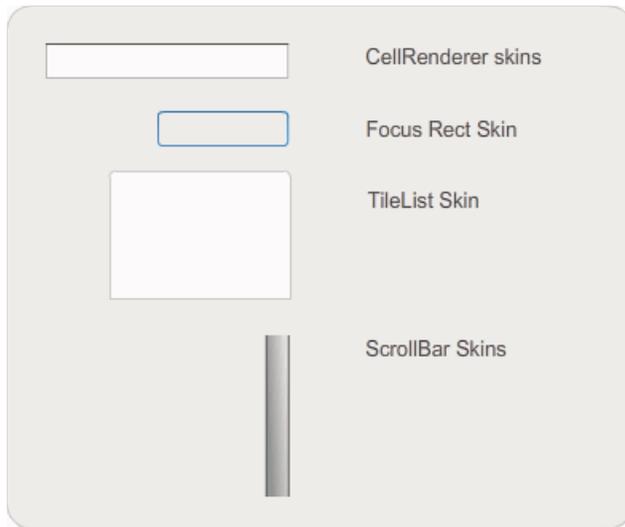
En el siguiente ejemplo se llama al método `setRendererStyle()` con el estilo `textFormat` para definir la fuente, el tamaño, el color y los atributos de texto de las etiquetas que se muestran en una instancia de `TileList`. El mismo proceso se aplica para definir el estilo `disabledTextFormat` que se aplica cuando se establece la propiedad `enabled` en `false`.

- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente `TileList` al escenario y asígnele el nombre de instancia **myTl**.
- 3 Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo.

```
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);
```

### Uso de aspectos con el componente TileList

El componente `TileList` tiene un aspecto `TileList`, un aspecto `CellRenderer` y un aspecto `ScrollBar`. Es posible editar estos aspectos para cambiar la apariencia del componente `TileList`:



Aspectos de TileList

**Nota:** si se cambia el aspecto ScrollBar en un componente, también cambiará en todos los demás componentes que lo utilicen.

El siguiente procedimiento cambia el color del aspecto CellRenderer Selected\_Up de TileList.

- 1 Cree un documento de Flash (ActionScript 3.0).
- 2 Arrastre un componente TileList al escenario y haga doble clic en él para abrir su panel de aspectos.
- 3 Haga doble clic en el aspecto CellRenderer, luego haga doble clic en el aspecto Selected\_Up y finalmente haga clic en el fondo rectangular.
- 4 Con el selector de color de relleno en el inspector de propiedades, seleccione el color #99FFFF para aplicarlo al aspecto Selected\_Up.
- 5 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario hasta volver al modo de edición de documentos.
- 6 En la ficha Parámetros del inspector de propiedades, haga doble clic en la segunda columna de la fila dataProvider para abrir el cuadro de diálogo Valores. Añada elementos con las siguientes etiquetas: 1er elemento, 2º elemento, 3er elemento, 4º elemento.
- 7 Seleccione Control > Probar película.
- 8 Haga clic en una de las celdas de TileList para seleccionarla y mueva el ratón fuera de la celda seleccionada.

La celda seleccionada debería aparecer como se muestra en la siguiente ilustración:



Componente TileList con el color de aspecto Selected\_Up modificado

## Personalización del componente UILoader

El componente UILoader puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable, como `width`, `height`, `scaleX` y `scaleY`.

El comportamiento de ajuste de tamaño del componente UILoader se controla con la propiedad `scaleContent`. Cuando `scaleContent` es `true`, se ajusta la escala del contenido para adaptarse a los límites del componente Loader (y vuelve a ajustar la escala cuando se llama al método `setSize()`). Cuando `scaleContent` es `false`, el tamaño del componente se ajusta al tamaño del contenido y el método `setSize()` y las propiedades de ajuste de tamaño no tiene efecto alguno.

El componente UILoader no tiene elementos de interfaz de usuario a los que se pueden aplicar estilos o aspectos.

## Personalización del componente UIScrollBar

El componente UIScrollBar puede transformarse horizontal y verticalmente durante la edición y en tiempo de ejecución. Sin embargo, un componente UIScrollBar vertical no permite modificar la anchura y un componente UIScrollBar horizontal no permite modificar la altura. Durante la edición, seleccione el componente en el escenario y utilice la herramienta Transformación libre o cualquiera de los comandos Modificar > Transformar. En tiempo de ejecución, utilice el método `setSize()` o cualquier propiedad aplicable de la clase UIScrollBar, como `width`, `height`, `scaleX` y `scaleY`.

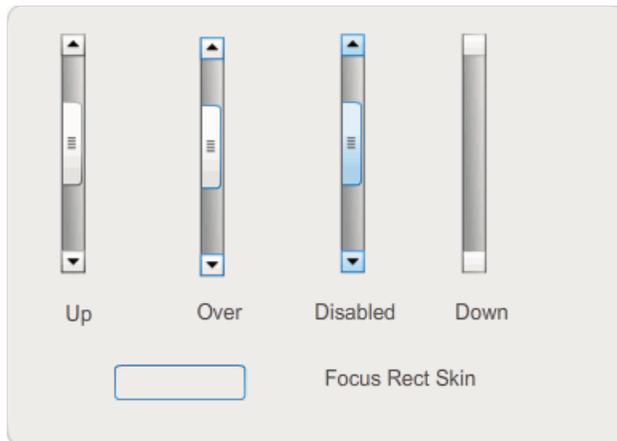
***Nota:** si se utiliza el método `setSize()`, sólo se puede cambiar la anchura de una barra de desplazamiento horizontal o la altura de una barra de desplazamiento vertical. Durante la edición, se puede definir la altura de una barra de desplazamiento horizontal o la anchura de una barra de desplazamiento vertical, pero se restablecerán los valores cuando se publique la película. Sólo es posible cambiar la dimensión de longitud de una barra de desplazamiento.*

### Uso de estilos con el componente UIScrollBar

Los estilos del componente UIScrollBar sólo especifican las clases de sus aspectos y un valor de `FocusRectPadding`, que especifica el número de píxeles que se utilizan como relleno entre el recuadro de delimitación del componente y su límite exterior. Para más información sobre cómo utilizar estilos de aspecto, consulte “Aspectos” en la página 105

### Uso de aspectos con el componente UIScrollBar

El componente UIScrollBar utiliza los siguientes aspectos.



Aspectos de UIScrollBar

Las barras de desplazamiento horizontal y vertical utilizan los mismos aspectos; cuando se muestra una barra de desplazamiento horizontal, el componente UIScrollBar gira los aspectos del modo correspondiente.

**Nota:** si se cambia el aspecto ScrollBar en un componente, también cambiará en todos los demás componentes que lo utilicen.

El siguiente ejemplo muestra cómo se puede cambiar el color del selector y los botones de flecha de UIScrollBar.

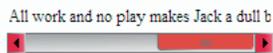
- 1 Cree un nuevo documento de Flash (ActionScript 3.0).
- 2 Arrastre el componente UIScrollBar al escenario y asígnele el nombre de instancia **mySb**. En la ficha Parámetros, establezca la dirección en horizontal.
- 3 Haga doble clic en la barra de desplazamiento para abrir su panel de aspectos.
- 4 Haga clic en el aspecto Up para seleccionarlo.
- 5 Establezca el control de zoom en 400% para ampliar el icono y poder editarlo.
- 6 Haga doble clic en el fondo de la flecha derecha (o la flecha arriba en una barra de desplazamiento vertical) hasta que se seleccione el fondo y aparezca su color en el selector de color de relleno, en el inspector de propiedades.
- 7 Seleccione el color #CC0033 para aplicarlo al fondo del botón.
- 8 Haga clic en el botón Atrás en la parte izquierda de la barra de edición situada encima del escenario hasta volver al modo de edición de documentos.
- 9 Repita los pasos 6, 7 y 8 en los elementos de selector y flecha izquierda (o flecha abajo en una barra de desplazamiento vertical).
- 10 Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo, para asociar la barra de desplazamiento a un componente TextField.

**Personalización de los componentes de la interfaz de usuario**

```
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 Seleccione Control > Probar película.**

El componente UIScrollBar debería aparecer como se muestra en la siguiente ilustración.



*ScrollBar horizontal con selector y flechas izquierda y derecha de color rojo*

# Capítulo 6: Uso del componente FLVPlayback

El componente FLVPlayback permite incluir fácilmente un reproductor de vídeo en la aplicación CS5 Professional para reproducir archivos de vídeo de Adobe Flash Video descargados de forma progresiva a través de HTTP, o reproducir archivos de vídeo sin interrupción, desde Macromedia Flash Media Server de Adobe o desde Flash Video Streaming Service (FVSS).

Con el lanzamiento de Adobe Flash Player 9 Actualización 3 (versión 9.0.115.0 o posterior), se han incorporado mejoras importantes para la reproducción de contenido de vídeo en Flash Player. Esta actualización incluye cambios al componente FLVPlayback que aprovecha el hardware de vídeo del sistema del usuario para proporcionar un mejor rendimiento de reproducción de vídeo. Los cambios al componente FLVPlayback también aumentan la fidelidad de los archivos de vídeo cuando se visualizan en el modo de pantalla completa.

Además, Flash Player 9 Actualización 3 mejora la funcionalidad del componente FLVPlayback añadiendo compatibilidad para los formatos de vídeo MPEG-4 de alta definición que utilizan la codificación H.264 estándar de la industria. Estos formatos incluyen MP4, M4A, MOV, MP4V, 3GP y 3G2.

**Nota:** no se admiten archivos MP4 protegidos, como los que se descargan desde Apple® iTunes® o encriptados digitalmente en FairPlay®.

El componente FLVPlayback, de uso sencillo, tiene las siguientes características y ventajas:

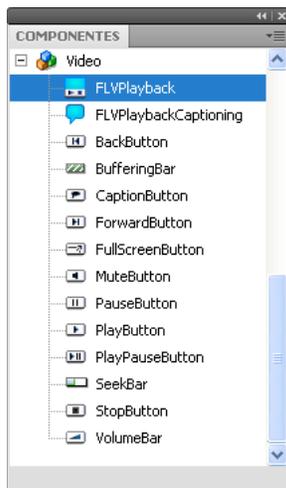
- Puede arrastrarse al escenario e implementarse rápidamente de forma correcta.
- Admite el modo de pantalla completa.
- Proporciona una colección de *aspectos* prediseñados que permiten personalizar la apariencia de los controles de reproducción.
- Permite seleccionar los valores de color y alfa de los aspectos prediseñados.
- Permite a los usuarios avanzados crear sus propios aspectos.
- Proporciona previsualización dinámica durante la edición.
- Proporciona propiedades de diseño para mantener el archivo de vídeo centrado cuando se cambia el tamaño.
- Permite iniciar la reproducción cuando se ha descargado una cantidad suficiente del archivo de vídeo descargado de forma progresiva.
- Proporciona puntos de referencia que permiten sincronizar el vídeo con el texto, los gráficos y la animación.
- Mantiene un tamaño de archivo SWF razonable.

## Uso del componente FLVPlayback

El uso del componente FLVPlayback consiste en colocarlo en el escenario y especificar un archivo de vídeo para que lo reproduzca. También pueden establecerse diversos parámetros que especifiquen su comportamiento y describan el archivo de vídeo.

El componente FLVPlayback también incluye una interfaz de programación de aplicaciones (API) ActionScript. La API incluye las siguientes clases (descritas con detalle en la [Referencia de ActionScript 3.0 para Flash Professional](#)): CuePointType, FLVPlayback, FLVPlaybackCaptioning, NCManager, NCManagerNative, VideoAlign, VideoError, VideoPlayer, VideoState (y varias clases de evento) AutoLayoutEvent, LayoutEvent, MetadataEvent, SkinErrorEvent, SoundEvent, VideoEvent y VideoProgressEvent.

El componente FLVPlayback incluye los componentes de interfaz de usuario personalizados de reproducción FLV. El componente FLVPlayback es una combinación del área de visualización, o reproductor de vídeo, donde se visualiza el archivo de vídeo y los controles que permiten que funcione. Los componentes de interfaz de usuario personalizados de reproducción FLV proporcionan botones de control y mecanismos que puede utilizar para, detener, pausar o controlar de alguna otra forma el archivo de vídeo. Estos controles son: BackButton, BufferingBar, CaptionButton (para FLVPlaybackCaptioning), ForwardButton, FullScreenButton, MuteButton, PauseButton, PlayButton, PlayPauseButton, SeekBar, StopButton y VolumeBar. El componente FLVPlayback y los controles personalizados de interfaz de usuario de reproducción FLV aparecen en el panel Componentes, como se indica en la siguiente ilustración:



Componentes FLVPlayback en el panel Componentes

El proceso de añadir controles de reproducción al componente FLVPlayback se denomina *aplicación de aspectos*. El componente FLVPlayback tiene un aspecto predeterminado inicial, SkinOverAll.swf, que proporciona los controles para reproducir, detener, rebobinar, avanzar y volver atrás, así como controles de barra de búsqueda, silencio, volumen, pantalla completa y subtitulación. Para cambiar este aspecto, puede optar entre:

- Seleccionarlo en una colección de aspectos prediseñados.
- Crear un aspecto personalizado y añadirlo a la colección de aspectos prediseñados
- Seleccionar controles individuales desde los componentes de interfaz de usuario personalizados de reproducción FLV y personalizarlos

Cuando se selecciona un aspecto prediseñado, se pueden elegir por separado los valores de color y alfa del aspecto, durante la edición o en tiempo de ejecución. Para más información, consulte [“Selección de un aspecto prediseñado”](#) en la página 160.

Cuando se selecciona un aspecto distinto, el aspecto seleccionado se convierte en el nuevo aspecto predeterminado.

Para obtener más información sobre la selección o creación de un aspecto para el componente FLVPlayback, consulte [“Personalización del componente FLVPlayback”](#) en la página 159.

## Creación de aplicaciones con el componente FLVPlayback

Se puede incluir el componente FLVPlayback en la aplicación de las maneras siguientes:

- Arrastre el componente FLVPlayback del panel Componentes al escenario y especifique un valor para el parámetro `source`.
- Use el Asistente de importación de vídeo para crear el componente en el escenario y personalizarlo eligiendo un aspecto.
- Utilice el constructor `FLVPlayback()` para crear dinámicamente una instancia de FLVPlayback en el escenario, suponiendo que el componente se encuentre en la biblioteca.

***Nota:** si se crea una instancia de FLVPlayback con ActionScript, también se le debe asignar un aspecto, estableciendo la propiedad `skin` mediante código ActionScript. Cuando se aplica un aspecto de este modo, no se publica automáticamente con el archivo SWF. Se deben copiar al servidor de aplicaciones el archivo SWF de aplicación y el archivo SWF de aspecto ya que, de lo contrario, el archivo SWF de aspecto no estará disponible cuando se ejecute la aplicación.*

### Arrastre el componente FLVPlayback desde el panel Componentes

- 1 En el panel Componentes, haga clic en el signo más (+) para abrir la entrada de vídeo.
- 2 Arrastre el componente FLVPlayback al escenario.
- 3 Con el componente FLVPlayback seleccionado en el escenario, busque la celda Valor del parámetro `source` en la ficha Parámetros del inspector de componentes e introduzca una cadena que especifique una de las opciones siguientes:
  - Una ruta de acceso local a un archivo de vídeo
  - Un URL a un archivo de vídeo
  - Un URL de un archivo SMIL (Synchronized Multimedia Integration Language) que describa la forma de reproducir un archivo de vídeoPara más información sobre cómo crear un archivo SMIL para describir uno o varios archivos FLV, consulte [“Uso de un archivo SMIL”](#) en la página 171.
- 4 En la ficha Parámetros del inspector de componentes, con el componente FLVPlayback seleccionado en el escenario, haga clic en la celda Valor para el parámetro `skin`.
- 5 Haga clic en el icono de lupa para abrir el cuadro de diálogo Seleccionar aspecto.
- 6 Seleccione una de las opciones siguientes:
  - En la lista desplegable Aspecto, seleccione uno de los aspectos prediseñados para asociar un conjunto de controles de reproducción al componente.
  - Si ha creado un aspecto personalizado, seleccione URL de aspecto personalizado en el menú desplegable e introduzca en el cuadro URL el URL del archivo SWF que contiene el aspecto.
  - Seleccione Ninguno y arrastre componentes individuales de interfaz de usuario personalizados de reproducción de FLV al escenario para añadir controles de reproducción.

***Nota:** en los primeros dos casos, aparece una previsualización del aspecto en el panel de visualización situado encima del menú emergente. Se puede utilizar el selector de color para cambiar el color del aspecto.*

Para cambiar el color de un control de interfaz de usuario personalizado, es necesario personalizarlo. Para más información sobre la utilización de controles personalizados de la interfaz de usuario, consulte [“Aplicación de aspectos a componentes individuales de interfaz de usuario personalizados de reproducción FLV”](#) en la página 161.

- 7 Haga clic en Aceptar para cerrar el cuadro de diálogo Seleccionar aspecto.
- 8 Seleccione > Control Probar película para ejecutar el archivo SWF e iniciar el vídeo.

El siguiente procedimiento utiliza el asistente de importación de vídeo para añadir un componente FLVPlayback:

**Uso de Asistente de importación de vídeo:**

- 1 Seleccione Archivo > Importar > Importar vídeo.
- 2 Seleccione una de las siguientes opciones para indicar la ubicación del archivo de vídeo:
  - En mi equipo local
  - Ya se ha implementado en un servidor web, Flash Video Streaming Service o Flash Media Server
- 3 En función de su elección, debe indicar la ruta del archivo o la URL que especifica la ubicación del archivo de vídeo. A continuación, haga clic en Siguiente.
- 4 Si ha seleccionado una ruta de archivo, aparecerá un cuadro de diálogo Implementación, donde puede seleccionar una de las opciones enumeradas para especificar cómo desea implementar el vídeo:
  - Descarga progresiva desde un servidor web estándar
  - Flujo de Flash Video Streaming Service
  - Flujo de Flash Media Server
  - Incorporar vídeo en SWF y reproducir en la línea de tiempo

***Importante:** no elija la opción de incorporar vídeo. El componente FLVPlayback sólo reproduce flujo de vídeo externo. Esta opción no colocará un componente FLVPlayback en el escenario.*

- 5 Haga clic en Siguiente.
- 6 Seleccione una de las opciones siguientes:
  - En la lista desplegable Aspecto, seleccione uno de los aspectos prediseñados para asociar un conjunto de controles de reproducción al componente.
  - Si ha creado un aspecto personalizado para el componente, seleccione URL de aspecto personalizado en el menú emergente e introduzca en el cuadro URL el URL del archivo SWF que contiene el aspecto.
  - Seleccione Ninguno y arrastre componentes individuales de interfaz de usuario personalizados de reproducción de FLV al escenario para añadir controles de reproducción.

***Nota:** en los primeros dos casos, aparece una previsualización del aspecto en el panel de visualización situado encima del menú emergente.*

- 7 Haga clic en Aceptar para cerrar el cuadro de diálogo Seleccionar aspecto.
- 8 Lea el mensaje del cuadro de diálogo Finalizar importación de vídeo para ver qué sucederá a continuación y después haga clic en Finalizar.
- 9 Si no ha guardado el archivo FLA, aparecerá un cuadro de diálogo Guardar como.
- 10 Seleccione > Control Probar película para ejecutar el archivo SWF e iniciar el vídeo.

El siguiente procedimiento añade el componente FLVPlayback mediante código ActionScript.

**Creación dinámica de una instancia mediante código ActionScript:**

- 1 Arrastre el componente FLVPlayback del panel Componentes al panel Biblioteca (Ventana > Biblioteca).

- 2 Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo. Cambie *install\_drive* por la unidad en la que instaló Flash y modifique la ruta de acceso para reflejar la ubicación de la carpeta Skins en la instalación:

En un equipo con Windows:

```
import fl.video.*;
var my_FLVPlybk = new FLVPlayback();
my_FLVPlybk.x = 100;
my_FLVPlybk.y = 100;
addChild(my_FLVPlybk);
my_FLVPlybk.skin = "file:///install_drive|/Program Files/Adobe/Adobe Flash
CS5/en/Configuration/FLVPlayback Skins/ActionScript 3.0/SkinOverPlaySeekMute.swf"
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/water.flv";
```

En un equipo con 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:** si no se establecen las propiedades *source* y *skin*, el clip de película generado aparece vacío.

- 3 Seleccione Control > Probar película para ejecutar el archivo SWF e iniciar el archivo de vídeo.

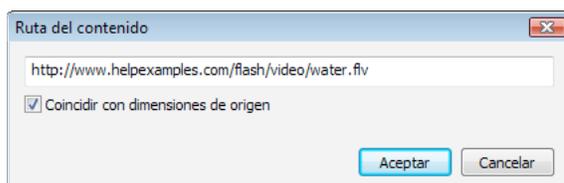
## Parámetros del componente FLVPlayback

Para cada instancia del componente FLVPlayback, se pueden establecer los siguientes parámetros en el inspector de componentes o en el inspector de propiedades: *align*, *autoPlay*, *cuePoints*, *preview*, *scaleMode*, *skin*, *skinAutoHide*, *skinBackgroundAlpha*, *skinBackgroundColor*, *source* y *volume*. Cada uno de estos parámetros tiene una propiedad de ActionScript del mismo nombre. Al asignar un valor a estos parámetros, se establecerá el estado inicial de la propiedad en la aplicación. Si se establece la propiedad en ActionScript, se sustituye el valor establecido en el parámetro. Para obtener información sobre los posibles valores para estos parámetros, consulte la clase FLVPlayback en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Especificación del parámetro source de FLVPlayback

El parámetro *source* permite especificar el nombre y la ubicación del archivo de vídeo, que informan a Flash sobre el modo de reproducir el archivo.

Abra el cuadro de diálogo Ruta del contenido haciendo doble clic en la celda Valor del parámetro *source* en el inspector de componentes.



Cuadro de diálogo Ruta del contenido de FLVPlayback

El cuadro de diálogo Ruta del contenido proporciona una casilla de verificación, Coincidir con dimensiones de FLV de origen, que especifica si la instancia de FLVPlayback en el escenario debería coincidir con las dimensiones del archivo de vídeo de origen. El archivo vídeo de origen contiene dimensiones de altura y anchura preferidas para la reproducción. Si selecciona esta opción, las dimensiones de la instancia de FLVPlayback cambian de tamaño para que coincidan con las dimensiones preferidas.

### El origen

Introduzca el URL o la ruta de acceso local del archivo de vídeo o de un archivo XML que describa la forma de reproducir el archivo de vídeo. Si no conoce la ubicación exacta del archivo de vídeo, haga clic en el icono de carpeta para abrir un cuadro de diálogo Navegador que le permitirá buscar la ubicación correcta. Al buscar un archivo de vídeo, si está en la ubicación del archivo SWF de destino (o en una subcarpeta), Flash utiliza automáticamente la ruta de acceso relativa a esa ubicación, por lo que puede utilizarlo desde un servidor web. De lo contrario será una ruta completa de Windows o Macintosh. Para introducir el nombre de un archivo XML local, escriba la ruta de acceso y el nombre.

Si especifica un URL HTTP, el archivo de vídeo se reproduce como un archivo de descarga progresiva. Si se especifica un URL que sea RTMP, el archivo de vídeo se transmite desde Flash Media Server o FVSS. Un URL a un archivo XML también podría ser un archivo de vídeo transmitido sin interrupción desde Flash Media Server o FVSS.

### Importante:

También se puede especificar la ubicación de un archivo SMIL que describa la forma de reproducir varios flujos de archivos de vídeo para múltiples anchos de banda. El archivo utiliza el lenguaje SMIL (Synchronized Multimedia Integration Language) para describir los archivos FLV. Para obtener una descripción del archivo SMIL, consulte “[Uso de un archivo SMIL](#)” en la página 171.

También se puede especificar el nombre y la ubicación del archivo de vídeo utilizando `FLVPlayback` de ActionScript. La propiedad `source` y los métodos `FLVPlayback.play()` y `FLVPlayback.load()`. Estas tres alternativas tienen prioridad sobre el parámetro `source` en el inspector de componentes. Para obtener más información, consulte las entradas `FLVPlayback.source`, `FLVPlayback.play()` y `FLVPlayback.load()` de la clase `FLVPlayback` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Compatibilidad con el modo de pantalla completa

La versión ActionScript 3.0 del componente `FLVPlayback` admite el modo de pantalla completa, que requiere Flash Player 9.0.28.0, o posterior y también que el HTML esté configurado correctamente para la visualización de pantalla completa. Algunos aspectos prediseñados incluyen un conmutador para activar y desactivar el modo de pantalla completa. El icono `FullScreenButton` aparece en la parte derecha de la barra de control en la siguiente ilustración.



Icono de pantalla completa en la barra de control

La compatibilidad con la pantalla completa sólo tiene lugar si se establece la propiedad `fullScreenTakeOver` en `true` (el valor predeterminado).

La compatibilidad con la pantalla completa se puede producir con o sin la compatibilidad de aceleración de hardware. Para información sobre la compatibilidad de aceleración de hardware, consulte “[Aceleración de hardware](#)” en la página 148.

### Para implementar la compatibilidad con la pantalla completa para FLVPlayback:

- 1 Añada el componente `FLVPlayback` a la aplicación y asigne un archivo de vídeo al mismo.

- 2 Seleccione un aspecto para el componente FLVPlayback que tiene el botón de pantalla completa (Ej: SkinUnderPlaySeekFullscreen.swf) o añada el componente de interfaz de usuario FullScreenButton al componente FLVPlayback de la sección Vídeo en el panel Componentes.
- 3 Seleccione Archivo > Configuración de publicación.
- 4 En el cuadro de diálogo Configuración de publicación, haga clic en la ficha HTML.
- 5 En la ficha HTML, seleccione Flash con compatibilidad de pantalla completa del menú emergente Plantilla.
- 6 Asimismo en la ficha HTML, seleccione la casilla de verificación Detectar versión de Flash y especifique una versión 9.0.28 o posterior, según la versión de Flash Player que esté utilizando.
- 7 Seleccione la ficha Formatos y asegúrese de que las opciones Flash (.swf) y HTML (.html) están seleccionadas. Puede reemplazar los nombres de archivo predeterminados.
- 8 Haga clic en Publicar y luego en Aceptar.

Como una alternativa al paso 7, se puede hacer clic en Aceptar y seleccionar Archivo > Publicar previsualización > Predeterminada - (HTML) para abrir automáticamente el archivo HTML exportado en su navegador predeterminado. De lo contrario, abra el archivo HTML exportado con el navegador para probar la opción de pantalla completa.

Para añadir el componente FLVPlayback con compatibilidad de pantalla completa a la página web, abra el archivo HTML exportado y copie el código que incorpora el archivo SWF en el archivo HTML para la página web. Este código debería ser similar al siguiente ejemplo:

```
//from the <head> section

<script language="javascript"> AC_FL_RunContent = 0; </script>
<script language="javascript"> DetectFlashVer = 0; </script>
<script src="AC_RunActiveContent.js" language="javascript"></script>
<script language="JavaScript" type="text/javascript">
<!--
// -----
// Globals
// Major version of Flash required
var requiredMajorVersion = 9;
// Minor version of Flash required
var requiredMinorVersion = 0;
// Revision of Flash required
var requiredRevision = 28;
// -----
// -->
</script>

//and from the <body> section

<script language="JavaScript" type="text/javascript">
<!--
if (AC_FL_RunContent == 0 || DetectFlashVer == 0) {
    alert("This page requires AC_RunActiveContent.js.");
} else {
    var hasRightVersion = DetectFlashVer(requiredMajorVersion,
        requiredMinorVersion, requiredRevision);
    if(hasRightVersion) { // if we've detected an acceptable version
        // embed the Flash movie
        AC_FL_RunContent(
            &apos;codebase&apos;;, &apos;http://download.macromedia.com/pub/
```

```
        shockwave/cabs/flash/swflash.cab#version=9,0,28,0&apos;;
        &apos;width&apos;;, &apos;550&apos;;,
        &apos;height&apos;;, &apos;400&apos;;,
        &apos;src&apos;;, &apos;fullscreen&apos;;,
        &apos;quality&apos;;, &apos;high&apos;;,
        &apos;pluginspage&apos;;, &apos;http://www.macromedia.com/go/
        getflashplayer&apos;;,
        &apos;align&apos;;, &apos;middle&apos;;,
        &apos;play&apos;;, &apos>true&apos;;,
        &apos;loop&apos;;, &apos>true&apos;;,
        &apos;scale&apos;;, &apos;showall&apos;;,
        &apos;wmode&apos;;, &apos;window&apos;;,
        &apos;devicefont&apos;;, &apos>false&apos;;,
        &apos;id&apos;;, &apos;fullscreen&apos;;,
        &apos;bgcolor&apos;;, &apos;#ffffff&apos;;,
        &apos;name&apos;;, &apos;fullscreen&apos;;,
        &apos;menu&apos;;, &apos>true&apos;;,
        &apos;allowScriptAccess&apos;;, &apos;sameDomain&apos;;,
        &apos;allowFullScreen&apos;;, &apos>true&apos;;,
        &apos;movie&apos;;, &apos;fullscreen&apos;;,
        &apos;salign&apos;;, &apos;&apos;; ); //end AC code
    } else { // Flash is too old or we can&apos;t detect the plug-in.
        var alternateContent = &apos;Alternative HTML content should be placed
            here.&apos;;
            + &apos;This content requires Adobe Flash Player.&apos;;
            + &apos;<a href=http://www.macromedia.com/go/getflash/>Get Flash</a>
                &apos;;;
        document.write(alternateContent); // Insert non-Flash content.
    }
}
// -->
</script>
<noscript>
    // Provide alternative content for browsers that do not support scripting
    // or for those that have scripting disabled.
    Alternative HTML content should be placed here. This content requires Adobe Flash Player.
    <a href="http://www.macromedia.com/go/getflash/">Get Flash</a>
</noscript>
```

Como una alternativa, se puede utilizar el archivo HTML exportado como la plantilla para la página web y añadir el resto del contenido. Si se procede de este modo, se debe cambiar el nombre del archivo HTML para que el archivo no se sobrescriba accidentalmente en caso de que más tarde se exporte el archivo HTML FLVPlayback nuevamente de Flash.

En cualquier caso, también se debe cargar en el servidor web el archivo AC\_RunActiveContent.js que se exporta a la misma carpeta que el archivo HTML

La compatibilidad de ActionScript para el modo de pantalla completa incluye las propiedades `fullScreenBackgroundColor`, `fullScreenSkinDelay`, y `fullScreenTakeOver` y el método `enterFullScreenDisplayState()`. Para obtener información sobre estos elementos de ActionScript, consulte la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Uso de `enterFullScreenDisplayState()`

También se puede invocar al modo de pantalla completa llamando al método `enterFullScreenDisplayState()` ActionScript, como se muestra en el siguiente ejemplo.

**Uso del componente FLVPlayback**

```
function handleClick(e:MouseEvent):void {
    myFLVPlybk.enterFullScreenDisplayState();
}
myButton.addEventListener(MouseEvent.CLICK, handleClick);
```

En este ejemplo, el modo de pantalla completa *no* se invoca haciendo clic en el conmutador de pantalla completa en un aspecto de FLVPlayback, sino haciendo clic en un botón (MyButton) que el creador de la página web ha incluido para invocar e modo de pantalla completa. Haciendo clic en le botón se activa el controlador de eventos handleClick, que llama al método `enterFullScreenDisplayState()`.

El método `enterFullScreenDisplayState()` establece la propiedad `Stage.displayState` en `StageDisplayState.FULL_SCREEN` y en consecuencia tiene las mismas restricciones que la propiedad `displayState`. Para más información sobre el método `enterFullScreenDisplayState()` y la propiedad `Stage.displayState`, consulte la [Referencia de ActionScript 3.0 para Flash Professional](#).

**Aceleración de hardware**

Flash Player 9.0.115.0 y las versiones posteriores incluyen código que saca provecho del hardware de vídeo disponible para mejorar el rendimiento y la fidelidad de los archivos FLV que FLVPlayback reproduce en el modo de pantalla completa. Si se cumplen los requisitos previos y se establece la propiedad `fullScreenTakeOver` en `true`, Flash Player utiliza la aceleración de hardware para escalar el archivo de vídeo, en lugar de hacerlo a través del software. Si el componente FLVPlayback se ejecuta en una versión anterior de Flash Player, o si los requisitos previos para la aceleración de hardware no existen, Flash Player escala el archivo de vídeo, de la misma manera como lo hizo anteriormente.

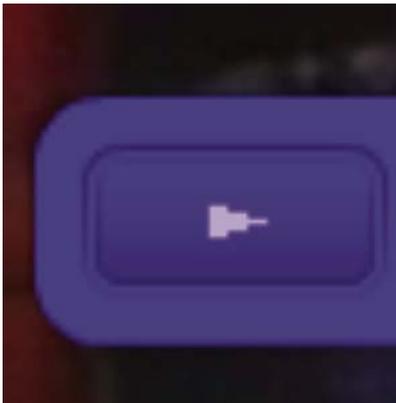
Para sacar partido de la aceleración de hardware para la compatibilidad de pantalla completa, el PC debe tener una tarjeta de vídeo compatible con DirectX 7 con 4 MB o más de VRAM (RAM de vídeo) Esta compatibilidad de hardware está disponible en Windows 2000 o Mac OS X 10.2 y en versiones posteriores de dichos sistemas operativos. DirectX® proporciona API que incluyen una interfaz entre el software y el hardware de vídeo para acelerar gráficos tridimensionales y bidimensionales, entre otras características.

Para sacar provecho del modo de aceleración de hardware, también se debe invocar el modo de pantalla completa de una de las siguientes maneras:

- Utilizando el conmutador de pantalla completa en un aspecto de FLVPlayback
- Utilizando el control de vídeo `FullScreenButton`
- Utilizando el método `enterFullScreenDisplayState()` de ActionScript. Para más información, consulte “[Uso de enterFullScreenDisplayState\(\)](#)” en la página 147.

Si se invoca el modo de pantalla completa estableciendo la propiedad `Stage.displayState` en `StageDisplayState.FULLSCREEN`, FLVPlayback no utiliza la aceleración de hardware, aun si se cuenta con memoria y hardware de vídeo.

Una consecuencia de utilizar la aceleración de hardware para la compatibilidad de pantalla completa es que los aspectos de FLVPlayback se escalan junto con el reproductor de vídeo y el archivo de vídeo. La siguiente imagen muestra el efecto del modo de pantalla completa con la aceleración de hardware en el aspecto de FLVPlayback y se muestra un detalle en resolución completa.



*Modo de pantalla completa en un monitor de 1600 x 1200 con un video de 320x240 píxeles*

Esta imagen muestra la utilización del modo de pantalla completa en un monitor de 1600 x 1200 con un archivo de vídeo que tiene una anchura de 320 y una altura de 240, que son las dimensiones predeterminadas de FLVPlayback. El efecto de distorsión en el aspecto aparece más pronunciado en los archivos FLV con dimensiones más pequeñas o en un monitor más grande. Contrariamente, el efecto de distorsión es menos pronunciado en archivos FLV más grandes o en monitores más pequeños. Por ejemplo, si se cambia de 640 x 480 a 1600 x 1200 se aumenta el tamaño del aspecto pero aparece menos distorsionado.

Se puede establecer la propiedad `skinScaleMaximum` para limitar el escalado del aspecto de FLVPlayback. El valor predeterminado es 4.0 o 400 %. Sin embargo, la limitación del escalado del aspecto requiere una combinación de hardware y software para escalar el FLV y esto puede afectar negativamente el rendimiento en FLV que tienen grandes dimensiones codificados a una velocidad alta de transferencia. Si el vídeo es grande (por ejemplo, 640 píxeles de ancho o más, 480 píxeles de alto o más), no debe establecer `skinScaleMaximum` con un valor pequeño, ya que podrían producirse problemas de rendimiento evidentes en monitores de gran tamaño. La propiedad `skinScaleMaximum` permite controlar las compensaciones entre el rendimiento y la calidad y la apariencia de un aspecto grande.

### Salir del modo de pantalla completa

Para salir del modo de pantalla completa, haga clic otra vez en el botón de pantalla completa o pulse la tecla Esc.

Si se establecen las siguientes propiedades y se llaman a los siguientes métodos se pueden producir cambios en el diseño provocando que el componente FLVPlayback abandone el modo de pantalla completa: `height`, `registrationHeight`, `registrationWidth`, `registrationX`, `registrationY`, `scaleX`, `scaleY`, `width`, `x`, `y`, `setScale()` o `setSize()`.

Si se establece la propiedad `align` o `scaleMode`, FLVPlayback las establece en `center` y `maintainAspectRatio` hasta que se salga del modo de pantalla completa.

Si se cambia el valor de la propiedad `fullScreenTakeOver` de `true` a `false` cuando se está utilizando la pantalla completa, el modo de aceleración de hardware también hace que Flash abandone el modo de pantalla completa.

## Alineación del diseño para reproducir varios archivos de vídeo

El componente FLVPlayback de ActionScript 3.0 tiene una propiedad `align` que especifica si el archivo de vídeo debe centrarse cuando cambia de tamaño, o ubicarse arriba, debajo, a la derecha o a la izquierda del componente. Además de las propiedades `x`, `y`, `width` y `height`, el componente ActionScript 3.0 también tiene propiedades `registrationX`, `registrationY`, `registrationWidth` y `registrationHeight`. Inicialmente, coinciden con las propiedades `x`, `y`, `width` y `height`. Cuando se cargan archivos de vídeo posteriores, el rediseño automático no los cambia, de modo que los nuevos archivos de vídeo pueden centrarse en el mismo lugar. Si `scaleMode = VideoScaleMode.MAINTAIN_ASPECT_RATIO`, los archivos FLV se ajustarán a las dimensiones originales del componente, en lugar de modificar la altura y anchura del mismo.

## Reproducción automática de archivos de vídeo descargados de forma progresiva

Cuando se carga un archivo de vídeo descargado de forma progresiva, FLVPlayback inicia la reproducción del archivo de vídeo únicamente cuando se ha descargado una cantidad suficiente de archivo para poder reproducir el archivo de vídeo de principio a fin.

Si se desea reproducir el archivo de vídeo antes de que se haya descargado una cantidad suficiente, es necesario llamar al método `play()` sin ningún parámetro.

Si se desea volver al estado en que se espera a que haya una cantidad suficiente de archivo de vídeo para descargar, se debe llamar al método `pause()` y, a continuación, al método `playWhenEnoughDownloaded()`.

## Uso de puntos de referencia

Un punto de referencia es un punto en el que el reproductor de vídeo distribuye un evento `cuePoint` mientras se reproduce un archivo de vídeo. Se pueden añadir puntos de referencia a un archivo FLV cuando se desee interactuar con otro elemento de la página Web. Quizá se desee mostrar texto o un gráfico, por ejemplo, o sincronizar con una animación de Flash, o pausar la reproducción del archivo FLV, buscar otro punto distinto del vídeo o cambiar a otro archivo FLV. Los puntos de referencia permiten recibir el control en el código ActionScript para sincronizar dichos puntos del archivo FLV con otras acciones de la página Web.

Hay tres tipos de puntos de referencia: navegación, eventos y ActionScript. Los puntos de referencia de navegación y eventos se denominan también puntos de referencia *incorporados* porque se incorporan en el flujo de archivos FLV y en el paquete de metadatos del archivo FLV.

Un *punto de referencia de navegación* permite buscar un determinado fotograma en el archivo FLV, ya que crea un fotograma clave en el archivo FLV, lo más cerca posible al tiempo especificado. Un *fotograma clave* es un segmento de datos que se produce entre los fotogramas de imagen del flujo del archivo FLV. Cuando se busca un punto de referencia de navegación, el componente busca el fotograma clave e inicia el evento `cuePoint`.

Un *punto de referencia de evento* permite sincronizar un instante específico del archivo FLV con un evento externo de la página Web. El evento `cuePoint` se produce precisamente en el instante especificado. Se pueden incorporar puntos de referencia de navegación y de evento en un archivo FLV mediante el asistente de importación de vídeo o Flash Video Encoder. Para más información sobre el asistente de importación de vídeo y Flash Video Encoder, consulte el Capítulo 16, "Trabajo con vídeo", en *Utilización de Flash*.

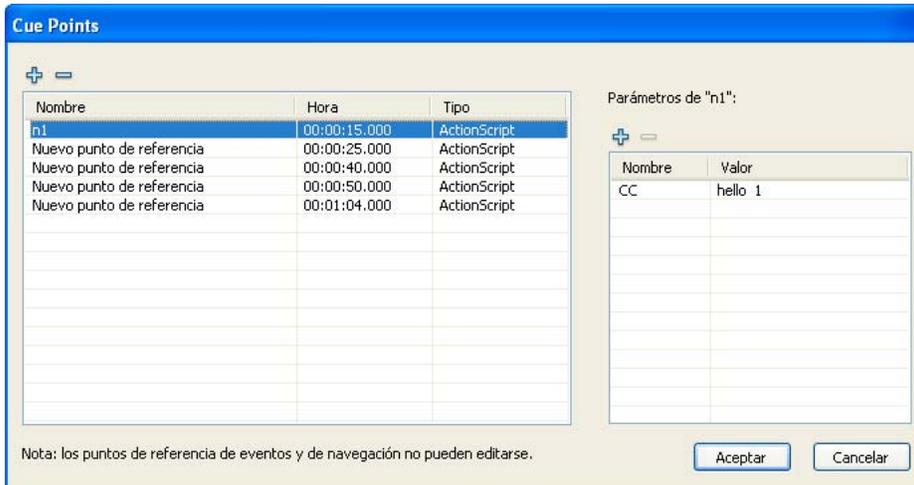
Un *punto de referencia de ActionScript* es un punto de referencia externo que puede añadirse a través del cuadro de diálogo Puntos de referencia de Flash Video del componente o a través del método `FLVPlayback.addASCuePoint()`. El componente almacena y rastrea los puntos de referencia de ActionScript independientemente del archivo FLV, por lo que son menos precisos que los puntos de referencia incorporados. La precisión de los puntos de referencia de ActionScript es de una décima de segundo. Para aumentar esta precisión, es necesario reducir el valor de la propiedad `playheadUpdateInterval`, ya que el componente genera el evento `cuePoint` para los puntos de referencia de ActionScript cuando se actualiza la cabeza lectora. Para obtener más información, consulte la propiedad `FLVPlayback.playheadUpdateInterval` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

En ActionScript y en los metadatos del archivo FLV, un punto de referencia se representa como un objeto con las siguientes propiedades: `name`, `time`, `type` y `parameters`. La propiedad `name` es una cadena que contiene el nombre asignado del punto de referencia. La propiedad `time` es un número que representa el tiempo en horas, minutos, segundos y milisegundos (HH:MM:SS.mmm) cuando se produce el punto de referencia. La propiedad `type` es una cadena cuyo valor es "navigation", "event" o "actionscript", en función del tipo de punto de referencia que se haya creado. La propiedad `parameters` es un conjunto de los pares nombre-valor especificados.

Cuando se produce un evento `cuePoint`, el punto de referencia está disponible en el objeto de evento a través de la propiedad `info`.

### Uso del cuadro de diálogo Puntos de referencia de Flash Video

Para abrir el cuadro de diálogo Puntos de referencia de Flash Video, haga doble clic en la celda Valor del parámetro `cuePoints` del inspector de componentes. El cuadro de diálogo tiene el siguiente aspecto:



Cuadro de diálogo Cue Points (Puntos de referencia)

El cuadro de diálogo muestra los puntos de referencia de ActionScript y los incorporados. Se puede utilizar este cuadro de diálogo para añadir y eliminar puntos de referencia de ActionScript y parámetros de puntos de referencia. También se pueden activar o desactivar los puntos de referencia incorporados. Sin embargo, no puede añadir, cambiar ni eliminar cuepoints incorporados.

#### Añadir un punto de referencia de ActionScript:

- 1 Haga doble clic en la celda Valor del parámetro `cuePoints` en el inspector de componentes para abrir el cuadro de diálogo Puntos de referencia de Flash Video.
- 2 Haga clic en el signo más (+) situado en la esquina superior izquierda, por encima de la lista de puntos de referencia, para añadir una entrada de punto de referencia de ActionScript predeterminada.

- 3 Haga clic en el texto Nuevo cuepoint de la columna Nombre y edite el texto para asignar un nombre al cuepoint.
- 4 Haga clic en el valor de tiempo de 00:00:00:000 para editarlo y asigne un tiempo para el punto de referencia. Se puede especificar el tiempo en horas, minutos, segundos y milisegundos (HH:MM:SS.mmm).  
Si hay varios puntos de referencia, el cuadro de diálogo mueve el nuevo punto de referencia a su posición cronológica en la lista.
- 5 Para añadir un parámetro para el punto de referencia seleccionado, haga clic en el signo más (+) situado encima de la sección Parámetros e introduzca valores en las columnas Nombre y Valor. Repita este paso para cada parámetro.
- 6 Para añadir más puntos de referencia de ActionScript, repita los pasos 2 a 5 para cada uno.
- 7 Haga clic en Aceptar para guardar los cambios.

#### Eliminar un punto de referencia de ActionScript:

- 1 Haga doble clic en la celda Valor del parámetro `cuePoints` en el inspector de componentes para abrir el cuadro de diálogo Puntos de referencia de Flash Video.
- 2 Seleccione los puntos de referencia que desea eliminar.
- 3 Haga clic en el signo menos (-) situado en la esquina superior izquierda, por encima de la lista de puntos de referencia, para eliminarlo.
- 4 Repita los pasos 2 y 3 para cada punto de referencia que desee eliminar.
- 5 Haga clic en Aceptar para guardar los cambios.

#### Para activar o desactivar un punto de referencia de archivo FLV incorporado:

- 1 Haga doble clic en la celda Valor del parámetro `cuePoints` en el inspector de componentes para abrir el cuadro de diálogo Puntos de referencia de Flash Video.
- 2 Seleccione el punto de referencia que desea activar o desactivar.
- 3 Haga clic en el valor de la columna Tipo para activar el menú emergente o haga clic en la flecha abajo.
- 4 Haga clic en el nombre del tipo de punto de referencia (por ejemplo, Event o Navigation) para activarlo. Haga clic en Disabled para desactivarlo.
- 5 Haga clic en Aceptar para guardar los cambios.

#### Uso de los puntos de referencia con ActionScript

Se puede utilizar ActionScript para añadir puntos de referencia de ActionScript, detectar eventos `cuePoint`, buscar puntos de referencia de cualquier tipo o de un tipo especificado, buscar un punto de referencia de navegación, activar o desactivar un punto de referencia, comprobar si un punto de referencia está activado y quitar un punto de referencia.

En los ejemplos de esta sección, se utiliza un archivo FLV denominado `cuepoints.flv`, que contiene los tres puntos de referencia siguientes:

Nombre	Tiempo	Tipo
point1	00:00:00.418	Navigation
point2	00:00:07.748	Navigation
point3	00:00:16.020	Navigation

### Añadir puntos de referencia de ActionScript

Es posible añadir puntos de referencia de ActionScript a un archivo FLV mediante el método `addASCuePoint()`. En el siguiente ejemplo, se añaden dos puntos de referencia de ActionScript al archivo FLV cuando está listo para se. Se añade el primer punto de referencia mediante un objeto de punto de referencia, que especifica el tiempo, el nombre y el tipo del punto de referencia en sus propiedades. En la segunda llamada, se especifica el tiempo y el nombre mediante los parámetros `time` y `name` del método.

```
// Requires an FLVPlayback instance called my_FLVPlybk on Stage
import fl.video.*;
import fl.video.MetadataEvent;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var cuePt:Object = new Object(); //create cue point object
cuePt.time = 2.02;
cuePt.name = "ASpt1";
cuePt.type = "actionscript";
my_FLVPlybk.addASCuePoint(cuePt); //add AS cue point
// add 2nd AS cue point using time and name parameters
my_FLVPlybk.addASCuePoint(5, "ASpt2");
```

Para obtener más información, consulte el método `FLVPlayback.addASCuePoint()` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

### Detección de eventos cuePoint

El evento `cuePoint` permite recibir el control en el código ActionScript cuando se produce un evento `cuePoint`. Cuando se producen puntos de referencia en el siguiente ejemplo, el detector de `cuePoint` llama a una función de controlador de eventos que muestra el valor de la propiedad `playheadTime` y el nombre y el tipo del punto de referencia. Utilice este ejemplo junto con el ejemplo en la sección anterior, Añadir puntos de referencia de ActionScript, para ver los 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 obtener más información sobre el evento `cuePoint`, consulte el evento `FLVPlayback.cuePoint` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

### Localización de puntos de referencia

El código ActionScript permite encontrar un punto de referencia de cualquier tipo, el punto de referencia más cercano en el tiempo o el siguiente punto de referencia con un nombre específico.

El controlador de eventos `ready_listener()` del siguiente ejemplo llama al método `findCuePoint()` para buscar el punto de referencia `ASpt1` y, a continuación, llama al método `findNearestCuePoint()` para buscar el punto de referencia de navegación más cercano en el tiempo al punto de referencia `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);
}
```

En el siguiente ejemplo, el controlador de eventos `ready_listener()` busca el punto de referencia `ASpt` y llama al método `findNextCuePointWithName()` para buscar el siguiente punto de referencia con el mismo nombre:

```
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 obtener más información sobre cómo encontrar puntos de referencia, consulte los métodos `FLVPlayback.findCuePoint()`, `FLVPlayback.findNearestCuePoint()` y `FLVPlayback.findNextCuePointWithName()` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

### Búsqueda de puntos de referencia de navegación

Se puede buscar un punto de referencia de navegación y buscar el anterior y siguiente punto de referencia de navegación a partir de un tiempo especificado. En el siguiente ejemplo se reproduce el archivo FLV `cuepoints.flv` y se busca el punto de referencia correspondiente a 7.748 cuando se produce el evento `ready`. Cuando se produce el evento `cuePoint`, el ejemplo llama al método `seekToPrevNavCuePoint()` para buscar el primer punto de referencia. Cuando se produce dicho evento `cuePoint`, el ejemplo llama al método `seekToNextNavCuePoint()` para buscar el último punto de referencia añadiendo 10 segundos a `eventObject.info.time`, que es el tiempo del punto de referencia actual.

**Uso del componente FLVPlayback**

```
import fl.video.*;

my_FLVPlybk.addEventListener(VideoEvent.READY, ready_listener);
function ready_listener(eventObject:Object):void {
    my_FLVPlybk.seekToNavCuePoint("point2");
}
my_FLVPlybk.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace(eventObject.info.time);
    if(eventObject.info.time == 7.748)
        my_FLVPlybk.seekToPrevNavCuePoint(eventObject.info.time - .005);
    else
        my_FLVPlybk.seekToNextNavCuePoint(eventObject.info.time + 10);
}
my_FLVPlybk.source = "http://helpexamples.com/flash/video/cuepoints.flv";
```

Para obtener más información, consulte los métodos `FLVPlayback.seekToNavCuePoint()`, `FLVPlayback.seekToNextNavCuePoint()` y `FLVPlayback.seekToPrevNavCuePoint()` de la [Referencia de ActionScript 3.0 para Flash Professional](#).

**Activación y desactivación de puntos de referencia de archivo FLV incorporados**

Es posible activar y desactivar puntos de referencia de archivo FLV incorporados, mediante el método `setFLVCuePointEnabled()`. Los puntos de referencia desactivados no activan eventos `cuePoint` ni funcionan con los métodos `seekToCuePoint()`, `seekToNextNavCuePoint()` o `seekToPrevNavCuePoint()`. Sin embargo, puede buscar los puntos de referencia desactivados mediante los métodos `findCuePoint()`, `findNearestCuePoint()` y `findNextCuePointWithName()`.

Se puede probar si un punto de referencia de archivo FLV incorporado está activado, mediante el método `isFLVCuePointEnabled()`. En el siguiente ejemplo, se desactivan los puntos de referencia incorporados `point2` y `point3` cuando el vídeo está listo para ser reproducido. Sin embargo, cuando se produce el primer evento `cuePoint`, el controlador de eventos prueba si el punto de referencia `point3` está desactivado y, si lo está, lo activa.

```
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 obtener más información, consulte los métodos `FLVPlayback.isFLVCuePointEnabled()` y `FLVPlayback.setFLVCuePointEnabled()` de la [Referencia de ActionScript 3.0 para Flash Professional](#).

**Eliminación de un punto de referencia de ActionScript**

Se puede quitar un punto de referencia de ActionScript mediante el método `removeASCuePoint()`. En el siguiente ejemplo, se quita el punto de referencia `ASpt2` cuando se produce el punto de referencia `ASpt1`:

**Uso del componente FLVPlayback**

```
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
my_FLVPlybk.addASCuePoint(2.02, "ASpt1");//add AS cue point
my_FLVPlybk.addASCuePoint(3.4, "ASpt2");//add 2nd Aspt
my_FLVPlybk.addEventListener(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 obtener más información, consulte `FLVPlayback.removeASCuePoint()` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

**Reproducción de múltiples archivos de vídeo**

Se puede reproducir una secuencia de archivos de vídeo en una instancia de `FLVPlayback`; para ello, simplemente hay que cargar un nuevo URL en la propiedad `source` cuando finalice la reproducción del anterior archivo de vídeo. Por ejemplo, el siguiente código ActionScript detecta el evento `complete`, que se produce cuando finaliza la reproducción de un archivo de vídeo. Cuando se produce este evento, el código establece el nombre y la ubicación del nuevo archivo vídeo en la propiedad `source` y llama al método `play()` para reproducir el nuevo 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");
    }
};
```

**Uso de múltiples reproductores de vídeo**

También es posible abrir varios reproductores de vídeo en una sola instancia del componente `FLVPlayback` para reproducir varios vídeos y pasar de uno a otro durante la reproducción.

El reproductor de vídeo inicial se crea al arrastrar el componente `FLVPlayback` al escenario. El componente asigna automáticamente el número 0 al reproductor de vídeo inicial y lo convierte en el reproductor predeterminado. Para crear un reproductor de vídeo adicional, simplemente establezca la propiedad `activeVideoPlayerIndex` en un nuevo número. Cuando se establece la propiedad `activeVideoPlayerIndex`, se convierte además al reproductor de vídeo especificado en el reproductor de vídeo *activo*, que es el que se verá afectado por las propiedades y métodos de la clase `FLVPlayback`. Sin embargo, si se establece la propiedad `activeVideoPlayerIndex`, el reproductor de vídeo no pasará a ser visible. Para que el reproductor de vídeo sea visible, establezca la propiedad `visibleVideoPlayerIndex` en el número del reproductor de vídeo. Para obtener más información sobre el modo en que estas propiedades interactúan con los métodos y propiedades de la clase `FLVPlayback`, consulte las propiedades `FLVPlayback.activeVideoPlayerIndex` y `FLVPlayback.visibleVideoPlayerIndex` de la [Referencia de ActionScript 3.0 para Flash Professional](#).

El siguiente código ActionScript carga la propiedad `source` para reproducir un archivo de vídeo en el reproductor de vídeo predeterminado y le añade un punto de referencia. Cuando se produce el evento `ready`, el controlador de eventos abre un segundo reproductor de vídeo estableciendo la propiedad `activeVideoPlayerIndex` en el número 1. Especifica un archivo FLV y un punto de referencia para el segundo reproductor de vídeo y luego define nuevamente el reproductor predeterminado (0) como el reproductor de vídeo activo.

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

Para cambiar a otro archivo FLV mientras se reproduce uno, debe realizarse el cambio en el código ActionScript. Los puntos de referencia permiten intervenir en puntos específicos del archivo FLV mediante un evento `cuePoint`. El siguiente código crea un detector del evento `cuePoint` y llama a una función de controlador que pone en pausa el reproductor de vídeo activo (0), cambia al segundo reproductor (1) y reproduce su archivo FLV:

**Uso del componente FLVPlayback**

```

import fl.video.*;
// add listener for a cuePoint event
my_FLVPlybk.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
// add the handler function for the cuePoint event
function cp_listener(eventObject:MetadataEvent):void {
    // display the no. of the video player causing the event
    trace("Hit cuePoint event for player: " + eventObject.vp);
    // test for the video player and switch FLV files accordingly
    if (eventObject.vp == 0) {
        my_FLVPlybk.pause(); //pause the first FLV file
        my_FLVPlybk.activeVideoPlayerIndex = 1; // make the 2nd player active
        my_FLVPlybk.visibleVideoPlayerIndex = 1; // make the 2nd player visible
        my_FLVPlybk.play(); // begin playing the new player/FLV
    } else if (eventObject.vp == 1) {
        my_FLVPlybk.pause(); // pause the 2nd FLV
        my_FLVPlybk.activeVideoPlayerIndex = 0; // make the 1st player active
        my_FLVPlybk.visibleVideoPlayerIndex = 0; // make the 1st player visible
        my_FLVPlybk.play(); // begin playing the 1st player
    }
}
my_FLVPlybk.addEventListener(VideoEvent.COMPLETE, complete_listener);
function complete_listener(eventObject:VideoEvent):void {
    trace("Hit complete event for player: " + eventObject.vp);
    if (eventObject.vp == 0) {
        my_FLVPlybk.activeVideoPlayerIndex = 1;
        my_FLVPlybk.visibleVideoPlayerIndex = 1;
        my_FLVPlybk.play();
    } else {
        my_FLVPlybk.closeVideoPlayer(1);
    }
};

```

Cuando se crea un nuevo reproductor de vídeo, la instancia de FLVPlayback establece sus propiedades en el valor del reproductor de vídeo predeterminado, excepto las propiedades `source`, `totalTime`, `isLive`, que la instancia de FLVPlayback siempre establece en sus valores predeterminados: empty string, 0, y false. Establece en false la propiedad `autoPlay`, que tiene el valor predeterminado true en el reproductor de vídeo predeterminado. La propiedad `cuePoints` no produce ningún efecto, ni siquiera si se carga posteriormente en el reproductor de vídeo predeterminado.

Los métodos y las propiedades que controlan el volumen, la posición, las dimensiones, la visibilidad y los controles de interfaz de usuario siempre son globales y su comportamiento no se ve afectado al establecer el valor de la propiedad `activeVideoPlayerIndex`. Para obtener más información sobre estos métodos y propiedades, y el efecto que tiene establecer la propiedad `activeVideoPlayerIndex`, consulte la propiedad `FLVPlayback.activeVideoPlayerIndex` en la [Referencia de ActionScript 3.0 para Flash Professional](#). Las propiedades y métodos restantes se aplican al reproductor de vídeo identificado por el valor de la propiedad `activeVideoPlayerIndex`.

Sin embargo, las propiedades y métodos que controlan las dimensiones *sí interactúan* con la propiedad `visibleVideoPlayerIndex`. Para obtener más información, consulte la propiedad `FLVPlayback.visibleVideoPlayerIndex` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

## Transmisión de archivos FLV desde Flash Media Server

Los requisitos de transmisión de archivos FLV desde Flash Media Server varían en función de si el proveedor de Flash Video Streaming Service ofrece detección de ancho de banda nativa. La detección de ancho de banda nativa es la detección de ancho de banda incorporada en el servidor de transmisión y proporciona un mejor rendimiento. Debe consultar a su proveedor para saber si ofrece detección de ancho de banda nativa.

Para acceder a los archivos FLV de Flash Media Server, utilice un URL como `rtmp://my_servername/my_application/stream.flv`.

Cuando se reproduce un flujo en vivo con Flash Media Server, es necesario establecer la propiedad `isLive` de `FLVPlayback` en `true`. Para obtener más información, consulte la propiedad `FLVPlayback.isLive` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

Para obtener más información sobre la administración de Flash Media Server, incluida la configuración de un flujo en vivo, consulte la documentación de Flash Media Server en [www.adobe.com/support/documentation/en/flashmediaserver/](http://www.adobe.com/support/documentation/en/flashmediaserver/).

### Si hay detección de ancho de banda nativa o no la hay

La clase `NCManagerNative` es una subclase de `NCManager` que admite la detección de ancho de banda nativa que algunos proveedores de Flash Video Streaming Service pueden ofrecer. Cuando se utiliza `NCManagerNative`, no se necesita ningún archivo especial en Flash Media Server. `NCManagerNative` también permite conectar con cualquier versión de Flash Media Server, sin necesidad de un archivo `main.asc`, si no se requiere la detección de ancho de banda.

Para utilizar `NCManagerNative` en lugar de la clase `NCManager` predeterminada, añada las siguientes líneas de código en el primer fotograma del archivo FLA:

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

### Si hay detección de ancho de banda no nativa

Si el proveedor de Flash Video Streaming Service no ofrece detección de ancho de banda pero la necesita, debe añadir el archivo `main.asc` a la aplicación FLV de Flash Media Server. Puede encontrar el archivo `main.asc` file en línea en [www.adobe.com/go/learn\\_fl\\_samples\\_es](http://www.adobe.com/go/learn_fl_samples_es). Se encuentra en el archivo `Samples.zip`, dentro del directorio `Samples\ComponentsAS2\FLVPlayback`.

#### Para configurar Flash Media Server para la transmisión de archivos FLV:

- 1 Cree una carpeta en la carpeta de la aplicación Flash Media Server y asígnele un nombre como **my\_application**.
- 2 Copie el archivo `main.asc` en la carpeta `my_application`.
- 3 Cree una carpeta denominada **streams** en la carpeta `my_application`.
- 4 Cree una carpeta denominada **\_definst\_** en la carpeta `streams`.
- 5 Coloque los archivos FLV en la carpeta **\_definst\_**.

## Personalización del componente FLVPlayback

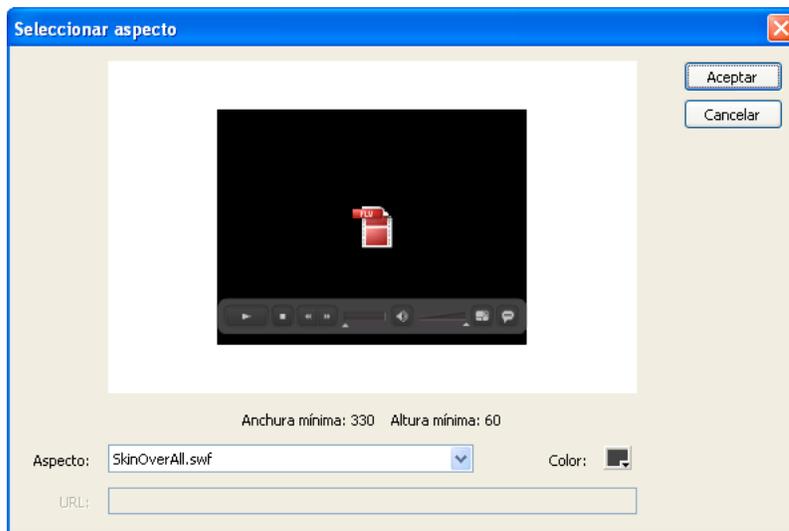
En esta sección, se explica la forma de personalizar el componente `FLVPlayback`. No obstante, la mayoría de los métodos que se utilizan para personalizar otros componentes no funcionan con el componente `FLVPlayback`. Para personalizar el componente `FLVPlayback`, utilice únicamente las técnicas que se describen en esta sección.

Existen las siguientes opciones para personalizar el componente FLVPlayback: seleccionar un aspecto prediseñado, aplicar un aspecto a los componentes individuales de interfaz de usuario personalizados de reproducción FLV o crear un nuevo aspecto. También se pueden utilizar las propiedades de FLVPlayback para modificar el comportamiento de un aspecto.

**Nota:** se debe cargar el archivo SWF de aspecto en el servidor web, junto con el archivo SWF de la aplicación para que el aspecto funcione con el componente FLVPlayback.

## Selección de un aspecto prediseñado

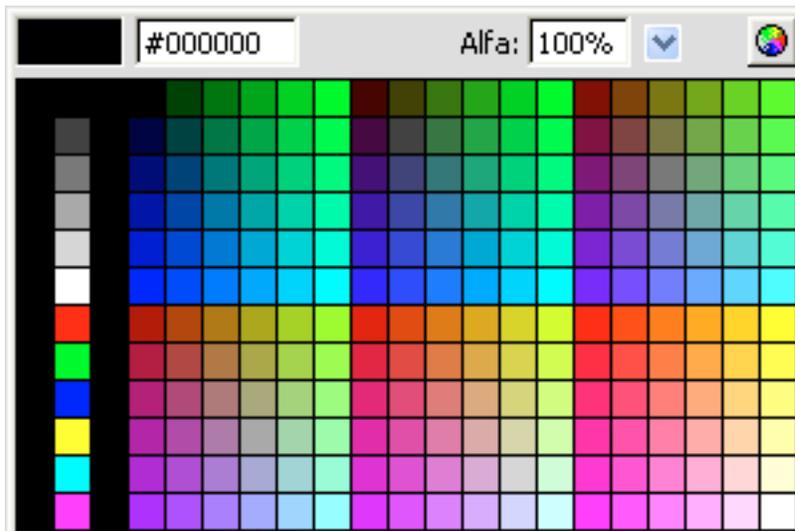
Para elegir un aspecto para el componente FLVPlayback, haga clic en la celda `value` del parámetro `skin` en el inspector de componentes. A continuación, haga clic en el icono de lupa para abrir el siguiente cuadro de diálogo Seleccionar aspecto, que le permite seleccionar un aspecto o proporcionar una URL que especifique la ubicación del archivo SWF de aspecto.



Cuadro de diálogo Seleccionar aspecto de FLVPlayback

Los aspectos que se listan en menú emergente Aspecto se encuentran en la carpeta de aplicación de Flash /Flash Configuration/FLVPlayback Skins/ActionScript 3.0. Se pueden diseñar nuevos aspectos disponibles en este cuadro de diálogo creándolos y colocándolos en el archivo SWF en la carpeta. El nombre aparece en el menú emergente con una extensión `.swf`. Para más información sobre la creación de un conjunto de aspectos, consulte [“Creación de un aspecto nuevo”](#) en la página 167.

En los aspectos que se asignan estableciendo la propiedad `skin`, ya sea mediante la definición del parámetro `skin` durante la edición o con ActionScript en tiempo de ejecución, se pueden asignar valores de color y alfa (transparencia) aunque no se seleccione el aspecto. Para asignar valores de color y alfa durante la edición, abra el selector de color en el cuadro de diálogo Seleccionar aspecto, como se muestra a continuación.



Selector de color en el cuadro de diálogo Seleccionar aspecto

Para elegir el color, haga clic en una de las muestras del panel o introduzca su valor numérico en el cuadro de texto. Para elegir el valor de alfa, utilice el deslizador o escriba un porcentaje en el cuadro de texto Alfa.

Para asignar valores de color y alfa en tiempo de ejecución, establezca las propiedades `skinBackgroundColor` y `skinBackgroundAlpha`. Establezca la propiedad `skinBackgroundColor` en un valor `0xRRGGBB` (rojo, verde, azul). Establezca la propiedad `skinBackgroundAlpha` en un valor entre 0.0 y 1.0. El siguiente ejemplo establece `skinBackgroundColor` en `0xFF0000` (rojo) y `skinBackgroundAlpha` en `.5`.

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

Los valores predeterminados son los últimos valores que elige el usuario.

Si desea aplicar aspectos al componente FLVPlayback mediante los componentes de interfaz de usuario personalizados de reproducción FLV, seleccione Ninguno en el menú emergente.

## Aplicación de aspectos a componentes individuales de interfaz de usuario personalizados de reproducción FLV

Los componentes de interfaz de usuario personalizados de reproducción FLV permiten personalizar la apariencia de los controles de FLVPlayback en el archivo FLA y permiten ver los resultados en una previsualización de la página web. Sin embargo, estos componentes no están diseñados para cambiar de tamaño. Debería editar un clip de película y su contenido con un tamaño específico. Por este motivo, es preferible tener el componente FLVPlayback en el escenario con el tamaño deseado, con la propiedad `scaleMode` establecida en `exactFit`.

Para empezar, arrastre simplemente los componentes de interfaz de usuario personalizados de reproducción de FLV que desee del panel Componentes al escenario y asigne a cada uno de ellos un nombre de instancia.

Estos componentes pueden funcionar sin ningún código ActionScript. Si se colocan en la misma línea de tiempo y el mismo fotograma que el componente FLVPlayback y no hay ningún aspecto establecido en el componente, el componente FLVPlayback se conectará automáticamente a ellos. Si hay varios componentes FLVPlayback en el escenario, o si el control personalizado y la instancia de FLVPlayback no están en la misma línea de tiempo, es necesario realizar alguna acción.

Cuando los componentes estén en el escenario, podrá editarlos como haría con cualquier otro símbolo. Cuando abra los componentes observará que la configuración de cada uno de ellos es ligeramente distinta.

## Componentes Button

Los componentes Button tienen una estructura similar. Los distintos tipos de botón son: BackButton, ForwardButton, MuteButton, PauseButton, PlayButton, PlayPauseButton y StopButton. La mayoría tiene un solo clip de película en el fotograma 1, con el nombre de instancia placeholder\_mc. Suele ser una instancia del estado normal del botón, aunque no siempre es así. En el fotograma 2, hay cuatro clips de película en el escenario para cada estado de visualización: normal, over, down, y disabled. (En tiempo de ejecución, el componente nunca pasa realmente al fotograma 2; estos clips de película se colocan aquí para facilitar la edición y para obligarles a que se carguen en el archivo SWF sin necesidad de activar la casilla de verificación Exportar en primer fotograma en el cuadro de diálogo Propiedades de símbolo. No obstante, debe activar la opción Exportar para ActionScript.)

Para aplicar aspectos al botón, edite cada uno de estos clips de película. Puede cambiar su tamaño y su apariencia.

Normalmente se muestra parte del código ActionScript en el fotograma 1. No es necesario modificar este script. Simplemente detiene la cabeza lectora en el fotograma 1 y especifica los clips de película que se utilizarán en cada estado.

## Botones PlayPauseButton, MuteButton, FullScreenButton y CaptionButton

Los botones PlayPauseButton, MuteButton, FullScreenButton y CaptionButton se configuran de forma distinta al resto de botones; sólo tienen un fotograma con dos capas y ningún script. En dicho fotograma, hay dos botones, uno encima del otro: en el caso de PlayPauseButton, un botón de reproducción y otro de pausa; en el caso de MuteButton, un botón para activar el silencio y otro para desactivarlo; en el caso de FullScreenButton, un botón para activar la pantalla completa y otro para desactivarla; en el caso de CaptionButton, un botón para activar la subtítulos y otro para desactivarla. Para aplicar un aspecto a estos botones, aplique un aspecto a cada uno de estos dos botones internos como se describe en [“Aplicación de aspectos a componentes individuales de interfaz de usuario personalizados de reproducción FLV”](#) en la página 161; no se requiere ninguna acción adicional.

El botón CaptionButton se utiliza en el componente FLVPlaybackCaptioning y debe asociarse a él y no al componente FLVPlayback.

## Botones BackButton y ForwardButton

Los botones BackButton y ForwardButton también se configuran de forma distinta al resto de botones. En el fotograma 2, disponen de clips de película adicionales que pueden utilizarse como fotogramas alrededor de uno de los botones o de ambos. Estos clips de película no son necesarios ni ofrecen ninguna función especial y simplemente se proporcionan como una opción más. Para utilizarlos, arrástrelos al escenario desde el panel Biblioteca y colóquelos en el lugar que elija. Si no le interesan, no los utilice o elimínelos del panel Biblioteca.

La mayoría de los botones proporcionados se basan en un conjunto común de clips de película, de forma que es posible cambiar la apariencia de todos los botones de una sola vez. Puede utilizar esta capacidad o puede reemplazar los clips de película comunes y dotar a cada botón de un aspecto distinto.

## Componente BufferingBar

El componente BufferingBar es simple. Está formado por una animación que se hace visible cuando el componente entra en estado de almacenamiento en búfer y no requiere ningún código de ActionScript especial para configurarlo. De forma predeterminada, es una barra discontinua que se mueve de izquierda a derecha y tiene aplicada una máscara rectangular que le confiere un aspecto de barra móvil. Esta configuración no presenta ninguna característica especial.

Aunque las barras de almacenamiento en búfer de los archivos SWF de aspecto utilizan la escala de 9 divisiones porque deben ajustar su escala en tiempo de ejecución, el componente de interfaz de usuario personalizado de reproducción de FLV BufferingBar *no puede* utilizar la escala de 9 divisiones porque tiene clips de película anidados. Si desea alargar o ensanchar el componente BufferingBar, quizá deba cambiar su contenido en lugar de ajustar su escala.

## Componentes SeekBar y VolumeBar

Los componentes SeekBar y VolumeBar son similares, aunque tienen distintas funciones. Cada uno de ellos tiene selectores, utiliza los mismos mecanismos de seguimiento de selectores y admite clips anidados para realizar un seguimiento del progreso y del grado de relleno.

En muchas partes del código ActionScript en el componente FLVPlayback se supone que el punto de registro (también denominado *punto de origen* o *punto cero*) del componente SeekBar o VolumeBar está situado en la esquina superior izquierda del contenido, de modo que es importante mantener esta convención. De lo contrario, podría tener problemas con los selectores y con los clips de película de progreso y grado de relleno.

Aunque las barras de búsqueda de los archivos SWF de aspecto utilizan la escala de 9 divisiones porque deben ajustar su escala en tiempo de ejecución, el componente de interfaz de usuario personalizado de reproducción de FLV SeekBar *no puede* utilizar la escala de 9 divisiones porque tiene clips de película anidados. Si desea alargar o ensanchar el componente SeekBar, quizá deba cambiar su contenido en lugar de ajustar su escala.

## Selector

El fotograma 2 contiene una instancia del clip de película de selector. Al igual que los componentes BackButton y ForwardButton, el componente nunca pasa realmente al fotograma 2; estos clips de película se colocan aquí para facilitar la edición y para obligarles a que se carguen en el archivo SWF sin necesidad de activar la casilla de verificación Exportar en primer fotograma en el cuadro de diálogo Propiedades de símbolo. No obstante, debe activar la opción Exportar para ActionScript.

Quizá haya observado que el clip de película del selector tiene un rectángulo en el fondo, cuyo alfa está establecido en 0. Este rectángulo aumenta el tamaño del área activa del selector para poder agarrarlo con mayor facilidad sin cambiar su aspecto, de forma similar al estado Zona activa de un botón. Dado que el selector se crea dinámicamente en tiempo de ejecución, debe ser un clip de película y no un botón. Este rectángulo con el alfa establecido en 0 no es necesario por ningún otro motivo y, normalmente, puede reemplazar el interior del selector por la imagen que desee. Sin embargo, es recomendable mantener el punto de registro centrado horizontalmente en mitad del clip de película del selector.

El siguiente código ActionScript del fotograma 1 del componente SeekBar controla el selector:

```
stop();  
handleLinkageID = "SeekBarHandle";  
handleLeftMargin = 2;  
handleRightMargin = 2;  
handleY = 11;
```

La llamada a la función `stop()` es necesaria debido al contenido del fotograma 2.

La segunda línea especifica el símbolo que se utilizará como selector. No es necesario cambiarlo si simplemente se edita la instancia del clip de película del selector en el fotograma 2. En tiempo de ejecución, el componente FLVPlayback crea una instancia del clip de película especificado en el escenario como elemento del mismo nivel de la instancia del componente Bar, lo que significa que tienen el mismo clip de película principal. De esta forma, si la barra se encuentra en el nivel raíz, el selector debe estar también en el nivel raíz.

La variable `handleLeftMargin` determina la ubicación original del selector (0%) y la variable `handleRightMargin` determina su ubicación final (100%). Los números indican los desplazamientos desde los extremos izquierdo y derecho del control de la barra, con números positivos que marcan los límites dentro de la barra y números negativos que marcan los límites fuera de ella. Estos desplazamientos especifican el lugar donde puede dirigirse el selector, en función de su punto de registro. Si coloca el punto de registro en mitad del selector, los extremos izquierdo y derecho del selector traspasarán los límites. Un clip de película de barra de búsqueda debe tener el punto de registro en la esquina superior izquierda de su contenido para que funcione correctamente.

La variable `handleY` determina la posición *y* del selector con respecto a la instancia de la barra. Esto se basa en los puntos de registro de cada clip de película. El punto de registro en el selector de ejemplo se encuentra en la punta del triángulo, lo que permite colocarlo con respecto a la parte visible, sin tener en cuenta el rectángulo de estado de zona activa invisible. Además, un clip de película de barra debe tener el punto de registro en la esquina superior izquierda de su contenido para que funcione correctamente.

De esta forma, con estos límites, si el control de la barra se establece en (100, 100) y tiene una anchura de 100 píxeles, el selector abarcará de 102 a 198 horizontalmente y se mantendrá en 111 verticalmente. Si cambia los valores de `handleLeftMargin` y `handleRightMargin` a -2 y de `handleY` a -11, el selector abarcará de 98 a 202 horizontalmente y se mantendrá en 89 verticalmente.

### Clips de película de progreso y grado de relleno

El componente `SeekBar` tiene un clip de película de *progreso* y el componente `VolumeBar` tiene un clip de película de *grado de relleno* pero, en la práctica, los componentes `SeekBar` o `VolumeBar` pueden tener o no alguno de estos componentes, o ambos. Ambos componentes presentan la misma estructura y un comportamiento similar, pero realizan un seguimiento de distintos valores. Un clip de película de progreso se rellena a medida que se descarga el archivo FLV (lo que resulta útil sólo en las descargas de HTTP, porque siempre aparece lleno en las transmisiones desde servidores FMS) y un clip de película de grado de relleno se va rellenando a medida que el selector se mueve de izquierda a derecha.

El componente `FLVPlayback` busca estas instancias de clip de película a través de un nombre de instancia específico, de modo que la instancia del clip de película de progreso debe tener el clip de película de barra como elemento principal y el nombre de instancia `progress_mc`. La instancia del clip de película de grado de relleno debe tener el nombre de instancia `fullness_mc`.

Puede establecer clips de película de progreso y de grado de relleno con o sin la instancia del clip de película `fill_mc` anidada en ellos. El clip de película `fullness_mc` de `VolumeBar` muestra el método *con* el clip de película `fill_mc` y el clip de película `progress_mc` de `SeekBar` muestra el método *sin* el clip de película `fill_mc`.

El método *con* el clip de película `fill_mc` anidado resulta útil si se desea utilizar un relleno cuya escala no puede ajustarse sin distorsionar la apariencia.

En el clip de película `fullness_mc` de `VolumeBar`, se aplica una máscara a la instancia del clip de película `fill_mc` anidado. Se le puede aplicar una máscara cuando se cree el clip de película o esperar a que se cree la máscara dinámicamente en tiempo de ejecución. Si se aplica una máscara con un clip de película, denomine a la instancia **`mask_mc`** y configúrela de forma que `fill_mc` aparezca tal y como lo haría con un porcentaje del 100%. Si no aplica una máscara a `fill_mc`, la máscara creada dinámicamente será rectangular y tendrá el mismo tamaño que `fill_mc` al 100%.

Hay dos formas de mostrar el clip de película `fill_mc` con la máscara, en función de si el valor de `fill_mc.slideReveal` es `true` o `false`.

Si `fill_mc.slideReveal` es `true`, `fill_mc` se mueve de izquierda a derecha para mostrarse a través de la máscara. Con un porcentaje del 0%, aparecerá en el extremo izquierdo y no se mostrará a través de la máscara. A medida que aumenta el porcentaje, se va moviendo a la derecha hasta alcanzar el 100% y finalmente vuelve al lugar donde se creó en el escenario.

Si `fill_mc.slideReveal` es `false` o `undefined` (comportamiento predeterminado), se cambiará el tamaño de la máscara de izquierda a derecha para mostrar algo más de `fill_mc`. Si el porcentaje es 0%, se cambia el tamaño de la máscara a 05 horizontalmente y, a medida que va aumentando el porcentaje, el valor de `scaleX` aumenta hasta que, al 100%, muestra `fill_mc` completamente. En este punto, el valor de `scaleX` no es necesariamente 100 porque es posible que se ajustara la escala de `mask_mc` cuando se creó.

El método sin `fill_mc` es más sencillo que el método con `fill_mc`, pero distorsiona horizontalmente el relleno. Si se desea evitar dicha distorsión, debe utilizarse `fill_mc`. La instancia `progress_mc` de `SeekBar` ilustra este método.

El clip de película de progreso o de grado de relleno ajusta su escala horizontalmente en función del porcentaje. Al 0%, el valor de `scaleX` de la instancia se establece en 0, haciéndolo invisible. A medida que aumenta el porcentaje, el valor de `scaleX` se ajusta hasta que, al llegar al 100%, el clip alcanza el mismo tamaño que tenía cuando se creó en el escenario. Nuevamente, en este punto, el valor de `scaleX` no es necesariamente 100 porque es posible que se ajustara la escala de la instancia del clip cuando se creó.

## Conexión de los componentes de interfaz de usuario personalizados de reproducción FLV

Si se colocan los componentes de interfaz de usuario en la misma línea de tiempo y el mismo fotograma que el componente `FLVPlayback` y no se ha establecido la propiedad `skin`, el componente `FLVPlayback` se conectará automáticamente a ellos sin necesidad de `ActionScript`.

Si hay varios componentes `FLVPlayback` en el escenario o si el control personalizado y el componente `FLVPlayback` no están en la misma línea de tiempo, se debe escribir código `ActionScript` para conectar los componentes de interfaz de usuario personalizados con la instancia del componente `FLVPlayback`. Primero, debe asignar un nombre a la instancia de `FLVPlayback` y utilizar código `ActionScript` para asignar las instancias del componente de interfaz de usuario personalizado de reproducción FLV a las propiedades de `FLVPlayback` correspondientes. En el siguiente ejemplo, la instancia de `FLVPlayback` es `my_FLVPlybk`, los nombres de propiedades de `FLVPlayback` se especifican después de los puntos (.) y las instancias de control de interfaz de usuario personalizado de reproducción FLV se sitúan a la derecha del signo igual (=):

```
//FLVPlayback instance = my_FLVPlybk
my_FLVPlybk.playButton = playbtn; // set playButton prop. to playbtn, etc.
my_FLVPlybk.pauseButton = pausebtn;
my_FLVPlybk.playPauseButton = playpausebtn;
my_FLVPlybk.stopButton = stopbtn;
my_FLVPlybk.muteButton = mutebtn;
my_FLVPlybk.backButton = backbtn;
my_FLVPlybk.forwardButton = forbtn;
my_FLVPlybk.volumeBar = volbar;
my_FLVPlybk.seekBar = seekbar;
my_FLVPlybk.bufferingBar = bufbar;
```

Los siguientes pasos crean controles `StopButton`, `PlayPauseButton`, `MuteButton` y `SeekBar` personalizados:

- 1 Arrastre el componente `FLVPlayback` al escenario y asígnele el nombre de instancia **my\_FLVPlybk**.
- 2 Establezca el parámetro `source` en <http://www.helpexamples.com/flash/video/cuepoints.flv> a través del inspector de componentes.
- 3 Establezca el parámetro `Skin` en `None`.
- 4 Arrastre una instancia de `StopButton`, `PlayPauseButton` y `MuteButton` al escenario y colóquelas sobre la instancia de `FLVPlayback`, apilándolas verticalmente a la izquierda. Asigne a cada botón un nombre de instancia en el inspector de propiedades (por ejemplo, **my\_stopbtn**, **my\_playpausebtn** y **my\_mutebtn**).
- 5 En el panel Biblioteca, abra la carpeta `FLVPlayback Skins` y, a continuación, la subcarpeta `SquareButton`.
- 6 Seleccione el clip de película `SquareBgDown` y haga doble clic en él para abrirlo en el escenario.

- 7 Haga clic con el botón derecho del ratón (Windows) o con la tecla Control presionada (Macintosh), elija Seleccionar todo en el menú y elimine el símbolo.
- 8 Seleccione la herramienta Óvalo, dibuje un óvalo en la misma ubicación y establezca el color azul como relleno (#0033FF).
- 9 En el inspector de propiedades, establezca la anchura ( (W:) en **40** y la altura (H:) en **20**. Establezca la coordenada x (X::) en **0.0** y la coordenada y (Y::) en **0.0**.
- 10 Repita los pasos 6 a 8 para SquareBgNormal, pero cambie el color de relleno por amarillo (#FFFF00).
- 11 Repita los pasos 6 a 8 para SquareBgOver, pero cambie el color de relleno por verde (#006600).
- 12 Edite los clips de película de los diversos iconos de símbolos en los botones (PauseIcon, PlayIcon, MuteOnIcon, MuteOffIcon y StopIcon). Encontrará estos clips de película en el panel Biblioteca, en la carpeta FLV Playback Skins/Label Button/Assets, donde *Label* es el nombre del botón; por ejemplo, Play, Pause, etc. Siga los pasos siguientes para cada elemento:
  - a Elija la opción Seleccionar todo.
  - b Cambie el color por rojo (#FF0000).
  - c Aplique una escala de 300%.
  - d Cambie la ubicación de X: del contenido por **7.0** para modificar la ubicación horizontal del icono en cada estado de botón.

*Nota: al cambiar la ubicación de esta forma, se evita tener que abrir cada estado de botón y mover la instancia de clip de película del icono.*
- 13 Haga clic en la flecha Atrás azul situada por encima de la línea de tiempo para volver a la escena 1, fotograma 1.
- 14 Arrastre un componente SeekBar al escenario y colóquelo en la esquina inferior derecha de la instancia de FLVPlayback.
- 15 En el panel Biblioteca, haga doble clic en el componente SeekBar para abrirlo en el escenario.
- 16 Ajuste la escala un 400%.
- 17 Seleccione el contorno y establezca el color rojo (#FF0000).
- 18 Haga doble clic en SeekBarProgress en la carpeta FLVPlayback Skins/Seek Bar y establezca el color amarillo #FFFF00).
- 19 Haga doble clic en SeekBarHandle en la carpeta FLVPlayback Skins/Seek Bar y establezca el color rojo (#FF0000).
- 20 Haga clic en la flecha Atrás azul situada por encima de la línea de tiempo para volver a la escena 1, fotograma 1.
- 21 Seleccione la instancia de SeekBar en el escenario y asígnele el nombre de instancia **my\_seekbar**.
- 22 En el panel Acciones del fotograma 1 de la línea de tiempo, añada una sentencia de importación para las clases de vídeo y asigne nombres de botón y de barra de búsqueda a las propiedades de FLVPlayback correspondientes, tal y como se muestra en el siguiente ejemplo:

```
import fl.video.*;
my_FLVPlybk.stopButton = my_stopbbtn;
my_FLVPlybk.playPauseButton = my_plypausbbtn;
my_FLVPlybk.muteButton = my_mutebbtn;
my_FLVPlybk.seekBar = my_seekbar;
```
- 23 Pulse Control+Intro para probar la película.

## Creación de un aspecto nuevo

La mejor forma de crear un archivo SWF de aspecto es copiar uno de los archivos de aspecto que incorpora Flash y utilizarlo como punto de partida. Los archivos FLA de estos aspectos se encuentran en la carpeta de la aplicación Flash, en Configuration/FLVPlayback Skins/FLA/ActionScript 3.0/. Para que el archivo SWF de aspecto finalizado esté disponible como una opción en el cuadro de diálogo Seleccionar aspecto, colóquelo en la carpeta Configuration/FLVPlayback Skins/ActionScript 3.0, ya sea en la carpeta de la aplicación Flash o en una carpeta Configuration/FLVPlayback Skins/ActionScript 3.0 local de un usuario.

Dado que es posible establecer el color de un aspecto independientemente de que se elija o no el aspecto, no es necesario editar el archivo FLA para modificar el color. Si se crea un aspecto que tiene un color específico y se quiere evitar que pueda modificarse en el cuadro de diálogo Seleccionar aspecto, debe establecerse `this.border_mc.colorMe = false;` en el código ActionScript del archivo FLV de aspecto. Para información sobre cómo establecer el color de un aspecto, consulte [“Selección de un aspecto prediseñado”](#) en la página 160.

Al observar los archivos FLA de aspectos de Flash instalados, puede parecer que algunos elementos del escenario no son necesarios, pero muchos de estos elementos se incluyen en capas de guía. Con el uso de la previsualización dinámica con la escala de 9 divisiones se puede ver rápidamente cómo aparecerá realmente del archivo SWF.

En las siguientes secciones se describen personalizaciones y cambios más complejos de los clips de película SeekBar, BufferingBar y VolumeBar.

## Uso del diseño del aspecto

Cuando se abre un archivo FLA de aspectos de Flash, los clips de película del aspecto aparecen dispuestos en la línea de tiempo principal. Estos clips y el código ActionScript que se encuentran en el mismo fotograma definen la disposición de los controles en tiempo de ejecución.

Aunque la capa de diseño se parezca mucho a la apariencia del aspecto en tiempo de ejecución, el contenido de esta capa no es visible en tiempo de ejecución. Sólo se utiliza para calcular dónde se ubicarán los controles. Los otros controles del escenario se utilizan en tiempo de ejecución.

En la capa de diseño hay un marcador de posición del componente FLVPlayback, denominado `video_mc`. Todos los demás controles se disponen con respecto a `video_mc`. Si se empieza con uno de los archivos FLA de Flash y se cambia el tamaño de los controles, probablemente se podrá ajustar el diseño moviendo estos clips de marcador de posición.

Cada uno de los clips de marcador de posición tiene un nombre de instancia específico. Los nombres de los clips de marcador de posición son: `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` y `volumeBarHandle_mc`. La parte que cambia de color cuando se elige un color de aspecto se denomina `border_mc`.

No importa qué clip se utilice para un control. Normalmente, en los botones se utiliza el clip de estado normal. En los demás controles, se utiliza el clip para dicho control, pero esto es sólo por comodidad. Lo único que importa es la ubicación de  $x$  (horizontal) e  $y$  (vertical) y la altura y anchura del marcador de posición.

Se pueden tener tantos clips adicionales como se desee, además de los controles estándar. El único requisito de estos clips es que sus símbolos de biblioteca tengan activada la casilla de verificación Exportar para ActionScript en el cuadro de diálogo Vinculación. Los clips personalizados de la capa de diseño pueden tener cualquier nombre de instancia, excepto los nombres de instancia reservados que aparecen en la lista anterior. El nombre de instancia sólo se necesita para establecer código ActionScript en los clips que determinan el diseño.

El clip `border_mc` es especial. Si se establece la propiedad `FlvPlayback.skinAutoHide` en `true`, el aspecto muestra cuándo se desplaza el ratón por encima del clip `border_mc`. Es importante para los aspectos que aparecen fuera de los límites del reproductor de vídeo. Para información sobre la propiedad `skinAutoHide`, consulte [“Modificación del comportamiento del aspecto”](#) en la página 171.

En los archivos FLA de Flash, `border_mc` se utiliza para el fondo cromático y para el borde de los botones para avanzar y retroceder.

El clip `border_mc` es también la parte del aspecto cuyo alfa y color cambia a través de las propiedades `skinBackgroundAlpha` y `skinBackgroundColor`. Para permitir la personalización del color y del alfa, el código ActionScript del archivo FLA de aspecto debe incluir lo siguiente:

```
border_mc.colorMe = true;
```

### ActionScript y diseño del aspecto

El siguiente código ActionScript se aplica de forma general a todos los controles. Algunos controles tienen código ActionScript específico que define un comportamiento adicional y que se explica en la sección correspondiente a dicho control.

El código ActionScript inicial es una sección amplia que especifica los nombres de clase para cada estado del componente. Todos estos nombres de clase se encuentran en el archivo `SkinOverAll fla`. El código tiene el siguiente aspecto para los botones de pausa y reproducción, por ejemplo:

```
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";
```

Los nombres de clase no tienen archivos de clase externos reales; simplemente se especifican en el cuadro de diálogo Vinculación para todos los clips de película de la biblioteca.

En el componente ActionScript 2.0, había clips de película en el escenario que se utilizaban realmente en tiempo de ejecución. En el componente ActionScript 3.0, estos clips de película continúan en el archivo FLA, pero simplemente para facilitar la edición. Ahora se incluyen todos en capas de guías y no se exportan. Todos los aspectos de la biblioteca se exportan en el primer fotograma y se crean dinámicamente con código como el siguiente, por ejemplo:

```
new fl.video.skin.PauseButtonDisabled();
```

Después de esa sección, se encuentra el código ActionScript que define los valores máximos de altura y anchura del aspecto. El cuadro de diálogo Seleccionar aspecto muestra estos valores, que se utilizan en tiempo de ejecución para evitar que el aspecto ajuste su escala por debajo del tamaño mínimo. Si no desea especificar un tamaño mínimo, déjelo como valor `undefined`, o inferior o igual a cero.

```
// 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;
```

En cada marcador de posición pueden aplicarse las siguientes propiedades:

Propiedad	Descripción
anchorLeft	Boolean. Ubica el control con respecto al lado izquierdo de la instancia de FLVPlayback. El valor predeterminado es <code>true</code> a menos que <code>anchorRight</code> se establezca explícitamente en <code>true</code> ; luego el valor predeterminado pasa a ser <code>false</code> .
anchorRight	Boolean. Ubica el control con respecto al lado derecho de la instancia de FLVPlayback. El valor predeterminado es <code>false</code> .
anchorBottom	Boolean. Ubica el control con respecto a la parte inferior de la instancia de FLVPlayback. El valor predeterminado es <code>true</code> a menos que <code>anchorTop</code> se establezca explícitamente en <code>true</code> ; luego el valor predeterminado pasa a ser <code>false</code> .
anchorTop	Boolean. Ubica el control con respecto a la parte superior de la instancia de FLVPlayback. El valor predeterminado es <code>false</code> .

Si las propiedades `anchorLeft` y `anchorRight` tienen el valor `true`, el control ajusta su escala horizontalmente en tiempo de ejecución. Si las propiedades `anchorTop` y `anchorBottom` tienen el valor `true`, el control ajusta su escala verticalmente en tiempo de ejecución.

Para ver los efectos de estas propiedades, consulte su uso en los aspectos de Flash. Los controles `BufferingBar` y `SeekBar` son los únicos que ajustan su escala, y se disponen uno encima del otro y tienen las propiedades `anchorLeft` y `anchorRight` establecidas en `true`. Todos los controles a la izquierda de `BufferingBar` y `SeekBar` tienen la propiedad `anchorLeft` establecida en `true`, y todos los controles a la derecha tienen la propiedad `anchorRight` establecida en `true`. Todos los controles tienen la propiedad `anchorBottom` establecida en `true`.

Se puede intentar editar los clips de película en la capa de diseño para crear un aspecto donde los controles se sitúen en la parte superior en lugar de en la parte inferior. Simplemente necesita mover los controles a la parte superior, con respecto a `video_mc`, y establecer `anchorTop` en `true` en todos los controles.

### Barra de almacenamiento en búfer

La barra de almacenamiento en búfer tiene dos clips de película: `bufferingBar_mc` y `bufferingBarFill_mc`. La posición de cada clip en el escenario con respecto al otro clip es importante porque se mantiene su posición relativa. La barra de almacenamiento en búfer utiliza dos clips independientes porque el componente ajusta la escala de `bufferingBar_mc` pero no de `bufferingBarFill_mc`.

El clip `bufferingBar_mc` tiene aplicada una escala de 9 divisiones, de forma que los bordes no se verán distorsionados al cambiar la escala. El clip `bufferingBarFill_mc` es muy amplio y no necesita ningún ajuste de escala. Se enmascara automáticamente en tiempo de ejecución para mostrar únicamente la parte situada por encima del clip `bufferingBar_mc` estirado. De forma predeterminada, las dimensiones exactas de la máscara mantendrán un margen igual a izquierda y derecha en el clip `bufferingBar_mc`, basándose en la diferencia entre las posiciones de `x` (horizontal) de `bufferingBar_mc` y `_mc`. Para personalizar la posición, utilice código ActionScript.

Si la barra de almacenamiento en búfer no necesita ajustar la escala o no utiliza escala en 9 divisiones, puede configurarla como el componente de interfaz de usuario personalizado de reproducción FLV `BufferingBar`. Para más información, consulte “[Componente BufferingBar](#)” en la página 162.

La barra de almacenamiento en búfer tiene la siguiente propiedad adicional:

Propiedad	Descripción
<code>fill_mc:MovieClip</code>	Especifica el nombre de instancia del relleno de la barra de almacenamiento en búfer. Su valor predeterminado es <code>bufferingBarFill_mc</code> .

## Barra de búsqueda y barra de volumen

La barra de búsqueda también tiene dos clips de película: seekBar\_mc y seekBarProgress\_mc. La posición de cada clip en la capa de diseño con respecto al otro clip es importante porque se mantiene su posición relativa. Aunque ambos clips ajustan su escala, el clip seekBarProgress\_mc no puede anidarse en seekBar\_mc porque seekBar\_mc utiliza la escala en 9 divisiones, que no funciona correctamente con clips de película anidados.

El clip seekBar\_mc tiene aplicada una escala de 9 divisiones, de forma que los bordes no se verán distorsionados al cambiar la escala. El clip seekBarProgress\_mc también ajusta su escala pero no se distorsiona. No utiliza la escala de 9 divisiones porque es un relleno y no se ve afectado cuando se distorsiona.

El clip seekBarProgress\_mc funciona sin fill\_mc, de forma muy similar al modo en que el clip progress\_mc funciona en los componentes de interfaz de usuario personalizados de reproducción de FLV. Dicho de otro modo, no se enmascara y ajusta su escala horizontalmente. Las dimensiones exactas del clip seekBarProgress\_mc al 100% se definen por los márgenes izquierdo y derecho del clip seekBarProgress\_mc. Estas dimensiones son iguales, de forma predeterminada, y se basan en la diferencia entre las posiciones de  $x$  (horizontal) de seekBar\_mc y seekBarProgress\_mc. Puede personalizar las dimensiones mediante código ActionScript en el clip de película de la barra de búsqueda, tal y como se muestra en el siguiente ejemplo:

```
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;
```

Se puede incluir este código en la línea de tiempo del clip de película SeekBar o bien junto el otro código ActionScript en la línea de tiempo principal. Si se personaliza con código en lugar de modificarse el diseño, no es necesario que el relleno esté en el escenario. Basta con que esté en la biblioteca y se haya establecido que se debe exportar para ActionScript en el fotograma 1 con el nombre de clase correcto.

Al igual que el componente de interfaz de usuario personalizado de reproducción de FLV SeekBar, es posible crear un clip de película de grado de relleno para la barra de búsqueda. Si la barra de búsqueda no necesita un ajuste de escala o si se ajusta su escala pero no utiliza la escala de 9 divisiones, puede configurar el clip progress\_mc o fullness\_mc mediante cualquiera de los métodos que se utilizan para los componentes de interfaz de usuario personalizados de reproducción FLV. Para más información, consulte .

Dado que la barra de volumen de aspectos de Flash no ajusta su escala, se crea de la misma forma que el componente de interfaz de usuario personalizado de reproducción de FLV VolumeBar. Para más información, consulte [“Componentes SeekBar y VolumeBar”](#) en la página 163. La excepción es que el selector se implementa de forma distinta.

## Selectores SeekBar y VolumeBar

Los selectores de SeekBar y VolumeBar se colocan en la capa de diseño, junto a la barra. De forma predeterminada, el margen izquierdo, el margen derecho y los valores del eje  $y$  del selector se establecen mediante su posición relativa al clip de película de la barra. El margen izquierdo se establece mediante la diferencia entre la posición  $x$  (horizontal) del selector y la posición  $x$  (horizontal) de la barra, y el margen derecho es igual al margen izquierdo. Puede personalizar estos valores mediante código ActionScript en el clip de película SeekBar o VolumeBar. El siguiente ejemplo es el mismo código ActionScript que se utiliza con los componentes de interfaz de usuario personalizados de reproducción de FLV:

```
this.seekBar_mc.handleLeftMargin = 2;  
this.seekBar_mc.handleRightMargin = 2;  
this.seekBar_mc.handleY = 11;
```

Se puede incluir este código en la línea de tiempo del clip de película SeekBar o bien junto el otro código ActionScript en la línea de tiempo principal. Si se personaliza con código en lugar de modificarse el diseño, no es necesario que el selector esté en el escenario. Basta con que esté en la biblioteca y se haya establecido que se debe exportar para ActionScript en el fotograma 1 con el nombre de clase correcto.

Más allá de estas propiedades, los selectores son simples clips de película configurados del mismo modo que en los componentes de interfaz de usuario personalizados de reproducción de FLV. Ambos tienen fondos rectangulares con la propiedad `alpha` establecida en 0. Estos fondos sólo sirven para aumentar la zona activa y no son necesarios.

### Clips de fondo y primer plano

Los clips de película `chrome_mc` y `forwardBackBorder_mc` se implementan como clips de fondo.

De los clips de película `ForwardBackBorder`, `ForwardBorder` y `BackBorder` situados en el escenario y los botones de marcador de posición para avanzar y retroceder, el único que *no* es una capa de guía es `ForwardBackBorder`. Sólo se encuentra en los aspectos que utilizan realmente los botones para avanzar y retroceder.

El único requisito para estos clips es que necesitan exportarse para Actionscript en el fotograma 1 en la biblioteca.

### Modificación del comportamiento del aspecto

La propiedad `bufferingBarHidesAndDisablesOthers` y la propiedad `skinAutoHide` permiten personalizar el comportamiento del aspecto de FLVPlayback.

Si se establece la propiedad `bufferingBarHidesAndDisablesOthers` en `true`, el componente FLVPlayback oculta SeekBar y su selector, y además desactiva los botones de reproducción y pausa cuando el componente pasa al estado de almacenamiento en búfer. Esto puede ser útil cuando se transmite un archivo FLV de FMS a través de una conexión lenta con un valor alto establecido en la propiedad `bufferTime` (10, por ejemplo). En esta situación, un usuario impaciente podría intentar iniciar la búsqueda haciendo clic en los botones de reproducción y pausa, lo que podría demorar incluso más la reproducción del archivo. Para evitarlo, puede establecer `bufferingBarHidesAndDisablesOthers` en `true` y desactivar el elemento SeekBar y los botones de reproducción y pausa mientras el componente está en el estado de almacenamiento en búfer.

La propiedad `skinAutoHide` sólo afecta a archivos SWF de aspectos prediseñados y no a los controles creados desde componentes de interfaz de usuario personalizados de reproducción de FLV. Si se establece en `true`, el componente FLVPlayback oculta el aspecto cuando el puntero del ratón no está sobre el área de visualización. El valor predeterminado de esta propiedad es `false`.

## Uso de un archivo SMIL

Para gestionar varios flujos en varios anchos de banda, la clase `VideoPlayer` utiliza una clase auxiliar (`NCManager`) que admite un subconjunto de SMIL. SMIL se utiliza para identificar la ubicación del flujo de vídeo, el diseño (anchura y altura) del archivo FLV y los archivos FLV de origen correspondientes a los distintos anchos de banda. También se puede utilizar para especificar la velocidad de transferencia y la duración del archivo FLV.

Utilice el parámetro `source` o la propiedad `FLVPlayback.source` (ActionScript) para especificar la ubicación de un archivo SMIL. Para obtener más información, consulte la propiedad `FLVPlayback.source` en la [Referencia de ActionScript 3.0 para Flash Professional](#).

En el siguiente ejemplo, se muestra un archivo SMIL que transmite archivos FLV de distintos anchos de banda desde un servidor FMS, mediante 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>
```

La etiqueta `<head>` puede contener las etiquetas `<meta>` y `<layout>`. La etiqueta `<meta>` sólo admite el atributo `base`, que se utiliza para especificar la URL del vídeo de flujo (RTMP desde un servidor FMS).

La etiqueta `<layout>` sólo admite el elemento `root-layout`, que se utiliza para establecer los atributos `height` y `width`; por lo tanto, determina el tamaño de la ventana en la que se representa el archivo FLV. Estos atributos sólo aceptan valores de píxeles, no porcentajes.

En el cuerpo del archivo SMIL, se puede incluir un único vínculo a un archivo FLV de origen o, si se transmiten varios archivos en distintos anchos de banda desde un servidor FMS (como en el ejemplo anterior), se puede utilizar la etiqueta `<switch>` para enumerar los archivos de origen.

Las etiquetas `video` y `ref` contenidas en la etiqueta `<switch>` son sinónimas: ambas pueden utilizar el atributo `src` para especificar archivos FLV. Además, cada una de ellas puede utilizar los atributos `region`, `system-bitrate` y `dur` para especificar la región, el ancho de banda mínimo requerido y la duración del archivo FLV.

En la etiqueta `<body>`, sólo se permite que las etiquetas `<video>``<src>` o `<switch>` aparezcan una vez.

En el siguiente ejemplo, se muestra una descarga progresiva de un solo archivo FLV que no utiliza la detección de ancho de banda:

```
<smil>
    <head>
        <layout>
            <root-layout width="240" height="180" />
        </layout>
    </head>
    <body>
        <video src="myvideo.flv" />
    </body>
</smil>
```

## <smil>

### Disponibilidad

Flash Professional 8

### Uso

```
<smil>
...
child tags
...
</smil>
```

### Atributos

Ninguno

### Etiquetas secundarias

<head>, <body>

### Etiqueta principal

Ninguno

### Descripción

Etiqueta de nivel superior, que identifica un archivo SMIL.

### Ejemplo

En el siguiente ejemplo, se muestra un archivo SMIL que especifica tres archivos FLV:

```
<smil>
<head>
<meta base="rtmp://myserver/myapp/" />
<layout>
<root-layout width="240" height="180" />
</layout>
</head>
<body>
<switch>
    <ref src="myvideo_cable.flv" dur="3:00.1"/>
    <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
    <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1"/>
</switch>
</body>
</smil>
```

### <head>

### Disponibilidad

Flash Professional 8

### Uso

```
<head>
...
child tags
...
</head>
```

### Atributos

Ninguno

### Etiquetas secundarias

<meta>, <layout>

### Etiqueta principal

<smil>

### Descripción

Admite las etiquetas <meta> y <layout>, y especifica la ubicación y el diseño predeterminado (altura y anchura) de los archivos FLV de origen.

### Ejemplo

En el siguiente ejemplo, se establece el diseño raíz en 240 por 180 píxeles:

```
<head>
  <meta base="rtmp://myserver/myapp/" />
  <layout>
    <root-layout width="240" height="180" />
  </layout>
</head>
```

## <meta>

### Disponibilidad

Flash Professional 8

### Uso

```
<meta/>
```

### Atributos

base

### Etiquetas secundarias

<layout>

### Etiqueta principal

Ninguno

### Descripción

Contiene el atributo `base`, que especifica la ubicación (URL RTMP) de los archivos FLV de origen.

### Ejemplo

En el siguiente ejemplo, se muestra una etiqueta `meta` de una ubicación de base en `myserver`:

```
<meta base="rtmp://myserver/myapp/" />
```

## <layout>

### Disponibilidad

Flash Professional 8

### Uso

```
<layout>  
...  
child tags  
...  
</layout>
```

### Atributos

Ninguno

### Etiquetas secundarias

```
<root-layout>
```

### Etiqueta principal

```
<meta>
```

### Descripción

Especifica la anchura y la altura del archivo FLV.

### Ejemplo

El siguiente ejemplo establece el diseño en 240 píxeles por 180 píxeles:

```
<layout>  
  <root-layout width="240" height="180" />  
</layout>
```

## <root-layout>

### Disponibilidad

Flash Professional 8

### Uso

```
<root-layout...attributes.../>
```

### Atributos

Anchura, altura

### Etiquetas secundarias

Ninguna

### Etiqueta principal

```
<layout>
```

### Descripción

Especifica la anchura y la altura del archivo FLV.

### Ejemplo

El siguiente ejemplo establece el diseño en 240 píxeles por 180 píxeles:

```
<root-layout width="240" height="180" />
```

## <body>

### Disponibilidad

Flash Professional 8

### Uso

```
<body>  
...  
child tags  
...  
</body>
```

### Atributos

Ninguno

### Etiquetas secundarias

<video>, <ref>, <switch>

### Etiqueta principal

<smil>

### Descripción

Contiene las etiquetas <video>, <ref> y <switch>, que especifican el nombre del archivo FLV de origen, el ancho de banda mínimo y la duración del archivo FLV. El atributo `system-bitrate` sólo se admite cuando se utiliza la etiqueta <switch>. En la etiqueta <body>, sólo se permite que las etiquetas <switch>, <video> o <ref> aparezcan una vez.

### Ejemplo

El siguiente ejemplo especifica tres archivos FLV, dos de ellos con la etiqueta `video` y el otro con la etiqueta `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>

### Disponibilidad

Flash Professional 8

### Uso

```
<video...attributes.../>
```

### Atributos

src, system-bitrate, dur

### Etiquetas secundarias

Ninguna

### Etiqueta principal

```
<body>
```

### Descripción

Sinónimo de la etiqueta `<ref>`. Admite los atributos `src` y `dur`, que especifican el nombre del archivo FLV de origen y su duración. El atributo `dur` admite los formatos de tiempo completo (00:03:00:01) y parcial (03:00:01).

### Ejemplo

En el siguiente ejemplo, se establecen el origen y la duración de un vídeo:

```
<video src="myvideo_mdm.flv" dur="3:00.1"/>
```

## <ref>

### Disponibilidad

Flash Professional 8

### Uso

```
<ref...attributes.../>
```

### Atributos

src, system-bitrate, dur

### Etiquetas secundarias

Ninguna

### Etiqueta principal

```
<body>
```

### Descripción

Sinónimo de la etiqueta `<video>`. Admite los atributos `src` y `dur`, que especifican el nombre del archivo FLV de origen y su duración. El atributo `dur` admite los formatos de tiempo completo (00:03:00:01) y parcial (03:00:01).

### Ejemplo

En el siguiente ejemplo, se establecen el origen y la duración de un vídeo:

```
<ref src="myvideo_cable.flv" dur="3:00.1"/>
```

## <switch>

### Disponibilidad

Flash Professional 8

### Uso

```
<switch>
...
child tags
...
</switch/>
```

### Atributos

Ninguno

### Etiquetas secundarias

<video>, <ref>

### Etiqueta principal

<body>

### Descripción

Se utiliza con las propiedades secundarias <video> o <ref> para enumerar los archivos FLV de flujo de vídeo transmitido en varios anchos de banda. La etiqueta <switch> admite el atributo `system-bitrate`, que especifica el ancho de banda mínimo, y los atributos `src` y `dur`.

### Ejemplo

El siguiente ejemplo especifica tres archivos FLV, dos de ellos con la etiqueta `video` y el otro con la etiqueta `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 del componente FLVPlaybackCaptioning

El componente FLVPlayback permite incluir un reproductor de vídeo en la aplicación de Adobe Flash CS5 Professional para reproducir archivos descargados de Adobe Flash Video (FLV o F4V) y archivos FLV o F4V sin interrupción. Para obtener más información sobre FLVPlayback, consulte “Uso del componente FLVPlayback” en la página 140.

El componente FLVPlaybackCaptioning permite incluir subtitulación oculta en el vídeo. El componente de subtitulación es compatible con el formato XML de texto temporizado aprobado por W3C e incluye estas funciones:

**Subtitulación con puntos de referencia de eventos integrados** Asocie puntos de referencia de eventos incorporados en un archivo FLV con XML para proporcionar subtítulos, en lugar de utilizar un archivo XML de texto temporizado.

**Varios componentes FLVPlaybackCaptioning** Cree varias instancias de FLVPlaybackCaptioning para varias instancias de FLVPlayback.

**Control mediante un conmutador** Permita la interacción del usuario con la subtitulación a través de un conmutador de subtitulación.

## Uso del componente FLVPlaybackCaptioning

El componente FLVPlaybackCaptioning se utiliza con uno o varios componentes FLVPlayback. En la situación más sencilla, se arrastra un componente FLVPlayback al escenario, se arrastra un componente FLVPlaybackCaptioning al mismo escenario, se identifican los URL de texto y se establecen los textos para mostrar. Además, se pueden establecer diversos parámetros para personalizar la subtitulación de FLVPlayback.

### Añadir subtitulación al componente FLVPlayback

Es posible añadir el componente FLVPlaybackCaptioning a cualquier componente FLVPlayback. Para obtener información más sobre la adición de componentes FLVPlayback a la aplicación, consulte “Creación de aplicaciones con el componente FLVPlayback” en la página 142.

#### Añadir el componente FLVPlaybackCaptioning desde el panel Componentes:

- 1 En el panel Componentes, abra la carpeta Vídeo.
- 2 Arrastre el componente FLVPlaybackCaptioning (o haga doble clic en él) y añádalo al mismo escenario donde está el componente FLVPlayback al que desea añadir subtitulación.

*Nota:* Adobe proporciona dos archivos que ayudan a un rápido aprendizaje del componente FLVPlaybackCaptioning: *caption\_video.flv* (un ejemplo de FLVPlayback) y *caption\_video.xml* (un ejemplo de subtitulación). Puede acceder a estos archivos en [www.helpexamples.com/flash/video/caption\\_video.flv](http://www.helpexamples.com/flash/video/caption_video.flv) y [www.helpexamples.com/flash/video/caption\\_video.xml](http://www.helpexamples.com/flash/video/caption_video.xml).

- 3 (Opcional) Arrastre el componente CaptionButton al mismo escenario donde están los componentes FLVPlayback y FLVPlaybackCaptioning. El componente CaptionButton permite al usuario activar y desactivar la subtitulación.

*Nota:* para activar el componente CaptionButton, debe arrastrarlo al mismo escenario donde están los componentes FLVPlayback y FLVPlaybackCaptioning.

4 Con el componente FLVPlaybackCaptioning seleccionado en el escenario, en la ficha Parámetros del inspector de propiedades, especifique la siguiente información requerida:

- Establezca `showCaptions` en `true`.
- Especifique en `source` el origen del archivo XML de texto temporizado que se va a descargar.

 *Mientras prueba los textos en Flash, debería establecer la propiedad `showCaptions` en `true`. Sin embargo, si incluye el componente `CaptionButton` para permitir a los usuarios activar y desactivar la subtítulos, debería establecer la propiedad `showCaptions` en `false`.*

Hay otros parámetros disponibles para personalizar el componente FLVPlaybackCaptioning. Para obtener más información, consulte [“Personalización del componente FLVPlaybackCaptioning”](#) en la página 190 y la *Referencia de ActionScript 3.0 para Flash Professional*.

5 Seleccione Control > Probar película para iniciar el vídeo.

### Creación dinámica de una instancia mediante código ActionScript:

- 1 Arrastre el componente FLVPlayback del panel Componentes al panel Biblioteca (Ventana > Biblioteca).
- 2 Arrastre el componente FLVPlaybackCaptioning del panel Componentes al panel Biblioteca.
- 3 Añada el siguiente código al panel Acciones, en el fotograma 1 de la línea de tiempo.

```
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 Cambie `unidad_instalación` por la unidad en la que instaló Flash y modifique la ruta de acceso para reflejar la ubicación de la carpeta Skins en la instalación:

**Nota:** *si se crea una instancia de FLVPlayback con ActionScript, también se le debe asignar un aspecto dinámicamente, estableciendo la propiedad `skin` mediante código ActionScript. Cuando se aplica un aspecto con ActionScript, no se publica automáticamente con el archivo SWF. Es necesario copiar al servidor el archivo SWF de aspecto y el archivo SWF de aplicación ya que, de lo contrario, el archivo SWF de aspecto no estará disponible cuando un usuario lo ejecute.*

## Configuración de los parámetros del componente FLVPlaybackCaptioning

Para cada instancia del componente FLVPlaybackCaptioning, se pueden establecer los siguientes parámetros en el inspector de propiedades o en el inspector de componentes, con el fin de personalizar el componente. En la siguiente lista se identifican las propiedades y se ofrece una breve explicación de las mismas:

**autoLayout** Determina si el componente FLVPlaybackCaptioning controla el tamaño del área de subtítulos. El valor predeterminado es `true`.

**captionTargetName** Identifica el nombre de instancia de TextField o MovieClip que contiene texto. El valor predeterminado es `auto`.

**flvPlaybackName** Identifica el nombre de instancia de FLVPlayback donde se desea incluir texto. El valor predeterminado es `auto`.

**simpleFormatting** Limita las instrucciones de formato del archivo XML de texto temporizado cuando se establece en `true`. El valor predeterminado es `false`.

**showCaptions** Determina si se muestra el texto. El valor predeterminado es `true`.

**source** Identifica la ubicación del archivo XML de texto temporizado.

Para obtener más información sobre todos los parámetros de FLVPlaybackCaptioning, consulte la *Referencia de ActionScript 3.0 para Flash Professional*.

### Especificar el parámetro source

El parámetro `source` se utiliza para especificar el nombre y la ubicación del archivo XML de texto temporizado que contiene los textos de la película. La ruta del URL se introduce directamente en la celda de origen del inspector de componentes.

### Mostrar subtítulos

Para ver la subtitulación, es necesario establecer el parámetro `showCaptions` en `true`.

Para obtener más información sobre todos los parámetros del componente FLVPlaybackCaptioning, consulte la *Referencia de ActionScript 3.0 para Flash Professional*.

En los ejemplos anteriores se mostraba cómo crear y activar el componente FLVPlaybackCaptioning para mostrar textos. Se pueden utilizar dos fuentes para los subtítulos: 1) un archivo XML de texto temporizado que contiene los subtítulos o (2) un archivo XML con el texto subtulado que se asocia con puntos de referencia de eventos incorporados.

## Uso de texto temporizado

El componente FLVPlaybackCaptioning permite subtítular el componente FLVPlayback asociado, mediante la descarga de un archivo XML de texto temporizado (TT). Para obtener más información sobre el formato de texto temporizado, consulte la información sobre AudioVideo Timed Text en [www.w3.org](http://www.w3.org).

En esta sección se ofrece información general sobre las etiquetas de texto temporizado compatibles y las etiquetas de archivo de subtitulación necesarias, así como un ejemplo de un archivo XML de texto temporizado. Para obtener información detallada sobre las etiquetas de texto temporizado compatibles, consulte “[Etiquetas de texto temporizado](#)” en la página 183.

El componente FLVPlaybackCaptioning admite las siguientes etiquetas de texto temporizado:

Categoría	Tarea
Formato de párrafo	Alinee un párrafo a la derecha, a la izquierda o al centro
Formato de texto	<ul style="list-style-type: none"><li>• Defina el tamaño del texto con tamaños de píxeles absolutos o estilo delta (por ejemplo, +2, -4)</li><li>• Defina el color del texto y la fuente</li><li>• Aplique negrita y cursiva al texto</li><li>• Defina la justificación del texto</li></ul>
Otro formato	<ul style="list-style-type: none"><li>• Defina el color de fondo de TextField para los textos</li><li>• Defina el color de fondo de TextField para los textos como transparente (alfa 0)</li><li>• Defina el ajuste de texto de TextField para los textos (activado o desactivado)</li></ul>

El componente FLVPlaybackCaptioning tiene el mismo código de tiempo que el archivo FLV. Cada texto debe tener un atributo `begin`, que determina cuándo debe aparecer el texto. Si el texto no tiene un atributo `dur` o `end`, desaparecerá cuando aparezca el siguiente texto o cuando finalice el archivo FLV.

A continuación se muestra un ejemplo de un archivo XML de texto temporizado. Este archivo (`caption_video.xml`) proporciona subtítulos al archivo `caption_video.flv`. Puede acceder a estos archivos en [www.helpexamples.com/flash/video/caption\\_video.flv](http://www.helpexamples.com/flash/video/caption_video.flv) y [www.helpexamples.com/flash/video/caption\\_video.xml](http://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>
```

## Etiquetas de texto temporizado

El componente FLVPlaybackCaptioning admite la utilización de etiquetas de texto temporizado para la subtítulos de archivos XML. Para obtener más información sobre las etiquetas de texto temporizado para audio y vídeo, consulte la información correspondiente en [www.w3.org](http://www.w3.org). La siguiente tabla contiene una lista de etiquetas que se admiten y etiquetas que no se admiten.

Función	Etiqueta/Valor	Uso/Descripción	Ejemplo
Etiquetas que se omiten	metadata	Se omite / Se permite en cualquier nivel del documento.	
	set	Se omite / Se permite en cualquier nivel del documento.	
	xml:lang	Se omite.	
	xml:space	Se omite / El comportamiento se sustituye por: xml:space="default".	

<b>Función</b>	<b>Etiqueta/Valor</b>	<b>Uso/Descripción</b>	<b>Ejemplo</b>
	layout	Se omite / También se incluyen las etiquetas región en una sección de la etiqueta layout.	
	etiqueta br	Se omiten todos los atributos y el contenido.	
Tiempo de medios para textos	atributos begin	Sólo se permiten en etiquetas p. Necesarios para implementar el tiempo de medios para textos.	<p begin="3s">
	atributos dur	Sólo se permiten en etiquetas p. Uso recomendado. Si no se incluyen, el texto finaliza con el archivo FLV o si no se inicia otro texto.	
	atributos end	Sólo se permiten en etiquetas p. Recomendado. Si no se incluye, el texto termina con el archivo FLV o cuando se inicia otro texto.	
Tiempo de reloj para textos	00:03:00.1	Formato de reloj completo.	
	03:00.1	Formato de reloj parcial.	
	10	Tiempos de desplazamiento sin unidades. El desplazamiento se representa en segundos.	
	00:03:00:05 00:03:00:05.1 30f 30t	No se admite. Los formatos de tiempo que incluyen fotogramas o marcas no se admiten.	
Etiqueta Body	body	Necesaria / Sólo se admite una etiqueta body.	<body><><div>...</div></body>>
Etiqueta Content	etiqueta div	Se pueden usar entre cero y varias. Se usa la primera etiqueta.	
	etiqueta p	Se pueden usar entre cero y varias.	
	etiqueta span	Contenedor lógico de una secuencia de unidades de contenido de texto. No se admiten etiquetas span anidadas. Se admiten etiquetas de estilo de atributo.	
	etiqueta br	Indica un salto de línea explícito.	
Etiquetas de estilo  (todas ellas se utilizan en la etiqueta p)	style	Referencia a uno o varios elementos de estilo. Puede usarse como etiqueta y como atributo. Como etiqueta, se requiere un atributo ID (style puede reutilizarse en el documento). Se admiten una o varias etiquetas style en una etiqueta style.	
	tts:background Color	Especifica una propiedad de estilo que define el color de fondo de una zona. El valor alfa se omite a menos que sea cero (alfa 0) para que el fondo sea transparente. El formato de color es #RRGGBBAA.	

Función	Etiqueta/Valor	Uso/Descripción	Ejemplo
	tts:color	Especifica una propiedad de estilo que define el color de primer plano. No se admite alfa en ningún color. El valor <code>transparent</code> da como resultado negro.	<pre>&lt;style id="3" style="2" tts:backgroundColor="white"/&gt; "transparent" = #00000000 "black"=#000000FF "silver"=#C0C0C0FF "grey"=#808080FF "white"=#FFFFFFFF "maroon"=#800000FF "red"=#FF0000FF "purple"=#800080FF "fuchsia"("magenta")= #FF00FFFF "green"=#008000FF "lime"=#00FF00FF "olive"=#808000FF "yellow"=#FFFF00FF "navy"=#000080FF "blue"=#0000FFFF "teal"=#008080FF "aqua"("cyan")=#00FFFFFF</pre>
	tts:fontFamily	Especifica una propiedad de estilo que define la familia de fuentes.	<pre>"default" = _serif "monospace" = _typewriter "sansSerif" = _sans "serif" = _serif "monospaceSansSerif" = _typewriter "monospaceSerif" = _typewriter "proportionalSansSerif" = _sans</pre>
	tts:fontSize	Especifica una propiedad de estilo que define el tamaño de fuente. Sólo se usa el primer valor (vertical) si se proporcionan dos. Se omiten los valores de porcentaje y las unidades. Se admiten los tamaños de píxeles absolutos (p. ej., 12) y estilo relativo (p. ej., +2).	
	tts:fontStyle	Especifica una propiedad de estilo que define el estilo de fuente.	<pre>"normal" "italic" "inherit"* * Valor predeterminado; hereda el estilo de la etiqueta donde se incluye.</pre>

Función	Etiqueta/Valor	Uso/Descripción	Ejemplo
	tts:fontWeight	Especifica una propiedad de estilo que define el grosor de fuente.	"normal" "bold" "inherit"* * Valor predeterminado; hereda el estilo de la etiqueta donde se incluye.
	tts:textAlign	Especifica una propiedad de estilo que define la alineación de las áreas en línea con respecto al área del bloque contenedor.	"left" "right" "center" "start" ("left") "end" ("right") "inherit"* *Hereda el estilo de la etiqueta donde se incluye. Si no se establece la etiqueta textAlign, el valor predeterminado es "left".
	tts:wrapOption	Especifica una propiedad de estilo que define si se aplica o no el ajuste de líneas automático en el contexto del elemento afectado. Esta configuración afecta a todos los párrafos del elemento del texto.	"wrap" "noWrap" "inherit"* *Hereda el estilo de la etiqueta donde se incluye. Si no se establece la etiqueta wrapOption, el valor predeterminado es "wrap".
Atributos que no se admiten	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 puntos de referencia con ActionScript

Los puntos de referencia permiten interactuar con un vídeo; por ejemplo, es posible alterar la reproducción de un archivo FLV o mostrar texto en momentos concretos del vídeo. Si no se dispone de un archivo XML de texto temporizado para utilizarlo con un archivo FLV, es posible incorporar puntos de referencia de eventos en un archivo FLV y asociar dichos puntos de referencia al texto. En esta sección se ofrece información sobre las normas de puntos de referencia del componente FLVPlaybackCaptioning, así como una breve información general sobre cómo asociar estos puntos de referencia al texto en la subtítulos. Para más información sobre cómo incorporar puntos de referencia con el asistente de importación de vídeo o Flash Video Encoder, consulte el Capítulo 16, "Trabajo con vídeo", en *Uso de Flash*.

### Aspectos básicos de las normas de puntos de referencia de FLVPlaybackCaptioning

En los metadatos del archivo FLV, un punto de referencia se representa como un objeto con las siguientes propiedades: `name`, `time`, `type` y `parameters`. Los puntos de referencia de ActionScript para FLVPlaybackCaptioning tienen los siguientes atributos:

**name** La propiedad `name` es una cadena que contiene el nombre asignado del punto de referencia. La propiedad `name` debe comenzar con el prefijo `fl.video.caption.2.0.` y contener una cadena a continuación. La cadena es una serie de enteros positivos que van incrementando para que cada nombre sea exclusivo. El prefijo contiene el número de versión, que coincide con el número de versión del componente FLVPlayback. En Adobe Flash CS4 y posterior, debe establecerse el número de versión `2.0`.

**time** La propiedad `time` representa el tiempo en el que debe mostrarse el texto.

**type** La propiedad `type` es una cadena cuyo valor es `"event"`.

**parameters** La propiedad `parameters` es un conjunto que admite los siguientes pares nombre-valor:

- **text:String** El texto del subtítulo con formato HTML. Este texto se pasa directamente a la propiedad `TextField.htmlText`. El componente FLVPlaybackCaptioning admite una propiedad `text:n` opcional, que admite el uso de varias pistas de lenguaje. Para más información, consulte ["Uso de varias pistas de idioma con puntos de referencia incorporados"](#) en la página 189.
- **endTime:Number** El tiempo en el que debe desaparecer el texto. Si no se especifica esta propiedad, el componente FLVPlaybackCaptioning presupone que no es un número (NaN) y se muestra un texto hasta que finaliza el archivo FLV (la instancia de FLVPlayback distribuye el evento `VideoEvent.COMPLETE`). Especifique la propiedad `endTime:Number` en segundos.
- **backgroundColor:uint** Este parámetro establece `TextField.backgroundColor`. Esta propiedad es opcional.
- **backgroundColorAlpha:Boolean** Si el valor de alfa de `backgroundColor` es 0%, el parámetro establece `TextField.background = !backgroundColor`. Esta propiedad es opcional.
- **wrapOption:Boolean** Este parámetro establece `TextField.wordWrap`. Esta propiedad es opcional.

## Aspectos básicos de la subtitulación para puntos de referencia de eventos incorporados

Si no se dispone de un archivo XML de texto temporizado que contenga textos para el archivo FLV, es posible asociar el archivo XML que contiene la subtitulación con puntos de referencia de eventos incorporados con el fin de crear subtitulación. En la muestra de XML se presupone que se han realizado los siguientes pasos para crear puntos de referencia de eventos incorporados en el vídeo:

- Añada los puntos de referencia de eventos (según las normas de FLVPlaybackCaptioning) y codifique el vídeo.
- En Flash, arrastre un componente FLVPlayback y un componente FLVPlaybackCaptioning al escenario.
- Defina las propiedades de origen de los componentes FLVPlayback y FLVPlaybackCaptioning (es decir, la ubicación del archivo FLV y la ubicación del archivo XML).
- Realice la publicación.

En el siguiente ejemplo se importa XML en Encoder:

```
<?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>
```

El componente FLVPlaybackCaptioning también admite el uso de varias pistas de idioma con puntos de referencia incorporados. Para más información, consulte [“Uso de varias pistas de idioma con puntos de referencia incorporados”](#) en la página 189.

## Uso de varias pistas de idioma con puntos de referencia incorporados

La propiedad `track` de FLVPlaybackCaptioning admite varias pistas de idioma con puntos de referencia incorporados, siempre y cuando el archivo XML de texto temporizado siga las normas de puntos de referencia de FLVPlaybackCaptioning. (Para más información, consulte [“Aspectos básicos de las normas de puntos de referencia de FLVPlaybackCaptioning”](#) en la página 187). Sin embargo, el componente FLVPlaybackCaptioning no admite varias pistas de idioma en distintos archivos XML. Para utilizar la propiedad `track`, establezca en ella un valor que no sea 0. Por ejemplo, si establece la propiedad `track` en 1 (`track == 1`), el componente FLVPlaybackCaptioning buscará los parámetros de puntos de referencia. Si no se encuentra ninguna coincidencia, se utiliza la propiedad `text` en los parámetros de puntos de referencia. Para obtener más información, consulte la propiedad `track` en la *Referencia de ActionScript 3.0 para Flash Professional*.

## Reproducción de varios archivos FLV con subtitulación

Es posible abrir varios reproductores de vídeo en una sola instancia del componente FLVPlayback para reproducir varios vídeos y pasar de uno a otro durante la reproducción. También se puede asociar subtitulación a cada reproductor de vídeo en el componente FLVPlayback. Para más información sobre cómo utilizar varios reproductores de vídeo, consulte [“Uso de múltiples reproductores de vídeo”](#) en la página 156. Para utilizar subtitulación en varios reproductores de vídeo, debe crearse una instancia del componente FLVPlaybackCaptioning para cada `VideoPlayer` y establecer `videoPlayerIndex` de FLVPlaybackCaptioning en el índice correspondiente. El valor predeterminado del índice de `VideoPlayer` es 0 cuando sólo existe un `VideoPlayer`.

A continuación se muestra un ejemplo de código que asigna subtitulación exclusiva a vídeos exclusivos. Para ejecutar este ejemplo, los URL ficticios deben reemplazarse por URL reales.

```
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");
```

## Personalización del componente FLVPlaybackCaptioning

Para empezar a utilizar rápidamente el componente FLVPlaybackCaptioning, se puede optar por utilizar los valores predeterminados de FLVPlaybackCaptioning, que colocan la subtítulos directamente sobre el componente FLVPlayback. Es posible personalizar el componente FLVPlaybackCaptioning para desactivar la subtítulos del vídeo.

El código siguiente muestra cómo crear dinámicamente un objeto FLVPlayback con el conmutador de subtítulos:

- 1 Coloque el componente FLVPlayback en el escenario en la posición 0,0 y proporcione el nombre de instancia **player**.
- 2 Coloque el componente FLVPlaybackCaptioning en el escenario en la posición 0,0 y asígnele el nombre de instancia **captioning**.
- 3 Coloque el componente CaptionButton en el escenario.
- 4 En el siguiente ejemplo de código, establezca la variable `testVideoPath:String` en un archivo FLV (mediante una ruta relativa o absoluta).

***Nota:** el ejemplo de código establece la variable `testVideoPath` en una muestra de vídeo Flash, `caption_video.flv`. Cambie esta variable por la ruta del componente de vídeo de subtítulos al que está añadiendo un componente de botón de texto.*

- 5 En el siguiente ejemplo de código, establezca la variable `testCaptioningPath:String` en un archivo XML de texto temporizado adecuado (utilizando una ruta relativa o absoluta).

***Nota:** el ejemplo de código establece la variable `testCaptioningPath` en el archivo XML de texto temporizado `caption_video.xml`. Cambie esta variable por la ruta del archivo XML de texto temporizado que contiene textos para el vídeo.*

- 6 Guarde el siguiente código como FLVPlaybackCaptioningExample.as en el mismo directorio que el archivo FLA.
- 7 Establezca DocumentClass del archivo FLA en 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 obtener más información sobre todos los parámetros de FLVPlaybackCaptioning, consulte la *Referencia de ActionScript 3.0 para Flash Professional*.