

Uso dei componenti ACTIONSCRIPT® 3.0

Note legali

Per le note legali, vedete http://help.adobe.com/it_IT/legalnotices/index.html.

Sommario

Capitolo 1: Introduzione

A chi è destinato questo manuale	1
Requisiti di sistema	1
Informazioni sulla documentazione	2
Convenzioni tipografiche	2
Termini utilizzati in questo manuale	2
Altro materiale di riferimento	2

Capitolo 2: Informazioni sui componenti ActionScript 3.0

Vantaggi dell'uso dei componenti	4
Tipi di componente	6
Aggiungere ed eliminare componenti da un documento	7
Individuare la versione del componente	9
Modello di gestione degli eventi ActionScript 3.0	10
Una semplice applicazione	10

Capitolo 3: Operazioni con i componenti

Architettura dei componenti	18
Operazioni con i file dei componenti	20
Effettuare il debug delle applicazioni di componenti	22
Impostare parametri e proprietà	22
La libreria	24
Ridimensionamento dei componenti	24
Anteprima dal vivo	24
Gestione degli eventi	25
Operazioni con l'elenco di visualizzazione	26
Operazioni con FocusManager	29
Operazioni con i componenti basati su List	30
Operazioni con un componente DataProvider	31
Operazioni con CellRenderer	39
Rendere accessibili i componenti	46

Capitolo 4: Uso dei componenti UI

Usare il componente Button	47
Usare il componente CheckBox	49
Usare il componente ColorPicker	52
Usare il componente ComboBox	55
Usare il componente DataGrid	58
Usare il componente Label	64
Usare il componente List	66
Usare il componente NumericStepper	71
Usare il componente ProgressBar	74
Usare il componente RadioButton	79

Sommario

Usare il componente ScrollPane	82
Usare il componente Slider	85
Usare il componente TextArea	88
Usare il componente TextInput	91
Usare il componente TileList	94
Usare il componente UILoader	97
Usare il componente UIScrollBar	99

Capitolo 5: Personalizzazione dei componenti UI

Personalizzazione dei componenti dell'interfaccia utente	102
Impostazione di stili	102
Informazioni sugli skin	105
Personalizzare il componente Button	108
Personalizzare il componente CheckBox	110
Personalizzare il componente ColorPicker	111
Personalizzare il componente ComboBox	113
Personalizzare il componente DataGrid	115
Personalizzare il componente Label	120
Personalizzare il componente List	121
Personalizzare il componente NumericStepper	124
Personalizzare il componente ProgressBar	125
Personalizzare il componente RadioButton	127
Personalizzare il componente ScrollPane	129
Personalizzare il componente Slider	130
Personalizzare il componente TextArea	131
Personalizzare il componente TextInput	133
Personalizzare il componente TileList	135
Personalizzare il componente UILoader	137
Personalizzare il componente UIScrollBar	137

Capitolo 6: Uso del componente FLVPlayback

Usare il componente FLVPlayback	140
Personalizzare il componente FLVPlayback	159
Usare un file SMIL	170

Capitolo 7: Uso del componente FLVPlaybackCaptioning

Usare il componente FLVPlaybackCaptioning	178
Usare le didascalie Timed Text	180
Usare i cue point con le didascalie	186
Riprodurre più file FLV con didascalie	188
Personalizzare il componente FLVPlaybackCaptioning	189

Capitolo 1: Introduzione

Adobe® Flash® CS5 è lo strumento professionale standard per la creazione di codice in grado di offrire esperienze Web di grande impatto. I componenti sono gli elementi strutturali di base delle applicazioni Internet dinamiche che assicurano tali esperienze. Un *componente* è un clip filmato con parametri che ne consentono la personalizzazione in fase di creazione in Flash o in fase di runtime con metodi, proprietà ed eventi Adobe® ActionScript®. I componenti sono concepiti per consentire agli sviluppatori di riutilizzare e condividere il codice, nonché di incorporare funzionalità complesse che i designer possono usare e personalizzare senza avvalersi di ActionScript.

I componenti consentono di creare in modo facile e rapido applicazioni complesse con un aspetto e un comportamento omogenei. In questo manuale viene illustrato come creare applicazioni con componenti Adobe ActionScript 3.0. Nella *Guida di riferimento del linguaggio e dei componenti Adobe® ActionScript® 3.0* è illustrata l'API (Application Programming Interface) di ogni componente.

Potete usare i componenti creati da Adobe®, scaricare i componenti realizzati da altri sviluppatori oppure ideare componenti personalizzati.

A chi è destinato questo manuale

Il manuale è rivolto agli sviluppatori che creano applicazioni Flash e desiderano utilizzare i componenti per accelerare la fase di sviluppo. È necessario disporre di una buona conoscenza dello sviluppo di applicazioni in Flash e della creazione di script ActionScript.

Se non si è molto esperti nella creazione di script ActionScript, è possibile aggiungere i componenti a un documento, impostare i relativi parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti e gestirne gli eventi nel pannello Comportamenti. Ad esempio, è possibile associare un comportamento Vai alla pagina Web a un componente Button che apre un URL in un browser Web quando si fa clic sul pulsante senza scrivere alcun codice ActionScript.

Se invece desiderate creare applicazioni più complesse, potete creare i componenti in modo dinamico, usare ActionScript per impostare le proprietà e chiamare i metodi in fase di runtime o utilizzare il modello di listener per gestire gli eventi.

Per ulteriori informazioni, vedete “[Operazioni con i componenti](#)” a pagina 18.

Requisiti di sistema

Per i componenti Flash non sono richiesti altri requisiti di sistema oltre a quelli definiti per Flash.

Eventuali file SWF che utilizzano i componenti Flash CS3 o successivi devono essere visualizzati con Adobe® Flash® Player 9.0.28.0 o una versione successiva e devono essere pubblicati per ActionScript 3.0 (a questo scopo selezionate File > Impostazioni di pubblicazione e scegliete la scheda Flash).

Informazioni sulla documentazione

In questa pubblicazione è illustrato in dettaglio l'uso dei componenti per sviluppare le applicazioni Flash. Dovete disporre di un'adeguata conoscenza di Adobe Flash e ActionScript 3.0. La documentazione specifica su Flash e i prodotti correlati è disponibile separatamente.

Questo documento è disponibile sotto forma di file PDF e di Guida in linea. Per visualizzare la Guida in linea, avviate Flash e selezionate ? > Guida di Flash > Uso dei componenti Adobe ActionScript 3.0.

Per informazioni su Flash, consultare i seguenti documenti:

- *Uso di Flash*
- *Guida per sviluppatori di ActionScript 3.0*
- *Guida di riferimento di Adobe ActionScript 3.0 per Flash Professional*

Convenzioni tipografiche

In questo manuale vengono utilizzate le seguenti convenzioni tipografiche:

- *Il carattere corsivo indica un valore che deve essere sostituito dall'utente (ad esempio, il percorso di una cartella).*
- `Carattere codice` indica il codice ActionScript, inclusi nomi dei metodi e delle proprietà.
- *Carattere codice corsivo: indica un elemento di codice che deve essere sostituito, ad esempio un parametro ActionScript.*
- **Carattere grassetto: indica un valore da immettere.**

Termini utilizzati in questo manuale

In questo manuale vengono utilizzati i termini seguenti:

In fase di runtime Durante l'esecuzione del codice in Flash Player.

Durante la (fase di) creazione Nel corso delle operazioni eseguite nell'ambiente di creazione Flash.

Altro materiale di riferimento

Oltre al contenuto di questi manuali, Adobe offre articoli aggiornati periodicamente, idee per la progettazione ed esempi nel Centro per sviluppatori Adobe e nell'Adobe Design Center.

Potete trovare ulteriori esempi di componenti all'indirizzo www.adobe.com/go/learn_fl_samples_it.

Centro per sviluppatori Adobe

Nel Centro per sviluppatori Adobe sono disponibili informazioni aggiornate su ActionScript, articoli relativi allo sviluppo di applicazioni reali e informazioni su importanti problemi emergenti. Visitate il Centro per sviluppatori all'indirizzo www.adobe.com/go/flash_devcenter_it.

Adobe Design Center

Consente di ottenere le informazioni più aggiornate sulla progettazione digitale e sulle immagini in movimento. Potete accedere alle opere dei più noti artisti, scoprire le nuove tendenze di progettazione e affinare le vostre competenze con esercitazioni, flussi di lavoro di base e tecniche avanzate. Vi consigliamo di controllare un paio di volte al mese se sono disponibili esercitazioni e articoli aggiornati, nonché elementi di gallerie fotografiche da cui trarre ispirazione. Visitate il Design Center all'indirizzo www.adobe.com/go/fl_designcenter_it.

Capitolo 2: Informazioni sui componenti ActionScript 3.0

I componenti di Adobe® Flash® Professional CS5 sono clip filmato dotati di parametri che consentono di modificarne l'aspetto e il comportamento. Un componente può essere un semplice controllo dell'interfaccia utente (ad esempio, RadioButton o CheckBox) oppure può includere contenuto (ad esempio, List o DataGrid).

I componenti consentono di creare in modo facile e rapido applicazioni Flash complesse con un comportamento e un aspetto omogenei. Anziché creare pulsanti, caselle combinate ed elenchi personalizzati, potete infatti usare i componenti Flash che implementano questi controlli trascinandoli dal pannello Componenti nel documento dell'applicazione. Anche l'aspetto dei componenti può essere facilmente personalizzato in base alla progettazione dell'applicazione.

Mentre tutte queste operazioni possono essere effettuate senza una conoscenza approfondita di ActionScript, potete usare ActionScript 3.0 per modificare il comportamento di un componente o implementare un nuovo comportamento. Tutti i componenti dispongono inoltre di un set univoco di metodi, proprietà ed eventi di ActionScript, che costituiscono la relativa API (Application Programming Interface) con la quale potete creare e modificare i componenti mentre l'applicazione è in esecuzione.

Mediante le API potete creare autonomamente componenti nuovi e personalizzati. Potete scaricare componenti creati da membri della comunità Flash dal sito Adobe Exchange all'indirizzo http://www.adobe.com/go/flash_exchange_it. Per informazioni sulla creazione di un componente, vedete http://www.adobe.com/go/learn_fl_creating_components_it.

L'architettura dei componenti ActionScript 3.0 include classi sulle quali si basano tutti i componenti, gli skin e gli stili che consentono di personalizzare l'aspetto dei componenti, un modello di gestione degli eventi, la gestione dell'attivazione, un'interfaccia per l'accessibilità e altro ancora.

***Nota:** Adobe Flash CS5 include sia i componenti ActionScript 2.0 che i componenti ActionScript 3.0. Non è possibile usare insieme questi due set di componenti, ma è necessario usare alternativamente un set o l'altro per una determinata applicazione. In Flash CS5 vengono presentati componenti ActionScript 2.0 o componenti ActionScript 3.0 a seconda che venga aperto un file ActionScript 2.0 o un file ActionScript 3.0. Quando create un nuovo documento Flash, dovete specificare se si tratta di un file Flash ActionScript 3.0 o ActionScript 2.0. All'apertura di un documento esistente, vengono esaminati automaticamente i valori di Impostazioni pubblicazioni per determinare quale set di componenti usare. Per informazioni sui componenti ActionScript 2.0, vedete la guida *Uso dei componenti Adobe® ActionScript® 2.0*.*

Per un elenco completo dei componenti disponibili in Flash ActionScript 3.0, vedete “[Tipi di componente](#)” a pagina 6.

Vantaggi dell'uso dei componenti

Grazie ai componenti potete separare il processo di progettazione dell'applicazione dal processo di codifica. Essi consentono agli sviluppatori di creare funzionalità che i designer possono usare nelle applicazioni. Da una parte, gli sviluppatori possono incorporare nei componenti le funzionalità usate di frequente, dall'altra i designer possono personalizzare le dimensioni, la posizione e il comportamento dei componenti modificandone i parametri e, inoltre, cambiare l'aspetto di un componente modificandone gli elementi grafici o skin.

I componenti condividono funzionalità di base quali stili, skin e gestione dell'attivazione. Quando aggiungete il primo componente a un'applicazione, le funzionalità di base occupano circa 20 kilobyte delle dimensioni. Con l'aggiunta di altri componenti, l'allocazione di memoria iniziale viene condivisa dai componenti aggiunti, limitando l'aumento delle dimensioni dell'applicazione.

In questa sezione sono descritti alcuni vantaggi offerti dai componenti ActionScript 3.0.

ActionScript 3.0 offre un potente linguaggio di programmazione orientato agli oggetti che costituisce una fase importante nell'evoluzione delle funzionalità di Flash Player. Il linguaggio è concepito per la creazione di applicazioni Internet complesse con una base di codice riutilizzabile. ActionScript 3.0 è basato su ECMAScript, il linguaggio di scripting standard internazionale, ed è conforme con la specifica del linguaggio ECMAScript (ECMA-262) edizione 3. Per un'introduzione accurata a ActionScript 3.0, vedete la *Guida per gli sviluppatori di ActionScript 3.0*. Per informazioni di riferimento sul linguaggio, vedete la [Guida di riferimento di ActionScript 3.0](#).

I componenti dell'interfaccia utente basata su FLA offrono un facile accesso agli skin per consentire la personalizzazione durante la creazione. Questi componenti dispongono inoltre di stili, tra cui quelli degli skin, per personalizzare l'aspetto dei componenti stessi e caricare gli skin in fase di runtime. Per ulteriori informazioni, vedete ["Personalizzazione dei componenti UI"](#) a pagina 102 e la [Guida di riferimento di ActionScript 3.0](#).

Il nuovo componente FVLPlayback consente di aggiungere il componente FLVPlaybackCaptioning unitamente al supporto per la visualizzazione a schermo intero, l'anteprima dal vivo migliorata, gli skin per l'aggiunta di impostazioni di colore e trasparenza, nonché funzioni migliorate per lo scaricamento di file FLV e il controllo del layout.

La finestra di ispezione Proprietà e la finestra di ispezione dei componenti consentono di modificare i parametri dei componenti durante la creazione in Flash. Per ulteriori informazioni, vedete ["Operazioni con i file dei componenti"](#) a pagina 20 e ["Impostare parametri e proprietà"](#) a pagina 22.

La nuova finestra di dialogo relativa alla raccolta disponibile per i componenti ComboBox, List e TileList consente di compilare la relativa proprietà `dataProvider` tramite l'interfaccia utente. Per ulteriori informazioni, vedete ["Creare un componente DataProvider"](#) a pagina 31.

Il modello di eventi di ActionScript 3.0 consente all'applicazione di ascoltare gli eventi e di richiamare i gestori di eventi per rispondere. Per ulteriori informazioni, vedete ["Modello di gestione degli eventi ActionScript 3.0"](#) a pagina 10 e ["Gestione degli eventi"](#) a pagina 25.

Le classi di gestori offrono un modo rapido di gestire l'attivazione e gli stili in un'applicazione. Per ulteriori informazioni, vedete la [Guida di riferimento di ActionScript 3.0](#).

La classe base UIComponent fornisce i metodi, le proprietà e gli eventi principali per i componenti che estendono tale classe. Tutti i componenti dell'interfaccia utente di ActionScript 3.0 ereditano dalla classe UIComponent. Per ulteriori informazioni, vedete la classe UIComponent nella [Guida di riferimento di ActionScript 3.0](#).

L'uso di un file SWC nei componenti UI basati su FLA fornisce le definizioni sotto forma di risorse all'interno della linea temporale del componente per accelerare la compilazione.

Una gerarchia di classi facile da estendere mediante ActionScript 3.0 consente di creare spazi dei nomi univoci, nonché di importare agevolmente classi e sottoclassi in base alle esigenze al fine di estendere i componenti.

Per ulteriori informazioni, vedete la [Guida di riferimento di ActionScript 3.0](#).

Nota: Flash CS5 supporta sia i componenti basati su FLA che i componenti basati su SWC. Per ulteriori informazioni, vedete ["Architettura dei componenti"](#) a pagina 18.

Tipi di componente

I componenti Flash vengono installati quando si installa Flash CS5.

Tra i componenti ActionScript 3.0 sono inclusi i seguenti componenti dell'interfaccia utente (UI):

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

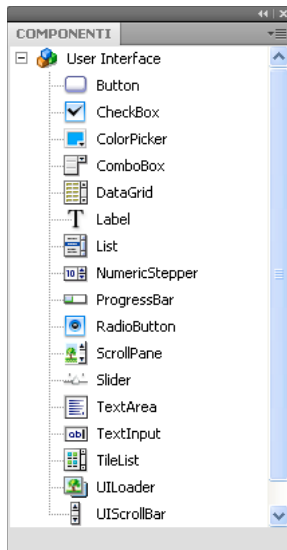
Oltre ai componenti dell'interfaccia utente, i componenti Flash ActionScript 3.0 comprendono i componenti e le classi di supporto seguenti:

- Il componente FLVPlayback (fl.video.FLVPlayback), basato su SWC.
Il componente FLVPlayback consente di includere con facilità un lettore video in un'applicazione Flash per la riproduzione di video in streaming progressivo via HTTP, da un servizio FVSS (* Flash® Video Streaming Service) o da Macromedia® Flash® Media Server di Adobe (FMS). Per ulteriori informazioni, vedete “[Uso del componente FLVPlayback](#)” a pagina 140.
- I componenti UI personalizzati FLVPlayback, basati su FLA e utilizzabili con le versioni ActionScript 2.0 e ActionScript 3.0 del componente FLVPlayback. Per ulteriori informazioni, vedete “[Uso del componente FLVPlayback](#)” a pagina 140.
- Il componente FLVPlaybackCaptioning, che fornisce funzionalità di didascalia per FLVPlayback. Vedete “[Uso del componente FLVPlaybackCaptioning](#)” a pagina 178.
Per un elenco completo dei componenti ActionScript 3.0 e delle loro classi di supporto, vedete la [Guida di riferimento di ActionScript 3.0](#).

Visualizzare componenti Flash

Per visualizzare i componenti Flash ActionScript 3.0 nel pannello Componenti, attenetevi alla procedura riportata di seguito.

- 1 Avviate Flash.
- 2 Create un nuovo file Flash (ActionScript 3.0) o aprite un documento Flash esistente nella cui finestra di dialogo Impostazioni pubblicazione sia specificato ActionScript 3.0.
- 3 Selezionate Finestra > Componenti per aprire il pannello Componenti, se non è già visualizzato.



Pannello Componenti con componenti dell'interfaccia utente

Potete anche scaricare componenti aggiuntivi da Adobe Exchange all'indirizzo http://www.adobe.com/go/flash_exchange_it. Per installare i componenti scaricati da Exchange, scaricate e installate Adobe® Extension Manager all'indirizzo http://www.adobe.com/go/exchange_it. Fate clic sul collegamento Adobe Exchange Home e cercate il collegamento Extension Manager.

Nel pannello Componenti di Flash potete visualizzare qualsiasi componente. Attenetevi alla procedura riportata di seguito per installare i componenti su un computer Windows® o Macintosh®.

Per installare i componenti su un computer Windows o Macintosh:

- 1 Uscite da Flash.
- 2 Inserite il file SWC o FLA che contiene il componente nella cartella seguente sul disco rigido:
 - In Windows:
C:\Programmi\Adobe\Adobe FlashCS5\lingua\Configuration\Components
 - Su Macintosh:
Macintosh HD:Applicazioni:Adobe Flash CS5:Configuration/Components
- 3 Avviate Flash.
- 4 Selezionate Finestra > Componenti per visualizzare il componente nel pannello Componenti, se non è già visualizzato.

Per ulteriori informazioni sui file dei componenti, vedete “Operazioni con i file dei componenti” a pagina 20.

Aggiungere ed eliminare componenti da un documento

Quando trascinate un componente basato su FLA dal pannello Componenti nello stage, Flash importa un clip filmato modificabile nella libreria. Quando trascinate un componente basato su SWC dal pannello Componenti nello stage, Flash importa un clip compilato nella libreria. Dopo che avete importato un componente nella libreria, potete trascinarne delle istanze nello stage sia dal pannello Libreria che dal pannello Componenti.

Aggiungere componenti in fase di creazione

Potete aggiungere un componente a un documento trascinandolo dal pannello Componenti. Le proprietà di ogni istanza di un componente possono essere impostate nella finestra di ispezione dei componenti o nella scheda Parametri della finestra di ispezione Proprietà.

- 1 Selezionate Finestra > Componenti.
- 2 Fate doppio clic sul componente nel pannello Componenti o trascinatelo nello stage.
- 3 Selezionate il componente nello stage.
- 4 Se la finestra di ispezione Proprietà non è visibile, selezionate Finestra > Proprietà > Proprietà.
- 5 Nella finestra di ispezione Proprietà, inserite un nome per l'istanza del componente.
- 6 Selezionate Finestra > Componenti, quindi la scheda Parametri per specificare i parametri per l'istanza.
Per ulteriori informazioni, vedete [“Impostare parametri e proprietà”](#) a pagina 22.
- 7 Ridimensionate il componente nel modo desiderato modificando i valori per larghezza (W:) e altezza (H:).
Per ulteriori informazioni sul ridimensionamento di specifici tipi di componente, vedete [“Personalizzazione dei componenti UI”](#) a pagina 102.
- 8 Selezionate Controllo > Prova filmato o premete Ctrl+Invio per compilare il documento e visualizzare i risultati delle impostazioni.

Inoltre, potete cambiare la formattazione del testo e il colore di un componente impostandone le proprietà dello stile oppure personalizzarne l'aspetto modificando gli skin del componente. Per ulteriori informazioni su questi argomenti, vedete [“Personalizzazione dei componenti UI”](#) a pagina 102.

Se trascinate un componente nello stage durante la fase di creazione, potete farvi riferimento utilizzando il relativo nome di istanza, ad esempio `myButton`.

Aggiungere componenti in fase di runtime mediante ActionScript

Per aggiungere un componente a un documento in fase di runtime con ActionScript, è necessario che il componente si trovi nella libreria dell'applicazione (Finestra > Libreria) quando compilate il file SWF. Per aggiungere un componente alla libreria, trascinatelo dal pannello Componenti al pannello Libreria. Per ulteriori informazioni sulla libreria, vedete [“La libreria”](#) a pagina 24.

Per rendere disponibile l'API di un componente in un'applicazione, dovete importare anche il relativo file di classe. I file di classe dei componenti sono installati in *pacchetti* contenenti una o più classi. Per importare la classe di un componente, usate l'istruzione `import` e specificate il nome del pacchetto e della classe. Per importare ad esempio la classe `Button`, usate l'istruzione `import` riportata di seguito:

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

Per sapere in quale pacchetto si trova un componente, vedete la [Guida di riferimento di ActionScript 3.0](#). Per ulteriori informazioni sulla posizione dei file di origine dei componenti, vedete [“Operazioni con i file dei componenti”](#) a pagina 20.

Per creare un'istanza di un componente, richiamate il metodo della funzione di costruzione ActionScript del componente. Ad esempio, con la seguente istruzione viene creata un'istanza `Button` chiamata `aButton`:

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

Il passaggio conclusivo consiste nel chiamare il metodo statico `addChild()` per aggiungere l'istanza del componente allo stage o al contenitore dell'applicazione. Ad esempio, con la seguente istruzione viene aggiunta l'istanza `aButton`:

```
addChild(aButton);
```

A questo punto, potete usare l'API del componente per specificare dinamicamente le dimensioni e la posizione del componente sullo stage, impostare l'ascolto degli eventi e definire le proprietà per modificarne il comportamento. Per ulteriori informazioni sull'API per un particolare componente, vedete la [Guida di riferimento di ActionScript 3.0](#).

Per ulteriori informazioni sul metodo `addChild()`, vedete “[Operazioni con l'elenco di visualizzazione](#)” a pagina 26.

Eliminare un componente

Per eliminare l'istanza di un componente dallo stage durante la creazione, selezionate semplicemente l'istanza e premete il tasto Canc. Viene rimossa l'istanza dallo stage, ma non il componente dall'applicazione.

Per eliminare un componente da un documento Flash dopo averlo posizionato nello stage o nella libreria, eliminate dalla libreria il componente e le risorse a esso associate. Non è sufficiente eliminare il componente dallo stage. Se il componente non viene rimosso dalla libreria, sarà incluso nell'applicazione al momento della compilazione.

- 1 Selezionate il simbolo relativo al componente nel pannello Libreria.
- 2 Fate clic sul pulsante Elimina nella parte inferiore del pannello Libreria oppure selezionate Elimina dal menu del pannello Libreria.

Ripetete questi passaggi per eliminare le eventuali risorse associate al componente.

Per informazioni sulla rimozione di un componente dal relativo contenitore mentre l'applicazione è in esecuzione, vedete “[Rimuovere un componente dall'elenco di visualizzazione](#)” a pagina 28.

Individuare la versione del componente

I componenti Flash ActionScript 3.0 sono dotati di una proprietà `version` che può essere visualizzata qualora fosse necessario fornirla al Supporto tecnico di Adobe o conoscere la versione del componente utilizzato.

Per visualizzare il numero di versione di un componente dell'interfaccia utente:

- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate il componente nello stage e assegnate un nome all'istanza. Ad esempio, trascinate un componente `ComboBox` nello stage e denominatelo `aCb`.
- 3 Premete il tasto **F9** o selezionate Finestra > Azioni per aprire il pannello Azioni.
- 4 Fate clic sul fotogramma 1 della linea temporale principale e aggiungete il codice seguente nel pannello Azioni:

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

Nel pannello Output dovrebbe essere visualizzato il numero di versione, simile a quello riportato nell'illustrazione seguente.

Per i componenti `FLVPlayback` e `FLVPlaybackCaptioning` è necessario usare il nome di classe anziché il nome di istanza, poiché il numero di versione è memorizzato in una costante di classe.

Per visualizzare il numero di versione dei componenti `FLVPlayback` e `FLVPlaybackCaptioning`:

- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate i componenti `FLVPlayback` e `FLVPlaybackCaptioning` nel pannello Libreria.
- 3 Premete il tasto **F9** o selezionate Finestra > Azioni per aprire il pannello Azioni.
- 4 Fate clic sul fotogramma 1 della linea temporale principale e aggiungete il codice seguente nel pannello Azioni.

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

I numeri di versione vengono visualizzati nel pannello Output.

Modello di gestione degli eventi ActionScript 3.0

In ActionScript 3.0 è stato introdotto un unico modello di gestione degli eventi che sostituisce i meccanismi di gestione degli eventi inclusi nelle versioni precedenti di ActionScript. Il nuovo modello di gestione si basa sulla specifica DOM (Document Object Model) Level 3.

Gli sviluppatori che hanno dimestichezza con l'impiego del metodo `addListener()` di ActionScript 2.0 possono trovare utili le informazioni seguenti, che sottolineano le differenze tra il modello listener di eventi di ActionScript 2.0 e il modello eventi di ActionScript 3.0. Nell'elenco riportato di seguito sono descritte alcune delle principali differenze tra i due modelli di evento:

- Per aggiungere listener di eventi in ActionScript 2.0 si usa `addListener()` in alcuni casi e `addEventListener()` in altri, mentre in ActionScript 3.0 si usa sempre `addEventListener()`.
- In ActionScript 2.0 non è presente un flusso di eventi, quindi il metodo `addListener()` può essere chiamato solo sull'oggetto che trasmette l'evento, mentre in ActionScript 3.0 il metodo `addEventListener()` può essere chiamato su qualsiasi oggetto appartenente al flusso di eventi.
- In ActionScript 2.0, i listener di eventi possono essere funzioni, metodi oppure oggetti mentre in ActionScript 3.0 possono essere listener di eventi solo funzioni o metodi.
- La sintassi `on(event)` non è più supportata in ActionScript 3.0, quindi non è possibile associare codice di eventi ActionScript a un clip filmato. Per aggiungere un listener di eventi, potete usare solo `addEventListener()`.

L'esempio seguente, impostato per l'intercettazione di un evento `MouseEvent.CLICK` su un componente `Button` denominato `aButton`, illustra il modello di base per la gestione degli eventi di 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);
}
```

Per ulteriori informazioni sulla gestione degli eventi in ActionScript 3.0, vedete la guida *Programmazione in ActionScript 3.0*. Per ulteriori informazioni sulla gestione degli eventi relativi ai componenti in ActionScript 3.0, vedete “[Gestione degli eventi](#)” a pagina 25.

Una semplice applicazione

In questa sezione sono illustrati i passaggi necessari per creare una semplice applicazione ActionScript 3.0 utilizzando i componenti Flash e lo strumento di creazione Flash. L'esempio viene fornito sia sotto forma di file FLA con il codice ActionScript incluso nella linea temporale, sia come file di classe ActionScript esterno con un file FLA contenente solo i componenti presenti nella libreria. In generale, è consigliabile sviluppare le applicazioni di grandi dimensioni usando file di classe esterni, per avere la possibilità di condividere il codice tra classi e applicazioni e facilitare la gestione delle applicazioni. Per ulteriori informazioni sulla programmazione in ActionScript 3.0, vedete la guida *Programmazione in ActionScript 3.0*.

Progettazione dell'applicazione

Il primo esempio di applicazione di componenti ActionScript è una variante dell'applicazione "Hello World" standard, il cui schema è quindi piuttosto semplice:

- L'applicazione si chiamerà Greetings.
- Il componente TextArea consente di visualizzare un saluto, che inizialmente è Hello World.
- Il componente ColorPicker consente di cambiare il colore del testo.
- I tre componenti RadioButton consentono di impostare la dimensione del testo scegliendo tra piccolo, grande e più grande.
- Il componente ComboBox consente di selezionare un saluto diverso da un elenco a discesa.
- Nell'applicazione vengono usati i componenti disponibili nel pannello Componenti e vengono creati gli elementi dell'applicazione mediante il codice ActionScript.

Con questa definizione del progetto, potete iniziare a realizzare l'applicazione.

Creare l'applicazione Greetings

La procedura seguente crea l'applicazione Greetings usando lo strumento di creazione Flash per ottenere un file FLA, posiziona i componenti sullo stage e aggiunge codice ActionScript alla linea temporale.

Creare l'applicazione Greetings in un file FLA:

- 1 Selezionate File > Nuovo.
- 2 Nella finestra di dialogo Nuovo documento, selezionate File Flash (ActionScript 3.0) e fate clic su OK.
Viene aperta una nuova finestra di Flash.
- 3 Selezionate File > Salva, assegnate al file Flash il nome **Greetings fla**, quindi fate clic sul pulsante Salva.
- 4 Nel pannello Componenti di Flash, selezionate un componente TextArea e trascinatelo nello stage.
- 5 Nella finestra Proprietà, con il componente TextArea selezionato nello stage, digitate **aTa** come nome di istanza e inserite le seguenti informazioni:
 - Inserite **230** per il valore W (larghezza).
 - Inserite **44** per il valore H (altezza).
 - Inserite **165** per il valore X (posizione orizzontale).
 - Inserite **57** per il valore Y (posizione verticale).
 - Digitate **Hello World!** per il parametro text nella scheda Parametri.
- 6 Trascinate un componente ColorPicker nello stage, posizionalo a sinistra di TextArea e assegnategli il nome di istanza **txtCp**. Immettete le seguenti informazioni nella finestra di ispezione Proprietà:
 - Inserite **96** per il valore X.
 - Inserite **72** per il valore Y.
- 7 Trascinate tre componenti RadioButton nello stage, uno alla volta, e assegnate loro i nomi di istanza **smallRb**, **largerRb** e **largestRb**. Immettete le seguenti informazioni per tali componenti nella finestra di ispezione Proprietà:
 - Inserite **100** per il valore W e **22** per il valore H per ognuno di essi.
 - Inserite **155** per il valore X.
 - Inserite **120** per il valore Y per smallRb, **148** per largerRb e **175** per largestRb.

- Inserite **fontRbGrp** per il parametro groupName per ognuno di essi.
 - Inserite le relative etichette nella scheda Parametri di **Small, Larger, Largest**.
- 8 Trascinate un componente ComboBox nello stage e assegnategli il nome di istanza **msgCb**. Immettete le seguenti informazioni per tale istanza nella finestra di ispezione Proprietà:
- Inserite **130** per il valore W.
 - Inserite **265** per il valore X.
 - Inserite **120** per il valore Y.
 - Nella scheda Parametri, immettete **Greetings** per il parametro prompt.
 - Fate doppio clic sul campo di testo del parametro dataProvider per aprire la finestra di dialogo Valori.
 - Fate clic sul segno più e sostituite il valore dell'etichetta con **Hello World!**
 - Ripetete il punto precedente per aggiungere i valori delle etichette **Have a nice day!** e **Top of the Morning!**
 - Fate clic su OK per chiudere la finestra di dialogo Valori.
- 9 Salvate il file.
- 10 Se non è già aperto, aprite il pannello Azioni premendo **F9** o scegliete Azioni dal menu Finestra. Fate clic sul fotogramma 1 della linea temporale principale e immettete il codice seguente nel pannello Azioni:

```
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);
```

Con le prime tre righe vengono importate le classi di eventi utilizzate dall'applicazione. Si verifica un evento quando un utente interagisce con uno dei componenti. Nelle cinque righe successive vengono registrati i gestori di eventi relativi agli eventi che l'applicazione dovrà ascoltare. Quando un utente fa clic su un componente RadioButton, si verifica un evento `click`. Quando un utente seleziona un colore diverso nel componente ColorPicker, si verifica un evento `change`. Quando un utente sceglie un saluto diverso dall'elenco a discesa, nel componente ComboBox si verifica un evento `change`.

Con la quarta riga viene importata la classe RadioButtonGroup per consentire all'applicazione di assegnare un listener di eventi al gruppo di componenti RadioButton, anziché assegnare il listener a ogni singolo pulsante.

- 11 Aggiungete la seguente riga di codice al pannello Azioni per creare l'oggetto TextFormat `tf`, che viene utilizzato dall'applicazione per modificare le proprietà di stile `size` e `color` del testo nel componente TextArea.

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

- 12 Aggiungete il codice seguente per creare la funzione di gestione degli eventi `rbHandler`. Questa funzione viene utilizzata per gestire un evento `click` quando un utente fa clic su uno dei componenti 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);
}
```

Questa funzione utilizza un'istruzione `switch` per esaminare la proprietà `target` dell'oggetto `event` e determinare quale componente `RadioButton` ha attivato l'evento. Il nome dell'oggetto che ha attivato l'evento è contenuto nella proprietà `currentTarget`. A seconda del componente `RadioButton` su cui l'utente ha fatto clic, le dimensioni del testo nel componente `TextArea` vengono modificate dall'applicazione in 14, 18 o 24 punti.

- 13** Aggiungete il codice seguente per implementare la funzione `cpHandler()` che gestisce la modifica del valore nel componente `ColorPicker`:

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

Questa funzione imposta la proprietà `color` dell'oggetto `TextFormat tf` sul colore selezionato in `ColorPicker`, quindi chiama `setStyle()` per applicare il colore al testo nell'istanza `TextArea aTa`.

- 14** Aggiungete il codice seguente per implementare la funzione `cbHandler()`, che gestisce la modifica della selezione nel componente `ComboBox`:

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

Questa funzione sostituisce semplicemente il testo in `TextArea` con il testo selezionato in `ComboBox`, `event.target.selectedItem.label`.

- 15** Selezionate **Controllo > Prova filmato** o premete **Ctrl+Invio** per compilare il codice e provare l'applicazione `Greetings`.

Nella seguente sezione viene illustrato come creare la stessa applicazione con una classe `ActionScript` esterna e un file `FLA` che include solo i componenti necessari nella libreria.

Creare l'applicazione `Greetings2` con un file di classe esterno:

- 1 Selezionate **File > Nuovo**.
- 2 Nella finestra di dialogo **Nuovo documento**, selezionate **File Flash (ActionScript 3.0)** e fate clic su **OK**.
Viene aperta una nuova finestra di `Flash`.
- 3 Selezionate **File > Salva**, assegnate al file `Flash` il nome **Greetings2.fla**, quindi fate clic sul pulsante **Salva**.
- 4 Trascinate ognuno dei seguenti componenti dal pannello **Componenti** alla libreria:
 - `ColorPicker`
 - `ComboBox`

- RadioButton
- TextArea

Poiché nel file SWF compilato verrà utilizzata ognuna di queste risorse, dovete aggiungerle alla libreria. Trascinate i componenti nella parte inferiore del pannello Libreria. Mentre aggiungete questi componenti alla libreria, vengono aggiunti automaticamente altri componenti (ad esempio List, TextInput e UI ScrollBox).

5 Nella finestra Proprietà, digitate **Greetings2** per la classe del documento.

Se viene visualizzato un avviso in cui è specificato che è impossibile trovare una definizione per la classe del documento, ignoratelo. La classe Greetings2, che sarà definita nella procedura seguente, specifica la funzionalità principale dell'applicazione.

6 Salvate il file Greetings2 fla.

7 Selezionate File > Nuovo.

8 Nella finestra di dialogo Nuovo documento, selezionate File ActionScript e fate clic su OK.

Viene aperta una nuova finestra Script.

9 Aggiungete il codice seguente nella finestra 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() {
```

Lo script definisce una classe ActionScript 3.0 di nome Greetings2 ed esegue le seguenti operazioni:

- Importa le classi che saranno utilizzate nel file. Generalmente queste istruzioni di importazione vengono aggiunte mentre si fa riferimento a diverse classi nel codice; tuttavia, per brevità, in questo esempio vengono importate tutte in un solo passaggio.
- Dichiarare le variabili che rappresentano i diversi tipi di oggetti componente che saranno aggiunti al codice. Un'altra variabile crea l'oggetto TextFormat tf.
- Definisce una funzione di costruzione, Greetings2(), per la classe. Nella procedura seguente verranno aggiunte delle righe a questa funzione e altri metodi alla classe.

10 Selezionate File > Salva, assegnate al file il nome **Greetings2.as**, quindi fate clic sul pulsante Salva.

11 Aggiungete le seguenti righe di codice alla funzione Greeting2():

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

La funzione dovrebbe avere un aspetto analogo al seguente:

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

12 Aggiungete le seguenti righe di codice dopo la parentesi graffa di chiusura del metodo `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);
}
```

Queste righe di codice eseguono le seguenti operazioni:

- Creano istanze dei componenti utilizzati nell'applicazione.
- Impostano la dimensione, la posizione e le proprietà di ciascun componente.
- Aggiungono ogni componente sullo stage utilizzando il metodo `addChild()`.

13 Dopo la parentesi graffa di chiusura del metodo `bldRadioButtons()`, aggiungete il codice seguente per il metodo `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;
}
}
}
```

Queste funzioni definiscono i listener di eventi per i componenti.

14 Selezionate File > Salva per salvare il file.

15 Selezionate Controllo > Prova filmato o premete Ctrl+Invio per compilare il codice e provare l'applicazione Greetings2.

Sviluppare ed eseguire ulteriori esempi

Dopo aver sviluppato ed eseguito l'applicazione Greetings, dovrete possedere le conoscenze di base necessarie per eseguire gli altri esempi di codice presentati nel manuale. Per ciascun esempio viene evidenziato e discusso il codice ActionScript 3.0 da utilizzare, quindi dovrebbe essere semplice copiare e incollare ognuno di questi esempi in un file FLA, compilarlo e quindi eseguirlo.

Capitolo 3: Operazioni con i componenti

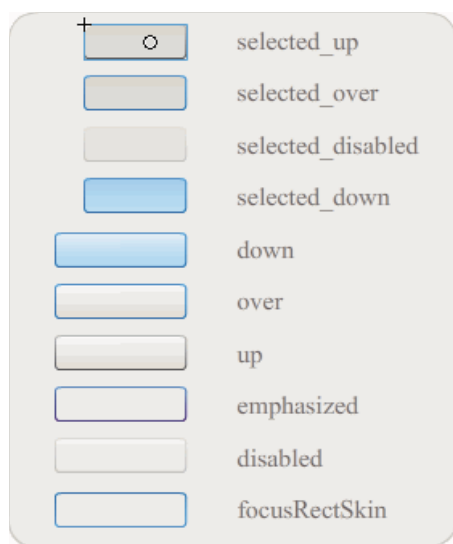
Architettura dei componenti

I componenti Adobe® ActionScript 3.0® sono supportati da Adobe® Flash Player versione 9.0.28.0 e successive. Questi componenti non sono compatibili con quelli creati prima di Flash CS4. Per informazioni sull'uso dei componenti Adobe® ActionScript® 2.0, vedete la guida *Usa dei componenti Adobe® ActionScript® 2.0* e la *Guida di riferimento dei componenti Adobe® ActionScript® 2.0*.

I componenti dell'interfaccia utente (UI) Adobe ActionScript 3.0 sono implementati come componenti basati su FLA, tuttavia Flash CS5 supporta sia i componenti basati su FLA che quelli basati su SWC. FLVPlayback e FLVPlaybackCaptioning, ad esempio, sono componenti basati su SWC. Potete inserire entrambi i tipi di componenti nella cartella Components, in modo che siano visualizzati nel pannello Components. Questi due tipi di componenti sono creati in modo diverso, quindi vengono descritti separatamente.

Componenti ActionScript 3.0 basati su FLA

I componenti di interfaccia utente ActionScript 3.0 sono basati su file FLA (.fla) con skin incorporati a cui potete accedere per la modifica facendo doppio clic sul componente nello stage. Gli skin e altre risorse del componente sono posizionati nel fotogramma 2 della linea temporale. Quando fate doppio clic sul componente, lo stato attivo passa automaticamente al fotogramma 2 e viene aperta una tavolozza degli skin del componente. L'illustrazione seguente mostra la tavolozza con gli skin che vengono visualizzati per il componente Button.



Skin per il componente Button

Per ulteriori informazioni sugli skin e sulla personalizzazione dei componenti, vedete [“Personalizzazione dei componenti UI”](#) a pagina 102 e [“Personalizzare il componente FLVPlayback”](#) a pagina 159.

Per velocizzare la compilazione delle applicazioni ed evitare conflitti con le impostazioni di ActionScript 3.0, i componenti UI basati su FLA di Flash CS5 includono inoltre un file SWC contenente il codice ActionScript già compilato del componente. Il file SWC `ComponentShim` viene posizionato sullo stage nel fotogramma 2 di ogni componente dell'interfaccia utente per rendere disponibili le definizioni precompilate. Affinché sia disponibile per ActionScript, è necessario che un componente sia posizionato nello stage o nella libreria e che nelle relative proprietà di concatenamento sia selezionata l'opzione `Esporta` nel primo fotogramma. Per creare un componente con ActionScript, dovete inoltre importare la classe con un'istruzione `import` che ne consenta l'accesso. Per informazioni sull'istruzione `import`, vedete la [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Componenti basati su SWC

Anche i componenti basati su SWC sono dotati di un file FLA e di un file di classe ActionScript, ma sono stati compilati ed esportati come file SWC. Un file SWC è un pacchetto di simboli Flash e codice ActionScript precompilati che consentono di evitare la ricompilazione dei simboli e del codice che rimangono invariati.

`FLVPlayback` e `FLVPlaybackCaptioning` sono componenti basati su SWC e sono dotati di skin esterni anziché incorporati. Al componente `FLVPlayback` è associato uno skin predefinito che potete cambiare selezionandone uno da una raccolta di skin predefiniti, personalizzando i controlli dai controlli UI nel pannello Componenti (`BackButton`, `BufferingBar` e così via) o creando uno skin personalizzato. Per ulteriori informazioni, vedete “[Personalizzare il componente FLVPlayback](#)” a pagina 159.

In Flash potete convertire un clip filmato in un clip compilato come illustrato di seguito:

Compilare un clip filmato

- Fate clic con il pulsante destro del mouse (Windows) o premete `Ctrl` e fate clic (Macintosh) sul nome di un clip filmato nel pannello Libreria, quindi selezionate `Converti in clip compilato`.

Il clip compilato si comporta esattamente come il clip filmato da cui è stato compilato, ma viene visualizzato e pubblicato molto più rapidamente rispetto al clip filmato normale. Il clip compilato non può essere modificato, ma le relative proprietà vengono visualizzate nella finestra di ispezione `Proprietà` e nella finestra di ispezione dei componenti.

I componenti SWC contengono un clip compilato, le definizioni ActionScript precompilate del componente e altri file che descrivono il componente. Se create un componente personalizzato, potete esportarlo come file SWC per la distribuzione.

Esportare un file SWC

- Selezionate il clip filmato nel pannello Libreria e fate clic con il pulsante destro del mouse (Windows) oppure fate clic tenendo premuto il tasto `Ctrl` (Macintosh), quindi selezionate `Esporta file SWC`.

***Nota:** il formato di un file SWC Flash CS4 o successivo è compatibile con il formato SWC Flex, quindi i file SWC possono essere scambiati tra i due prodotti, ma non necessariamente senza modifiche.*

Per informazioni sulla creazione di componenti basati su SWC, visitate www.adobe.com/go/learn_fl_creating_components_it.

API dei componenti ActionScript 3.0

Ogni componente ActionScript 3.0 è basato su una classe ActionScript 3.0 presente in una cartella del pacchetto e con un nome nel formato `fl.nome_pacchetto.nome_classe`. Ad esempio, il componente `Button` è un'istanza della classe `Button` il cui nome di pacchetto è `fl.controls.Button`. Quando importate la classe di un componente nell'applicazione, dovete fare riferimento al nome del pacchetto. Per importare la classe `Button`, usate l'istruzione riportata di seguito:

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

Per ulteriori informazioni sulla posizione dei file di classe dei componenti, vedete “[Operazioni con i file dei componenti](#)” a pagina 20.

La classe di un componente definisce i metodi, le proprietà, gli eventi e gli stili che consentono di interagire con il componente all'interno dell'applicazione. I componenti UI ActionScript 3.0 sono sottoclassi delle classi Sprite e UIComponent e ne ereditano le proprietà, i metodi e gli eventi. La classe Sprite è il blocco costitutivo di base dell'elenco di visualizzazione ed è simile a MovieClip ma è priva di una linea temporale. La classe UIComponent è la classe di base per tutti i componenti visivi, sia interattivi che non interattivi. Il percorso di ereditarietà di ciascun componente, nonché le sue proprietà, metodi, eventi e stili sono descritti nella *Guida di riferimento di ActionScript 3.0 per Flash Professional di Adobe*.

Tutti i componenti ActionScript 3.0 utilizzano il modello di gestione degli eventi di ActionScript 3.0. Per ulteriori informazioni sulla gestione degli eventi, vedete “[Gestione degli eventi](#)” a pagina 25 e *Programmazione in ActionScript 3.0*.

Operazioni con i file dei componenti

In questa sezione viene illustrato dove sono memorizzati i file dei componenti, dove trovare i file di origine ActionScript e come aggiungere e rimuovere i componenti dal pannello Componenti.

Percorso di memorizzazione dei file dei componenti

I componenti Flash vengono memorizzati nella cartella Configuration dell'applicazione.

Nota: per informazioni su questo tipo di cartella, vedete “*Cartelle di configurazione installate con Flash*” in *Uso di Flash*.

I componenti vengono installati nei seguenti percorsi:

- Windows 2000 o Windows XP: C:\Programmi\Adobe\Adobe Flash CS5\lingua\Configuration\Components
- Mac OS X: Macintosh HD:Applicazioni:Adobe Flash CS5:Configuration:Components

Nella cartella Components i componenti dell'interfaccia utente (UI) si trovano nel file User Interface.fla, mentre i componenti FLVPlayback (FLVPlaybackAS3.swc) e FLVPlaybackCaptioning si trovano nella cartella Video.

Inoltre, potete memorizzare i componenti nei seguenti percorsi basati sull'utente:

- Windows 2000 or Windows XP: C:\Documents and Settings*nome utente*\Impostazioni locali\Application Data\Adobe\Adobe Flash CS5\it\Configuration\Components
- Windows Vista: C:\Utenti*nome utente*\Impostazioni locali\Application Data\Adobe\Adobe Flash CS5\it\Configuration\Components

Nota: in Windows, la cartella Dati applicazioni è nascosta per impostazione predefinita. Per visualizzare le cartelle e i file nascosti, selezionate Risorse del computer per aprire Esplora risorse e scegliete Strumenti>Opzioni cartella, quindi selezionate la scheda Visualizzazione. Nella scheda Visualizzazione, selezionate il pulsante di opzione Visualizza cartelle e file nascosti.

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

Percorso di memorizzazione dei file di origine dei componenti

I file di classe ActionScript (.as) (o *file di origine*) per i componenti sono installati nelle seguenti cartelle dell'applicazione per Windows 2000 o Windows XP:

Componenti dell'interfaccia utente C:\Programmi\Adobe\Adobe Flash CS5\it\Configuration\Component Source\ActionScript 3.0\User Interface\fl

FLVPlayback C:\Programmi\Adobe\Adobe Flash CS5\it\Configuration\Component Source\ActionScript 3.0\FLVPlayback\fl\video

FLVPlaybackCaptioning C:\Programmi\Adobe\Adobe Flash CS5\it\Configuration\Component Source\ActionScript 3.0\FLVPlaybackCaptioning\fl\video

Per Mac OS X i file di origine dei componenti sono memorizzati nel seguente percorso:

Componenti dell'interfaccia utente Macintosh HD:Applicazioni:Adobe Flash CS5:Configuration:Component Source>ActionScript 3.0>User Interface:fl

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

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

File di origine dei componenti e variabile Classpath

Poiché i componenti ActionScript 3.0 contengono codice compilato, evitate di specificare il percorso dei file di classe ActionScript nella variabile Classpath. In caso contrario, la compilazione dell'applicazione richiederà più tempo. Se, tuttavia, nell'impostazione di Classpath sono presenti file di classe dei componenti, questi ultimi avranno sempre la precedenza rispetto al codice compilato contenuto nei componenti.

Uno dei casi in cui è consigliabile aggiungere il percorso dei file di origine dei componenti all'impostazione di Classpath è l'esecuzione del debug di un'applicazione con componenti. Per ulteriori informazioni, vedete “[Effettuare il debug delle applicazioni di componenti](#)” a pagina 22.

Modificare i file dei componenti

Se in Flash aggiornate, aggiungete o rimuovete componenti basati su SWC o aggiungete nuovi componenti basati su FLA, dovete ricaricarli nel pannello Componenti per renderli disponibili. Per ricaricare i componenti potete riavviare Flash o scegliere Ricarica dal menu del pannello Componenti. In questo modo, tutti i componenti aggiunti alla cartella Components vengono rilevati automaticamente da Flash.


Ricaricare i componenti nel pannello Componenti mentre Flash è in esecuzione:

- Scegliete Ricarica dal menu del pannello Componenti.

Rimuovere un componente dal pannello Componenti:

- Rimuovete il file FLA, SWC o MXP dalla cartella Components e riavviate Flash o scegliete Ricarica dal menu del pannello Componenti. Un file MXP è un file di componente scaricato da Adobe Exchange.

Potete rimuovere e sostituire i componenti basati su SWC mentre Flash è in esecuzione e, in caso scegliate di ricaricarli, le modifiche vengono applicate immediatamente. Se, tuttavia, modificate o eliminate i componenti basati su FLA, le modifiche non vengono applicate finché non terminate e riavviate Flash. Tuttavia potete aggiungere i componenti basati su FLA e caricarli con il comando Ricarica.

 Adobe consiglia di effettuare una copia dei file dei componenti Flash (.fla o .as) prima di modificarli, in modo da poterli ripristinare se necessario.

Effettuare il debug delle applicazioni di componenti

Nei componenti ActionScript 3.0 è contenuto tutto il relativo codice di origine che consente di ridurre il tempo di compilazione dell'applicazione. Il debugger di Flash non è tuttavia in grado di esaminare il codice all'interno dei clip compilati, quindi se desiderate eseguire il debug dell'applicazione fino al codice di origine, dovete aggiungere i file di origine dei componenti all'impostazione di Classpath.

Il percorso delle cartelle dei pacchetti di componenti è relativo al percorso dei file di origine del tipo di componente. Per fare riferimento a tutti i file di origine ActionScript 3.0 per tutti i componenti UI, aggiungete il seguente percorso a Classpath per i pacchetti dell'interfaccia utente:

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

Nota: in questo modo viene ignorato il codice compilato contenuto in tutti i componenti UI e aumenta il tempo di compilazione dell'applicazione. Se per qualsiasi motivo il file di origine di un componente è stato modificato, è possibile che il risultato prodotto da tale componente presenti un comportamento diverso.

Per impostare il percorso di classe, scegliete Preferenze dal menu Modifica e selezionate ActionScript dall'elenco Categoria, quindi fate clic sul pulsante Impostazioni di ActionScript 3.0. Per aggiungere una nuova voce, fate clic sul segno più sopra la finestra in cui sono visualizzate le impostazioni correnti.

La variabile \$(AppConfig) fa riferimento alla cartella Configuration di Flash CS5 nel percorso di installazione di Flash CS5. In genere il percorso è simile al seguente:

- Windows 2000 o Windows XP: C:\Programmi\Adobe\Adobe Flash CS5\lingua\Configuration\
- Mac OS X: Macintosh HD:Applicazioni:Adobe Flash CS5:Configuration

Nota: se dovete modificare il file di origine di un componente, si consiglia di copiare il file di origine originale in un percorso diverso e aggiungere tale percorso a Classpath.

Per ulteriori informazioni sulla posizione dei file di origine dei componenti, vedete la sezione "[Percorso di memorizzazione dei file di origine dei componenti](#)" a pagina 21.

Impostare parametri e proprietà

L'aspetto e il comportamento dei componenti possono essere modificati tramite i relativi parametri. Un parametro è una proprietà della classe del componente presente nella finestra di ispezione Proprietà e nella finestra di ispezione dei componenti. Le proprietà utilizzate più comunemente sono visualizzate come parametri di creazione, mentre le altre devono essere impostate con ActionScript. Tutti i parametri impostabili durante la creazione possono essere impostati anche con ActionScript. L'impostazione di un parametro con ActionScript sovrascrive qualsiasi valore impostato durante la creazione.

La maggior parte dei componenti dell'interfaccia utente ActionScript 3.0 eredita le proprietà e i metodi dalla classe `UIComponent` e da una classe base. Le classi `Button` e `CheckBox`, ad esempio, ereditano le proprietà dalla classe `UIComponent` e dalla classe `BaseButton`. Potete accedere sia alle proprietà ereditate di un componente, sia a quelle della relativa classe. Ad esempio, il componente `ProgressBar` eredita la proprietà `ProgressBar.enabled` da `UIComponent`, ma dispone anche di una proprietà specifica: `ProgressBar.percentComplete`. Potete accedere a entrambe queste proprietà per interagire con un'istanza del componente `ProgressBar`. Per ulteriori informazioni sulle proprietà di un componente, vedete la voce relativa alla classe nella [Guida di riferimento di ActionScript 3.0](#).

Potete impostare i parametri dell'istanza di un componente tramite la finestra di ispezione Proprietà e la finestra di ispezione dei componenti.

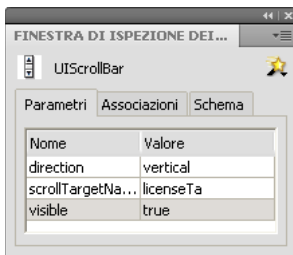
Inserire un nome di istanza di un componente nella finestra di ispezione Proprietà:

- 1 Selezionate Finestra > Proprietà > Proprietà.
- 2 Selezionate un'istanza di un componente sullo stage.
- 3 Immettete il nome per l'istanza del componente nella casella <Nome istanza>, situata sotto l'elenco a discesa che visualizza la voce Clip filmato. In alternativa, fate clic sulla scheda Parametri e immettete un nome di istanza nella casella sotto la parola *Componente*. Inserite i valori dei parametri che desiderate impostare.

Per semplificare la lettura del codice ActionScript, è buona norma aggiungere al nome dell'istanza un suffisso per indicare il tipo di componente. Ad esempio il nome dell'istanza `licenseSb` indica che il componente è una barra di scorrimento che consente di scorrere un accordo di licenza nell'area di testo `licenseTa`.

Inserire i parametri di un'istanza di componente nella finestra di ispezione dei componenti:

- 1 Selezionate Finestra > Finestra di ispezione dei componenti.
- 2 Selezionate un'istanza di un componente sullo stage.
- 3 Fate clic sulla scheda Parametri e inserite i valori per tutti i parametri elencati.



Parametri dei componenti nella finestra di ispezione dei componenti

Impostare le proprietà dei componenti in ActionScript

In ActionScript si utilizza l'operatore punto (`.`) (sintassi del punto) per accedere alle proprietà o ai metodi che appartengono a un oggetto o a un'istanza nello stage. Un'espressione in cui è utilizzato questo tipo di sintassi inizia con il nome dell'istanza seguito da un punto e termina con l'elemento che si desidera specificare. Ad esempio, il codice ActionScript seguente imposta la proprietà `width` dell'istanza `CheckBox aCh` in modo che corrisponda a 50 pixel di larghezza:

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

L'istruzione `if` seguente consente di verificare se l'utente ha selezionato la casella di controllo:

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

La libreria

Quando aggiungete un componente a un documento per la prima volta, Flash lo importa come clip filmato nel pannello Libreria. Potete inoltre trascinare un componente dal pannello Componenti direttamente nel pannello Libreria, quindi aggiungerne un'istanza nello stage. In ogni caso, dovete aggiungere un componente alla libreria per poter accedere agli elementi della relativa classe.

Se aggiungete un componente alla libreria e create un'istanza di tale componente utilizzando ActionScript, dovete importare innanzitutto la relativa classe con l'istruzione `import`. Nell'istruzione `import` dovete specificare sia il nome del pacchetto del componente sia il nome della relativa classe. Ad esempio, l'istruzione seguente consente di importare la classe `Button`:

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

Quando inserite un componente nella libreria, Flash importa automaticamente anche una cartella della relative risorse, che contiene gli *skin* per i diversi stati. Gli *skin* di un componente includono l'insieme dei simboli che compongono la visualizzazione grafica del componente nell'applicazione. Un singolo *skin* costituisce la rappresentazione grafica, o clip filmato, che indica un particolare stato del componente.

Il contenuto della cartella `Assets` consente di cambiare gli *skin* del componente, se lo desiderate. Per ulteriori informazioni, vedete “[Personalizzazione dei componenti UI](#)” a pagina 102.

Dopo avere inserito un componente nella libreria, potete aggiungerne delle istanze al documento trascinando la relativa icona nello stage dal pannello Libreria o dal pannello Componenti.

Ridimensionamento dei componenti

Per ridimensionare le istanze dei componenti, usate lo strumento Trasformazione libera oppure il metodo `setSize()`. Il metodo `setSize()` può essere chiamato da qualsiasi istanza del componente che deve essere ridimensionato (vedete `UIComponent.setSize()`). Il codice seguente ridimensiona un'istanza del componente `List` a 200 pixel di larghezza e 300 pixel di altezza:

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

I componenti non vengono ridimensionati automaticamente in base all'etichetta. Se l'istanza di un componente aggiunta a un documento non è sufficientemente grande da visualizzare la relativa etichetta, il testo dell'etichetta viene troncato. Dovete ridimensionare il componente in modo da rendere visibile l'intera etichetta.

Per ulteriori informazioni sul ridimensionamento dei componenti, vedete le singole voci nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

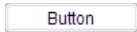
Anteprima dal vivo

La funzione Anteprima dal vivo, attivata per impostazione predefinita, consente di esaminare sullo stage, nelle loro dimensioni approssimative, l'aspetto che i componenti assumeranno nel contenuto Flash pubblicato.

Per attivare o disattivare la funzione Anteprima dal vivo:

- Selezionate **Controllo > Attiva anteprima dal vivo**. Un segno di spunta indica che l'opzione è attivata.

L'anteprima dal vivo presenta parametri diversi a seconda dei componenti. Per informazioni su quali parametri dei componenti vengono riprodotti nell'anteprima dal vivo, vedete ciascuna voce componente nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).



Un componente Button con la funzione Anteprima dal vivo attivata



Un componente Button con la funzione Anteprima dal vivo disattivata

I componenti visualizzati mediante la funzione Anteprima dal vivo non sono funzionali. Per verificare la funzionalità dovete usare il comando Controllo > Prova filmato.

Gestione degli eventi

Gli eventi di ogni componente vengono trasmessi a seguito dell'interazione dell'utente. Ad esempio, quando un utente fa clic su un componente Button, viene inviato un evento `MouseEvent.CLICK`; mentre quando un utente seleziona una voce dal componente List, viene inviato un evento `Event.CHANGE`. Un evento può inoltre essere generato quando si verifica un evento significativo per il componente. Ad esempio, una volta terminato il caricamento del contenuto per un'istanza `UILoader`, viene generato un evento `Event.COMPLETE`. Per gestire un evento, potete scrivere del codice ActionScript che viene eseguito quando l'evento si verifica.

Gli eventi di un componente includono gli eventi di tutte le classi dalle quali il componente eredita. Questo significa che tutti i componenti di interfaccia utente ActionScript 3.0 ereditano gli eventi dalla classe `UIComponent`, che costituisce la classe base di tali componenti. Per vedere l'elenco degli eventi trasmessi da un componente, vedete la sezione Eventi della voce classe del componente nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Per una descrizione completa della gestione degli eventi in ActionScript 3.0, vedete la guida *Programmazione in ActionScript 3.0*.

Informazioni sui listener di eventi

I punti chiave riportati di seguito riguardano la gestione degli eventi relativi ai componenti ActionScript 3.0:

- Tutti gli eventi vengono trasmessi da un'istanza della classe di un componente. L'istanza del componente è il *broadcaster*.
- Per registrare un *listener* di eventi, chiamate il metodo `addEventListener()` per l'istanza del componente. Ad esempio, la seguente riga di codice consente di aggiungere un listener per l'evento `MouseEvent.CLICK` all'istanza `aButton` del componente Button:

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

Il secondo parametro del metodo `addEventListener()` registra il nome della funzione, `clickHandler`, che dovrà essere chiamata quando si verifica l'evento. Questa funzione è inoltre nota come *funzione di callback*.

- Potete registrare più listener in un'istanza del componente.

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

- Potete registrare un listener in più istanze del componente.

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

- La funzione del gestore di eventi viene passata a un oggetto evento contenente le informazioni relative al tipo di evento e all'istanza da cui è stato trasmesso. Per ulteriori informazioni, vedete “[Informazioni sull'oggetto evento](#)” a pagina 26.
- Il listener rimane attivo finché l'applicazione non viene terminata o finché non viene rimosso in modo esplicito usando il metodo `removeEventListener()`. Ad esempio, la seguente riga di codice consente di rimuovere il listener per l'evento `MouseEvent.CLICK` dall'istanza `aButton`:

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

Informazioni sull'oggetto evento

L'oggetto evento eredita dalla classe di oggetti `Event` e ha una serie di proprietà che contengono informazioni sull'evento che si è verificato, tra cui le proprietà `target` e `type`, che forniscono informazioni relative all'evento:

Proprietà	Descrizione
<code>type</code>	Una stringa che indica il tipo dell'evento.
<code>target</code>	Un riferimento all'istanza di componente che trasmette l'evento.

Quando un evento dispone di proprietà aggiuntive, queste vengono elencate nella descrizione della classe dell'evento nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

L'oggetto evento viene generato e passato automaticamente alla funzione gestore di eventi quando si verifica un evento.

Potete usare l'oggetto evento all'interno della funzione per accedere al nome dell'evento trasmesso o al nome di istanza del componente che ha trasmesso l'evento. Dal nome di istanza potete accedere alle altre proprietà del componente. Ad esempio, il codice seguente usa la proprietà `target` dell'oggetto evento `evtObj` per accedere alla proprietà `label` dell'istanza `aButton` e visualizzarla nel pannello Output:

```
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");  
}
```

Operazioni con l'elenco di visualizzazione

Tutti i componenti ActionScript 3.0 ereditano dalla classe `DisplayObject`, quindi hanno accesso ai relativi metodi e proprietà per interagire con l'elenco di visualizzazione. L'*elenco di visualizzazione* è la rappresentazione gerarchica degli oggetti visualizzati e degli elementi visivi in un'applicazione e include i seguenti elementi:

- Lo stage, che costituisce il contenitore di livello superiore.
- Gli oggetti di visualizzazione, che includono forme, clip filmato, campi di testo e altri elementi.

- I contenitori di oggetti di visualizzazione, ovvero tipi speciali di oggetti di visualizzazione che possono contenere oggetti di visualizzazione secondari.

L'ordine degli oggetti nell'elenco di visualizzazione ne determina la profondità nel contenitore principale. La profondità di un oggetto ne indica la posizione dall'alto verso il basso o dal primo piano allo sfondo nello stage o nel relativo contenitore di visualizzazione. L'ordine di profondità è evidente quando gli oggetti sono sovrapposti, ma esiste anche quando non lo sono. A ogni oggetto nell'elenco di visualizzazione corrisponde una profondità nello stage. Per modificare la profondità di un oggetto, posizionandolo in primo piano o sullo sfondo rispetto ad altri oggetti, dovete cambiarne la posizione nell'elenco di visualizzazione. L'ordine predefinito degli oggetti nell'elenco di visualizzazione corrisponde all'ordine in cui sono inseriti nello stage. La posizione 0 nell'elenco di visualizzazione corrisponde all'estremità inferiore dell'ordine di profondità.

Aggiungere un componente all'elenco di visualizzazione

Potete aggiungere un oggetto a un oggetto `DisplayObjectContainer` chiamando il metodo `addChild()` o `addChildAt()` del contenitore. Nel caso dello stage, potete inoltre aggiungere un oggetto al relativo elenco di visualizzazione creando l'oggetto durante la fase di creazione oppure, nel caso dei componenti, trascinando l'oggetto dal pannello Componenti nello stage. Per aggiungere un oggetto a un contenitore con `ActionScript`, create innanzitutto un'istanza dell'oggetto richiamando la relativa funzione di costruzione con l'operatore `new`, quindi chiamando il metodo `addChild()` o `addChildAt()` per inserire l'oggetto nello stage e nell'elenco di visualizzazione. Il metodo `addChild()` consente di inserire l'oggetto nella successiva posizione dell'elenco di visualizzazione, mentre l'oggetto `addChildAt()` specifica la posizione in cui aggiungere l'oggetto. Se specificate una posizione già occupata, l'oggetto presente in quella posizione e quelli nelle posizioni superiori vengono spostati verso l'alto di una posizione. La proprietà `numChildren` dell'oggetto `DisplayObjectContainer` specifica il numero di oggetti di visualizzazione contenuti. Potete recuperare un oggetto dall'elenco di visualizzazione chiamando il metodo `getChildAt()` e specificando la posizione oppure, se conoscete il nome dell'oggetto, chiamando il metodo `getChildByName()`.

***Nota:** quando aggiungete un componente con `ActionScript`, dovete assegnare un nome alla relativa proprietà `name` se desiderate accedervi per nome nell'elenco di visualizzazione.*

Nell'esempio seguente vengono visualizzati i nomi e le posizioni di tre componenti nell'elenco di visualizzazione. Trascinate innanzitutto un'istanza degli oggetti `NumericStepper`, `Button` e `ComboBox` nello stage in modo che si sovrappongano e assegnatevi i nomi di istanza `aNs`, `aButton` e `aCb`. Aggiungete il codice seguente nel pannello Azioni al fotogramma 1 nella linea temporale:

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

Nel pannello Output dovrebbero essere visualizzate le seguenti righe:

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

Spostare un componente nell'elenco di visualizzazione

Potete cambiare la posizione di un oggetto nell'elenco di visualizzazione e la relativa profondità chiamando il metodo `addChildAt()` e specificando il nome dell'oggetto e la posizione in cui desiderate inserirlo come parametri del metodo. Ad esempio, aggiungete il seguente codice all'esempio precedente per posizionare l'oggetto `NumericStepper` nella posizione superiore e ripetete il ciclo per visualizzare le nuove posizioni del componente nell'elenco di visualizzazione:

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

I dati seguenti devono essere visualizzati nel pannello Output:

```
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
```

Sullo schermo, l'oggetto NumericStepper dovrebbe inoltre essere visualizzato in primo piano rispetto agli altri componenti.

Notate che `numChildren` è il numero di oggetti (da 1 a n) nell'elenco di visualizzazione, mentre la prima posizione nell'elenco corrisponde a 0. Quindi, se nell'elenco sono presenti tre oggetti, la posizione di indice del terzo oggetto è 2. Ciò significa che potete fare riferimento all'ultima posizione nell'elenco di visualizzazione, ovvero l'oggetto di livello superiore in termini di profondità di visualizzazione, come `numChildren - 1`.

Rimuovere un componente dall'elenco di visualizzazione

Potete rimuovere un componente da un contenitore di oggetti di visualizzazione e dal relativo elenco di visualizzazione con i metodi `removeChild()` e `removeChildAt()`. Nell'esempio seguente vengono creati tre componenti Button sovrapposti nello stage e viene aggiunto un listener di eventi per ognuno di essi. Quando l'utente fa clic su ogni componente Button, questo viene rimosso dal gestore di eventi dall'elenco di visualizzazione e dallo stage.

- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate un componente Button dal pannello Componenti al pannello Libreria.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale e aggiungete il codice seguente:

```
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);  
}
```

Per una spiegazione completa dell'elenco di visualizzazione, vedete "Programmazione della visualizzazione" in *Programmazione in ActionScript 3.0*.

Operazioni con FocusManager

Quando l'utente preme il tasto Tab per spostarsi all'interno di un'applicazione Flash o fa clic all'interno di un'applicazione, la classe FocusManager determina quale componente deve essere attivato. Per attivare FocusManager, non è necessario aggiungerne un'istanza a un'applicazione o scrivere codice, a meno che non stiate creando un componente.

Se viene attivato un oggetto RadioButton, FocusManager esamina tale oggetto e tutti quelli con lo stesso valore groupName e imposta l'attivazione sull'oggetto con la proprietà selected impostata su true.

Ciascun componente Window a scelta obbligatoria contiene un'istanza FocusManager; pertanto, i controlli della finestra gestita da tale componente diventano una serie di tabulazioni distinta che impedisce a un utente di passare inavvertitamente a componenti di altre finestre premendo il tasto Tab.

FocusManager utilizza il livello di profondità (o z-order) degli elementi inclusi nel contenitore come schema di navigazione (o *ciclo di tabulazioni*) predefinito. Un utente solitamente si sposta nel ciclo di tabulazioni utilizzando il tasto Tab, con lo stato di attivazione che passa dal primo componente attivato fino all'ultimo, per ritornare di nuovo al primo. I livelli di profondità vengono impostati principalmente in base all'ordine in cui i componenti vengono trascinati nello stage. Tuttavia, lo z-order finale può essere determinato anche utilizzando i comandi *Elabora > Disponi > Porta in primo piano/Porta sullo sfondo*. Per ulteriori informazioni sui livelli di profondità, vedete [“Operazioni con l'elenco di visualizzazione”](#) a pagina 26.

Potete chiamare il metodo `setFocus()` per attivare un'istanza di componente in un'applicazione. Ad esempio, il codice seguente crea un'istanza FocusManager per il contenitore corrente (`this`) e assegna l'attivazione all'istanza Button `aButton`.

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

Potete determinare il componente attivo corrente chiamando il metodo `getFocus()` e verificare il componente successivo che verrà attivato, nel ciclo di tabulazioni, chiamando il metodo `getNextFocusManagerComponent()`. Nell'esempio seguente sono presenti sullo stage tre componenti (CheckBox, RadioButton e Button), ciascuno dei quali ha un listener che intercetta gli eventi `MouseEvent.CLICK` e `FocusEvent.MOUSE_FOCUS_CHANGE`. Quando si verifica l'evento `MouseEvent.CLICK` (ovvero quando l'utente fa clic su un componente), la funzione `showFocus()` chiama il metodo `getNextFocusManagerComponent()` per determinare quale componente, all'interno del ciclo di tabulazioni, riceverà lo stato di attivazione. Quindi, la funzione chiama il metodo `setFocus()` per attivare quel componente. Quando si verifica l'evento `FocusEvent.MOUSE_FOCUS_CHANGE`, la funzione `fc()` visualizza il nome del componente sul quale è avvenuto l'evento. Questo evento viene generato quando l'utente fa clic su un componente diverso da quello successivo all'interno del ciclo di tabulazioni.

```
// 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);
}
```

Per creare un componente Button che venga attivato quando l'utente preme Invio, impostate la proprietà `FocusManager.defaultButton` sull'istanza Button che deve corrispondere al pulsante predefinito, come nel codice seguente:

```
import fl.managers.FocusManager;

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

La classe `FocusManager` sostituisce il rettangolo di attivazione predefinito di Flash Player e disegna un rettangolo di attivazione personalizzato con gli angoli arrotondati.

Per ulteriori informazioni sulla creazione di uno schema di attivazione in un'applicazione Flash, vedete la classe `FocusManager` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#). Per creare un gestore di attivazione personalizzato, dovete creare una classe che implementa l'interfaccia `IFocusManager`. Per ulteriori informazioni, vedete l'interfaccia `IFocusManager` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Operazioni con i componenti basati su List

I componenti `List`, `DataGrid` e `TileList` ereditano tutti dalla classe base `SelectableList`. Per questo motivo, questi componenti sono considerati basati su `List`. Un componente `ComboBox` è costituito da una casella di testo e da un componente `List`, quindi è anch'esso un componente basato su `List`.

Un componente List è costituito da righe. DataGrid e TileList sono costituiti da righe e possono essere suddivisi in più colonne. L'intersezione di una riga con una colonna forma una cella. In un componente List, che è costituito da un'unica colonna di righe, ogni riga è una cella. Di seguito sono riportati i due aspetti importanti di una cella:

- I valori dei dati contenuti nelle celle sono detti voci. Una *voce* è un oggetto ActionScript usato per memorizzare le unità di informazione in un componente List. Un componente List può essere paragonato a un array, dove ogni spazio indicizzato dell'array è una voce. In un componente List, una voce è un oggetto a cui è solitamente associata una proprietà `label` che viene visualizzata e una proprietà `data` usata per la memorizzazione dei dati. Un *fornitore di dati* è un modello di dati delle voci contenute in un componente List e consente all'utente di compilare un componente basato su List semplicemente assegnandolo alla proprietà `dataProvider` del componente.
- Una cella può contenere diversi tipi di dati, ad esempio testo e immagini, clip filmato o qualsiasi classe creata dall'utente. Per questo motivo è necessario disegnare o eseguire il rendering di una cella tenendo conto del relativo contenuto. Quindi, i componenti basati su List dispongono di un *renderer di celle* per il rendering delle relative celle. Nel caso di DataGrid, ogni colonna è un oggetto `DataGridColumn` che dispone anche di una proprietà `cellRenderer` per consentire il rendering appropriato di ogni colonna in base al contenuto.

A tutti i componenti basati su List sono associate le proprietà `cellRenderer` e `dataProvider` che possono essere impostate per il caricamento e il rendering delle celle di questi componenti. Per informazioni sull'uso di queste proprietà e sulle operazioni con i componenti basati su List, vedete “[Operazioni con un componente DataProvider](#)” a pagina 31 e “[Operazioni con CellRenderer](#)” a pagina 39.

Operazioni con un componente DataProvider

Un componente `DataProvider` è un'origine dati che può essere utilizzata per fornire dati ai componenti `ComboBox`, `DataGrid`, `List` e `TileList`. A ognuna di queste classi di componenti è associata una proprietà `dataProvider` a cui potete assegnare un oggetto `DataProvider` per inserire dati nelle celle del componente. Un fornitore di dati è solitamente una raccolta di dati, come un oggetto `Array` o `XML`.

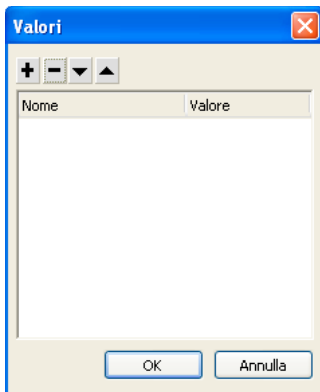
Creare un componente DataProvider

Per i componenti `ComboBox`, `List` e `TileList` potete creare un `DataProvider` usando il parametro `dataProvider` nell'ambiente di creazione. Per il componente `DataGrid` non è disponibile un parametro `dataProvider` nella finestra di ispezione Proprietà, poiché può avere più colonne e il relativo fornitore di dati è quindi più complesso. Potete usare ActionScript per creare un `DataProvider` per questi componenti, oltre che per `DataGrid`.

Usare il parametro dataProvider

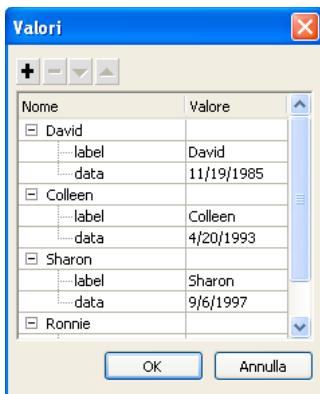
Per creare un fornitore di dati semplice per i componenti `ComboBox`, `List` e `TileList`, fate clic sul parametro `dataProvider` nella scheda Parametri della finestra di ispezione Proprietà o nella finestra di ispezione dei componenti.

Se fate doppio clic sulla cella valore, nella quale inizialmente è contenuto un oggetto `Array` vuoto, viene aperta la finestra di dialogo Valori, che consente di immettere più valori dei dati e delle etichette per creare il fornitore di dati.



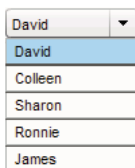
Finestra di dialogo Valori per dataProvider

Fate clic sul segno più per aggiungere una voce a dataProvider. Fate clic sul segno meno per eliminare una voce. Fate clic sulla freccia su per spostare una voce selezionata verso l'alto nell'elenco o fate clic sulla freccia giù per spostare la voce verso il basso nell'elenco. Nella figura seguente è illustrata la finestra di dialogo Valori con un elenco di nomi di bambini e i relativi compleanni.



Finestra di dialogo Valori con dati

L'oggetto Array creato è costituito da coppie di campi di etichette e di valori. I campi delle etichette sono `label` e `data` e i campi dei valori sono i nomi dei bambini e i relativi compleanni. Il campo dell'etichetta identifica il contenuto visualizzato nel componente List, in questo caso i nomi dei bambini. Il componente ComboBox risultante sarà simile al seguente:



Il componente ComboBox compilato da DataProvider

Una volta completata l'aggiunta dei dati, fate clic su OK per chiudere la finestra di dialogo. Nell'oggetto Array nel parametro dataProvider sono ora inserite le voci create.

Operazioni con i componenti

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

Parametro `dataProvider` con dati

Potete accedere ai dati delle etichette e dei valori creati usando `ActionScript` per accedere alla proprietà `dataProvider` del componente.

Creare un componente DataProvider mediante ActionScript

Potete creare un `DataProvider` creando un oggetto `Array` o `XML` e specificando l'oggetto come parametro `value` per la funzione di costruzione `DataProvider`.

Nota: in `ActionScript 3.0` non potete assegnare un oggetto `Array` o `XML` direttamente a una proprietà `dataProvider`, in quanto la proprietà è definita come un oggetto `DataProvider` e può ricevere solo un oggetto del tipo `DataProvider`.

Nell'esempio seguente, un componente `List`, che è costituito da un'unica colonna di righe, viene compilato con i nomi di alcuni bambini e i relativi compleanni. Nell'esempio, l'elenco viene definito nell'`Array items` e viene specificato come parametro quando l'istanza `DataProvider` (`new DataProvider(items)`) viene creata e assegnata alla proprietà `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);
```

L'oggetto `Array` è costituito da coppie di campi di etichette e di valori. I campi delle etichette sono `label` e `data` e i campi dei valori sono i nomi dei bambini e i relativi compleanni. Il campo dell'etichetta identifica il contenuto visualizzato nel componente `List`, in questo caso i nomi dei bambini. Il componente `List` risultante sarà simile al seguente:

David
Colleen
Sharon
Ronnie
James

Il componente `List` compilato da `DataProvider`

Il valore del campo di dati è disponibile quando l'utente seleziona una voce nell'elenco facendo clic su di essa e causando la generazione di un evento `change`. Nell'esempio seguente viene aggiunto un componente `TextArea` (`aTa`) e un gestore di eventi (`changeHandler`) all'esempio precedente per visualizzare il compleanno del bambino quando l'utente seleziona un nome nel 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);

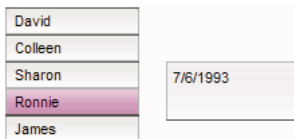
addChild(aList);
addChild(aTa);

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 questo punto, quando un utente seleziona il nome di un bambino nel componente List, nel componente TextArea viene visualizzato il relativo compleanno, come illustrato nella figura seguente. Questa operazione viene eseguita dalla funzione `changeHandler()` quando imposta la proprietà `text` del componente TextArea (`aTa.text`) sul valore del campo di dati nella voce selezionata (`event.target.selectedItem.data`). La proprietà `event.target` è l'oggetto che ha attivato l'evento, in questo caso List.



Visualizzazione del campo dei dati dall'oggetto `DataProvider` di un componente `List`

In un oggetto `DataProvider` potete includere dati diversi dal testo. Nell'esempio seguente vengono inclusi clip filmato in un `DataProvider` che fornisce i dati a `TileList`. Per creare l'oggetto `DataProvider` viene chiamato il metodo `addItem()` che aggiunge ogni voce dopo la creazione del clip filmato, una casella colorata.

Operazioni con i componenti

```

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();
}

```

Per compilare un oggetto DataProvider, potete inoltre utilizzare dati XML, anziché un array. Ad esempio, il codice seguente memorizza i dati in un oggetto XML di nome `employeesXML`, quindi passa l'oggetto come parametro `value` della funzione di costruzione di `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;

```

Potete fornire dati come attributi dei dati XML, come nel codice precedente, o come proprietà dei dati XML, come nel codice riportato di seguito:

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

L'oggetto `DataProvider` ha inoltre un set di metodi e proprietà che ne consentono l'accesso e la gestione. Potete utilizzare l'API `DataProvider` per aggiungere, rimuovere, sostituire, ordinare e unire voci in un oggetto `DataProvider`.

Gestione di un componente `DataProvider`

Potete aggiungere voci a un oggetto `DataProvider` con i metodi `addItem()` e `addItemAt()`. Nell'esempio riportato di seguito vengono aggiunte voci inserite dall'utente in un campo di testo di un componente `ComboBox` modificabile. Si presuppone che un componente `ComboBox` sia stato trascinato nello stage e che sia stato assegnato il nome di istanza `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);
}
```

Potete inoltre sostituire e rimuovere voci in un componente tramite il relativo `DataProvider`. Nell'esempio seguente vengono implementati due componenti `List` distinti, `listA` e `listB` e viene fornito un componente `Button` con l'etichetta `Sync`. Quando l'utente fa clic sul componente `Button`, nell'esempio viene utilizzato il metodo `replaceItemAt()` per sostituire le voci in `listB` con le voci in `listA`. Se la lunghezza di `listA` è maggiore di `listB`, nell'esempio viene chiamato il metodo `addItem()` per aggiungere voci supplementari a `listB`. Se la lunghezza di `listB` è maggiore di `listA`, nell'esempio viene chiamato il metodo `removeItemAt()` per rimuovere le voci supplementari da `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++);
    }
}
}
```

Potete inoltre unire e ordinare un oggetto `DataProvider` utilizzando i metodi `merge()`, `sort()` e `sortOn()`. Nell'esempio seguente vengono compilate due istanze di `DataGrid` (`aDg` e `bDg`) con liste parziali per due squadre di softball. Viene aggiunto un componente `Button` con l'etichetta `Merge` e, quando l'utente fa clic su di esso, il gestore di eventi (`mrgHandler`) unisce la lista per `bDg` con la lista per `aDg`, quindi ordina l'oggetto `DataGrid` risultante nella colonna `Name`.

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

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

Operazioni con i componenti

```
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");
}
```

Per ulteriori informazioni, vedete la classe `DataProvider` nella [Guida di riferimento di ActionScript 3.0](#).

Operazioni con CellRenderer

`CellRenderer` è una classe con componenti basati su `List` (ad esempio, `List`, `DataGrid`, `TileList` e `ComboBox`), utilizzata per gestire e visualizzare il contenuto delle celle personalizzate per le righe dell'elenco. Una cella personalizzata può contenere del testo, un componente predefinito come `CheckBox` oppure qualsiasi classe di oggetti di visualizzazione creata. Per il rendering dei dati utilizzando una classe `CellRenderer` personalizzata, potete estendere la classe `CellRenderer` o implementare l'interfaccia `ICellRenderer` per creare una classe `CellRenderer` personalizzata.

Le classi `List`, `DataGrid`, `TileList` e `ComboBox` sono sottoclassi della classe `SelectableList`. Nella classe `SelectableList` è incluso uno stile `cellRenderer` che definisce l'oggetto di visualizzazione usato dal componente per il rendering delle celle.

Potete modificare la formattazione degli stili usati da `CellRenderer` chiamando il metodo `setRendererStyle()` dell'oggetto `List` (vedete “[Formattare le celle](#)” a pagina 39). Oppure potete definire una classe personalizzata da usare come `CellRenderer` (vedete “[Definire una classe CellRenderer personalizzata](#)” a pagina 40).

Formattare le celle

La classe `CellRenderer` include numerosi stili per il controllo del formato delle celle.

Gli stili riportati di seguito consentono di definire gli skin usati per i diversi stati della cella (disattivato, giù, sopra e su):

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

Gli stili seguenti vengono applicati alla formattazione del testo:

- `disabledTextFormat`

- `textFormat`
- `textPadding`

Potete impostare questi stili chiamando il metodo `setRendererStyle()` dell'oggetto `List` o il metodo `setStyle()` dell'oggetto `CellRenderer`. Potete ottenere questi stili chiamando il metodo `getRendererStyle()` dell'oggetto `List` o il metodo `getStyle()` dell'oggetto `CellRenderer`. Inoltre, potete accedere a un oggetto che definisce tutti gli stili del renderer (come proprietà con nome dell'oggetto) tramite il metodo `rendererStyles` dell'oggetto `List` o il metodo `getStyleDefinition()` dell'oggetto `CellRenderer`.

Per ripristinare il valore predefinito di uno stile, potete chiamare il metodo `clearRendererStyle()`.

Per ottenere o impostare l'altezza delle righe nell'elenco, usate la proprietà `rowHeight` dell'oggetto `List`.

Definire una classe `CellRenderer` personalizzata

Creare una classe che estende la classe `CellRenderer` per definire un oggetto `CellRenderer` personalizzato

Il codice seguente, ad esempio, include due classi: la classe `ListSample` crea un'istanza di un componente `List` e usa l'altra classe, `CustomRenderer`, per definire il renderer di celle da usare per il componente `List`; la classe `CustomRenderer` estende la classe `CellRenderer`.

- 1 Selezionate `File > Nuovo`.
- 2 Nella finestra di dialogo `Nuovo documento` che viene visualizzata, selezionate `File Flash (ActionScript 3.0)` e fate clic su `OK`.
- 3 Selezionate `Finestra > Componenti` per visualizzare il pannello `Componenti`.
- 4 Dal pannello `Componenti`, trascinate un componente `List` nello stage.
- 5 Se la finestra di ispezione `Proprietà` non è visualizzata, selezionate `Finestra > Proprietà > Proprietà`.
- 6 Con il componente `List` selezionato, impostate le proprietà nella finestra di ispezione `Proprietà`:
 - Nome istanza: `myList`
 - W (larghezza): `200`
 - H (altezza): `300`
 - X: `20`
 - Y: `20`
- 7 Selezionate il fotogramma `1` nel livello `1` della linea temporale, quindi selezionate `Finestra > Azioni`.
- 8 Digitate lo script riportato di seguito nel pannello `Azioni`:

```
myList.setStyle("cellRenderer", CustomCellRenderer);
myList.addItem({label:"Burger -- $5.95"});
myList.addItem({label:"Fries -- $1.95"});
```
- 9 Selezionate `File > Salva`. Assegnate un nome al file e fate clic sul pulsante `OK`.
- 10 Selezionate `File > Nuovo`.
- 11 Nella finestra di dialogo `Nuovo documento` che viene visualizzata, selezionate `File ActionScript` e fate clic sul pulsante `OK`.
- 12 Nella finestra `Script`, inserite il codice seguente per definire la classe `CustomCellRenderer`:

```

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

```

13 Selezionate File > Salva. Assegnate al file il nome CustomCellRenderer.as, inseritelo nella stessa directory del file FLA, quindi fate clic sul pulsante OK.

14 Selezionate Controllo > Prova filmato.

Usare una classe che implementa l'interfaccia ICellRenderer per definire un oggetto CellRenderer personalizzato

Potete anche definire un oggetto CellRenderer usando una classe qualsiasi che erediti la classe DisplayObject e implementi l'interfaccia ICellRenderer. Il codice seguente, ad esempio, definisce due classi: la classe ListSample2 aggiunge un oggetto List all'elenco di visualizzazione e definisce il relativo CellRenderer affinché usi la classe CustomRenderer; la classe CustomRenderer estende la classe CheckBox (che a sua volta estende la classe DisplayObject) e implementa l'interfaccia ICellRenderer. La classe CustomRenderer definisce i metodi getter e setter per le proprietà data e listData, definiti nell'interfaccia ICellRenderer. Altre proprietà e metodi definiti nell'interfaccia ICellRenderer (la proprietà selected e il metodo setSize()) sono già definiti nella classe CheckBox:

1 Selezionate File > Nuovo.

2 Nella finestra di dialogo Nuovo documento che viene visualizzata, selezionate File Flash (ActionScript 3.0) e fate clic su OK.

3 Selezionate Finestra > Componenti per visualizzare il pannello Componenti.

4 Dal pannello Componenti, trascinate un componente List nello stage.

5 Se la finestra di ispezione Proprietà non è visualizzata, selezionate Finestra > Proprietà > Proprietà.

6 Con il componente List selezionato, impostate le proprietà nella finestra di ispezione Proprietà:

- Nome istanza: myList
- W (larghezza): 100
- H (altezza): 300
- X: 20
- Y: 20

7 Selezionate il fotogramma 1 nel livello 1 della linea temporale, quindi selezionate Finestra > Azioni.

8 Digitate lo script riportato di seguito nel pannello Azioni:

```

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

```

9 Selezionate File > Salva. Assegnate un nome al file e fate clic sul pulsante OK.

10 Selezionate File > Nuovo.

11 Nella finestra di dialogo Nuovo documento che viene visualizzata, selezionate File ActionScript e fate clic sul pulsante OK.

12 Nella finestra Script, inserite il codice seguente per definire la classe CustomCellRenderer:

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

13 Selezionate File > Salva. Assegnate al file il nome CustomCellRenderer.as, inseritelo nella stessa directory del file FLA, quindi fate clic sul pulsante OK.

14 Selezionate Controllo > Prova filmato.

Usare un simbolo della libreria per definire un oggetto CellRenderer

Per definire un oggetto CellRenderer, potete anche usare un simbolo della libreria. Il simbolo deve essere esportato per ActionScript e al nome della classe per il simbolo della libreria deve essere associato un file di classe che implementa l'interfaccia ICellRenderer o che estende la classe CellRenderer (o una delle relative sottoclassi).

Nell'esempio seguente viene definito un oggetto CellRenderer personalizzato usando un simbolo della libreria.

- 1 Selezionate File > Nuovo.
- 2 Nella finestra di dialogo Nuovo documento che viene visualizzata, selezionate File Flash (ActionScript 3.0) e fate clic su OK.
- 3 Selezionate Finestra > Componenti per visualizzare il pannello Componenti.
- 4 Dal pannello Componenti, trascinate un componente List nello stage.
- 5 Se la finestra di ispezione Proprietà non è visualizzata, selezionate Finestra > Proprietà > Proprietà.
- 6 Con il componente List selezionato, impostate le proprietà nella finestra di ispezione Proprietà:
 - Nome istanza: myList
 - W (larghezza): 100
 - H (altezza): 400

- X: 20
 - Y: 20
- 7 Nel pannello Parametri, fate doppio clic sulla seconda colonna della riga dataProvider.
 - 8 Nella finestra di dialogo Valori che viene visualizzata, fate clic sul segno più due volte per aggiungere due elementi di dati, con le etichette impostate su label0 e label1, quindi fate clic sul pulsante OK.
 - 9 Con lo strumento Testo, disegnate un campo di testo sullo stage.
 - 10 Con il campo di testo selezionato, impostate le proprietà nella finestra di ispezione Proprietà:
 - Tipo di testo: testo dinamico
 - Nome istanza: textField
 - W (larghezza): 100
 - Dimensione carattere: 24
 - X: 0
 - Y: 0
 - 11 Con il campo di testo selezionato, scegliete **Elabora >Converti in simbolo**.
 - 12 Nella finestra di dialogo **Converti in simbolo**, configurate le seguenti impostazioni e fate clic su OK.
 - Nome: MyCellRenderere
 - Tipo: MovieClip
 - Esporta per ActionScript: selezionato
 - Esporta nel primo fotogramma: selezionato
 - Classe: MyCellRenderere
 - Classe base: flash.display.MovieClip

Se viene visualizzato un Avviso classe ActionScript, fate clic sul pulsante OK nella finestra di avvertenza.
 - 13 Eliminate l'istanza del nuovo simbolo di clip filmato dallo stage.
 - 14 Selezionate il fotogramma 1 nel livello 1 della linea temporale, quindi selezionate **Finestra > Azioni**.
 - 15 Digitate lo script riportato di seguito nel pannello Azioni:

```
myList.setStyle("cellRenderere", MyCellRenderere);
```
 - 16 Selezionate **File > Salva**. Assegnate un nome al file e fate clic sul pulsante OK.
 - 17 Selezionate **File > Nuovo**.
 - 18 Nella finestra di dialogo **Nuovo documento** che viene visualizzata, selezionate **File ActionScript** e fate clic sul pulsante OK.
 - 19 Nella finestra **Script**, inserite il codice seguente per definire la classe MyCellRenderere:

```
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 Selezionate File > Salva. Assegnate al file il nome MyCellRenderer.as, inseritelo nella stessa directory del file FLA, quindi fate clic sul pulsante OK.

21 Selezionate Controllo > Prova filmato.

Proprietà di CellRenderer

La proprietà `data` è un oggetto che contiene tutte le proprietà impostate per `CellRenderer`. Ad esempio, nella classe seguente, che definisce un oggetto `CellRenderer` personalizzato che estende la classe `CheckBox`, la funzione `setter` per la proprietà `data` passa il valore di `data.label` alla proprietà `label` ereditata dalla classe `CheckBox`:

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

La proprietà `selected` definisce se una cella nell'elenco è selezionata o meno.

Applicare un oggetto CellRenderer per una colonna di un oggetto DataGrid

Un oggetto `DataGrid` può avere più colonne ed è possibile specificare diversi `renderer` di celle per ogni colonna. Ogni colonna di un oggetto `DataGrid` è rappresentata da un oggetto `DataGridColumn` e la classe `DataGridColumn` include una proprietà `cellRenderer` per cui viene definito l'oggetto `CellRenderer` per la colonna.

Definire un oggetto CellRenderer per una cella modificabile

La classe `DataGridCellEditor` definisce un `renderer` usato per le celle modificabili in un oggetto `DataGrid`, che diventa il `renderer` per una cella quando la proprietà `editable` dell'oggetto `DataGrid` è impostata su `true` e l'utente fa clic sulla cella da modificare. Per definire un oggetto `CellRenderer` per una cella modificabile, impostate la proprietà `itemEditor` per ogni elemento dell'array `columns` dell'oggetto `DataGrid`.

Usare immagini, file SWF o clip filmato come CellRenderer

La classe `ImageCell`, una sottoclasse di `CellRenderer`, definisce un oggetto usato per il rendering delle celle in cui il contenuto principale della cella è un'immagine, un file SWF o un clip filmato. La classe `ImageCell` include i seguenti stili per definire l'aspetto della cella:

- `imagePadding`— Il riempimento che separa il bordo della cella dal bordo dell'immagine, in pixel
- `selectedSkin`— Lo skin usato per indicare lo stato selezionato
- `textOverlayAlpha`— L'opacità della sovrapposizione dietro l'etichetta della cella
- `textPadding`— Il riempimento che separa il bordo della cella dal bordo del testo, in pixel

La classe `ImageCell` è l'oggetto `CellRenderer` predefinito per la classe `TileList`.

Rendere accessibili i componenti

Per rendere accessibile il contenuto visivo delle applicazioni Flash agli utenti ipovedenti, potete usare uno screen reader che fornisce una descrizione sonora del contenuto nelle schermate. Per informazioni su come rendere accessibile l'applicazione Flash per uno screen reader, vedete il Capitolo 18, "Creazione di contenuto accessibile", in *Usa di Flash*.

Per rendere un componente ActionScript 3.0 accessibile per uno screen reader, dovete anche importare la relativa classe di accessibilità e chiamare il metodo `enableAccessibility()` di tale classe. I seguenti componenti ActionScript 3.0 possono essere resi accessibili per uno screen reader:

Componente	Classe di accessibilità
Button	ButtonAccImpl
CheckBox	CheckBoxAccImpl
ComboBox	ComboBoxAccImpl
List	ListAccImpl
RadioButton	RadioButtonAccImpl
TileList	TileListAccImpl

Le classi di accessibilità dei componenti sono disponibili nel pacchetto `fl.accessibility`. Per rendere accessibile ad esempio un componente `CheckBox` per uno screen reader, aggiungete all'applicazione le istruzioni riportate di seguito:

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

L'accessibilità per il componente viene attivata una sola volta, indipendentemente dal numero di istanze del componente.

Nota: *l'attivazione dell'accessibilità aumenta leggermente le dimensioni del file includendo le classi necessarie durante la compilazione.*

La maggior parte dei componenti è inoltre navigabile tramite la tastiera. Per ulteriori informazioni sull'attivazione di componenti accessibili e sulla navigazione con la tastiera, vedete le sezioni relative all'interazione dell'utente in ["Uso dei componenti UI"](#) a pagina 47 e le classi di accessibilità nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Capitolo 4: Uso dei componenti UI

In questo capitolo viene illustrato come usare i componenti dell'interfaccia utente (UI) ActionScript 3.0 inclusi in Flash.

Usare il componente Button

Il componente Button è un pulsante di forma rettangolare ridimensionabile che l'utente può premere usando il mouse o la barra spaziatrice per avviare un'azione all'interno dell'applicazione. Potete aggiungere un'icona personalizzata a un componente Button e modificarne il comportamento da pulsante di comando a pulsante di tipo premuto/non premuto. Un pulsante di tipo premuto/non premuto rimane premuto quando viene scelto e ritorna nello stato normale (non premuto) se viene scelto nuovamente.

Il componente Button è una parte fondamentale di qualsiasi form e applicazione Web. È possibile utilizzare i pulsanti ovunque si desideri che l'utente esegua un'operazione. Ad esempio, la maggior parte dei form dispone di un pulsante Invia. Un altro esempio è quello dei pulsanti Avanti e Indietro disponibili nelle presentazioni.

Interazione dell'utente con il componente Button

Un pulsante può essere abilitato o disabilitato in un'applicazione. Nello stato disabilitato, non può ricevere input dal mouse o dalla tastiera. Un pulsante abilitato invece può ricevere input dal mouse o dalla tastiera. Quando un'istanza Button è attiva, è possibile utilizzare i seguenti tasti per controllarla:

Chiave	Descrizione
Maiusc+Tab	Attiva l'oggetto precedente.
Barra spaziatrice	Preme o rilascia il pulsante e attiva l'evento <code>click</code> .
Tab	Attiva l'oggetto successivo.
Invio	Attiva l'oggetto successivo se un pulsante è impostato come predefinito per FocusManager.

Per ulteriori informazioni sul controllo dell'attivazione, vedete l'interfaccia IFocusManager e la classe FocusManager nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#) e “Operazioni con FocusManager” a pagina 29.

Un'anteprima dal vivo di ogni istanza Button riflette le modifiche apportate ai parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti durante la creazione.

Nota: se le dimensioni dell'icona sono maggiori di quelle del pulsante, l'icona si estende oltre i bordi del pulsante.

Per definire un pulsante come pulsante di comando predefinito di un'applicazione, ovvero il pulsante che riceve l'evento click quando l'utente preme Invio, impostate `FocusManager.defaultButton`. Ad esempio, nel codice seguente viene impostata come pulsante predefinito un'istanza Button chiamata `submitButton`.

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

Quando aggiungete il componente Button a un'applicazione, potete renderlo accessibile a uno screen reader aggiungendo le seguenti righe di codice ActionScript:

```
import fl.accessibility.ButtonAccImpl;

ButtonAccImpl.enableAccessibility();
```

L'accessibilità per il componente viene attivata una sola volta, indipendentemente dal numero di istanze del componente.

Parametri del componente Button

Potete impostare i seguenti parametri di creazione nella finestra di ispezione Proprietà (Finestra > Proprietà > Proprietà) o nella finestra di ispezione dei componenti (Finestra > Finestra di ispezione dei componenti) per ogni istanza Button: `emphasized`, `label`, `labelPlacement`, `selected` e `toggle`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Quando assegnate un valore a questi parametri, impostate lo stato iniziale della proprietà nell'applicazione. Se impostate la proprietà in ActionScript, il valore impostato nel parametro viene ignorato. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe Button nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente Button

La procedura seguente indica come aggiungere un componente Button a un'applicazione durante la creazione. In questo esempio, il componente Button modifica lo stato di un componente ColorPicker quando fate clic su di esso.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente Button dal pannello Componenti allo stage e immettete i seguenti valori per tale componente nella finestra di ispezione Proprietà:
 - Inserite il nome di istanza **aButton**.
 - Inserite **Show** per il parametro label.
- 3 Aggiungete un componente ColorPicker allo stage e assegnategli il nome di istanza **aCp**.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 seconda riga di codice consente di registrare `clickHandler()` come funzione di gestione degli eventi per l'evento `MouseEvent.CLICK`. L'evento si verifica quando l'utente fa clic su `Button`, causando l'esecuzione di una delle seguenti azioni da parte della funzione `clickHandler()`, a seconda del valore di `Button`:

- “Show” rende visibile il componente `ColorPicker` e cambia l'etichetta di `Button` in “Disable”.
- “Disable” disattiva il componente `ColorPicker` e cambia l'etichetta di `Button` in “Enable”.
- “Enable” attiva il componente `ColorPicker` e cambia l'etichetta di `Button` in “Hide”.
- “Hide” rende invisibile il componente `ColorPicker` e cambia l'etichetta di `Button` in “Show”.

5 Selezionate **Controllo > Prova filmato** per eseguire l'applicazione.

Creare un'applicazione con il componente Button

Con la procedura seguente viene creato un componente `Button` di tipo premuto/non premuto usando `ActionScript` e viene visualizzato il tipo di evento nel pannello `Output` quando si fa clic su `Button`. Nell'esempio viene creata l'istanza `Button` richiamando la funzione di costruzione della classe; l'istanza viene quindi aggiunta allo stage chiamando il metodo `addChild()`.

- 1 Create un nuovo documento Flash (`ActionScript 3.0`).
- 2 Trascinate il componente `Button` dal pannello `Componenti` al pannello `Libreria` del documento corrente.
Questa operazione aggiunge il componente alla libreria senza renderlo visibile nell'applicazione.
- 3 Aprite il pannello `Azioni`, selezionate il fotogramma 1 nella linea temporale principale, quindi inserite il codice seguente per creare un'istanza `Button`:

```
import fl.controls.Button;

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

Il metodo `move()` consente di posizionare il pulsante in corrispondenza della posizione 50 (coordinata x), 50 (coordinata y) nello stage.

- 4 A questo punto, aggiungere il codice `ActionScript` seguente per creare un listener di eventi e una funzione di gestione degli eventi:

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

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

- 5 Selezionare **Controllo > Prova filmato**.

Quando fate clic sul pulsante, viene visualizzato il messaggio “Event type: click” nel pannello `Output`.

Usare il componente CheckBox

Un componente `CheckBox` è una casella quadrata che può essere selezionata o deselegionata. Quando è selezionata, nella casella è visualizzato un segno di spunta. Al componente `CheckBox` potete aggiungere un'etichetta di testo a sinistra, a destra, sopra o sotto.

Potete usare i componenti CheckBox per ottenere una serie di valori `true` o `false` che non si escludano a vicenda. Ad esempio, in un'applicazione che raccoglie informazioni sul tipo di auto che si desidera acquistare, possono essere usati componenti CheckBox per la selezione delle caratteristiche.

Interazione dell'utente con il componente CheckBox

Un componente CheckBox può essere abilitato o disabilitato in un'applicazione. Se è abilitato e l'utente fa clic su di esso o sulla relativa etichetta, il componente viene attivato e visualizza l'aspetto corrispondente allo stato premuto. Se l'utente sposta il puntatore fuori dall'area di delimitazione del componente CheckBox o dall'etichetta tenendo premuto contemporaneamente il pulsante del mouse, l'aspetto del componente torna allo stato originale e rimane attivo. Lo stato di un componente CheckBox non cambia finché non si rilascia il pulsante del mouse. Inoltre, questo componente ha due stati disabilitato, selezionato e deselezionato, che usano rispettivamente `selectedDisabledSkin` e `disabledSkin` e non consentono alcuna interazione mediante il mouse o la tastiera.

Se un componente CheckBox è disabilitato, tale stato viene visualizzato indipendentemente dall'azione dell'utente. Quando è disattivato, un componente CheckBox non può ricevere input dal mouse o dalla tastiera.

Un'istanza CheckBox viene attivata se un utente fa clic su di essa oppure se si sposta su di essa con il tasto di tabulazione. Una volta attivata l'istanza CheckBox, è possibile utilizzare i seguenti tasti per controllarla:

Chiave	Descrizione
Maiusc+Tab	Attiva l'elemento precedente.
Barra spaziatrice	Seleziona o deseleziona il componente e attiva l'evento <code>change</code> .
Tab	Attiva l'elemento successivo.

Per ulteriori informazioni sul controllo dell'attivazione, vedete “Operazioni con FocusManager” a pagina 29 e la classe FocusManager nella *Guida di riferimento di ActionScript 3.0 per Flash Professional*.

Un'anteprima dal vivo di ogni istanza CheckBox riflette le modifiche apportate ai parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti durante la creazione.

Quando aggiungete il componente CheckBox a un'applicazione, potete renderlo accessibile a uno screen reader aggiungendo le seguenti righe di codice ActionScript:

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

Attivare l'accessibilità per il componente una sola volta, indipendentemente dal numero di istanze del componente.

Parametri del componente CheckBox

I seguenti parametri di creazione possono essere impostati per ogni istanza del componente CheckBox: `label`, `labelPlacement` e `selected`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe CheckBox nella *Guida di riferimento di ActionScript 3.0 per Flash Professional*.

Creare un'applicazione mediante il componente CheckBox

Nella procedura seguente viene descritto come aggiungere un componente CheckBox a un'applicazione durante la creazione, usando un estratto dal form di un'applicazione di gestione prestiti. Nel form viene chiesto se il richiedente è proprietario di una casa ed è incluso un componente CheckBox mediante il quale l'utente può rispondere affermativamente (“yes”). In caso di risposta affermativa, vengono presentati due pulsanti di scelta che l'utente userà per indicare il valore relativo della casa.

Creare un'applicazione mediante il componente CheckBox

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinare il componente CheckBox dal pannello Componenti allo stage.
- 3 Nella finestra di ispezione Proprietà, effettuare le operazioni seguenti:
 - Inserite **homeCh** come nome dell'istanza.
 - Inserite **140** come valore della larghezza (W).
 - Inserite “**Own your home?**” per il parametro label.
- 4 Trascinate due componenti RadioButton dal pannello Componenti allo stage e posizionatele sotto e a destra del componente CheckBox. Immettete i seguenti valori per tali componenti nella finestra di ispezione Proprietà:
 - Inserite **underRb** e **overRb** come nomi di istanza.
 - Inserite **120** per il parametro W (larghezza) di entrambi i componenti RadioButton.
 - Immettete **Under \$500,000?** per il parametro label di **underRb**.
 - Immettete **Over \$500,000?** per il parametro label di **overRb**.
 - Inserite **valueGrp** per il parametro groupName di entrambi i componenti RadioButton.
- 5 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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;
}
```

Questo codice crea un gestore di eventi per un evento `CLICK` che attiva i componenti RadioButton `underRb` e `overRb` se il componente CheckBox `homeCh` è selezionato e li disattiva se `homeCh` non è selezionato. Per ulteriori informazioni, vedete la classe `MouseEvent` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

- 6 Selezionare Controllo > Prova filmato.

Nell'esempio seguente viene duplicata l'applicazione precedente, creando invece i componenti CheckBox e RadioButton con ActionScript.

Creare un componente CheckBox mediante ActionScript

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente CheckBox e il componente RadioButton dal pannello Componenti al pannello Libreria del documento corrente. Se il pannello Libreria non è aperto, premete `Ctrl+L` o selezionate Finestra > Libreria per aprirlo.

In questo modo i componenti sono disponibili per l'applicazione, ma non vengono collocati nello stage.

- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi inserite il codice seguente per creare e posizionare le istanze dei componenti:

```
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?";
```

Questo codice usa le funzioni di costruzione `CheckBox()` e `RadioButton()` per creare i componenti e il metodo `addChild()` per inserirli nello stage, mentre per posizzarli sullo stage usa il metodo `move()`.

- 4 A questo punto, aggiungere il codice ActionScript seguente per creare un listener di eventi e una funzione di gestione degli eventi:

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

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

Questo codice crea un gestore di eventi per l'evento `CLICK` che attiva i pulsanti di scelta `underRb` e `overRb` se il componente `CheckBox` `homeCh` è selezionato e li disattiva se `homeCh` non è selezionato. Per ulteriori informazioni, vedete la classe `MouseEvent` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

- 5 Selezionare Controllo > Prova filmato.

Usare il componente `ColorPicker`

Il componente `ColorPicker` consente all'utente di selezionare un colore da un elenco di campioni. La modalità predefinita del componente `ColorPicker` mostra un singolo colore all'interno di un pulsante quadrato. Quando l'utente fa clic sul pulsante, viene visualizzato l'elenco dei colori disponibili in un pannello di campioni insieme a un campo di testo con il valore esadecimale del colore attualmente selezionato.

Per impostare i colori che saranno visualizzati nel componente `ColorPicker`, impostate la relativa proprietà `colors` con i valori di colore che desiderate visualizzare.

Interazione dell'utente con il componente ColorPicker

Un componente ColorPicker consente all'utente di selezionare un colore e di applicarlo a un altro oggetto nell'applicazione. Ad esempio, se desiderate consentire all'utente di personalizzare gli elementi dell'applicazione, come il colore dello sfondo o del testo, potete includere un componente ColorPicker e applicare il colore selezionato dall'utente.

L'utente sceglie un colore selezionando il relativo campione nel pannello o inserendo il valore esadecimale del colore nel campo di testo. Dopo che l'utente ha selezionato un colore, potete usare la proprietà `selectedColor` di ColorPicker per applicare il colore al testo o a un altro oggetto nell'applicazione.

Un'istanza ColorPicker viene attivata quando l'utente si sposta su di essa mediante il puntatore del mouse o il tasto Tab. Quando il pannello dei campioni di un componente ColorPicker è aperto, potete usare i seguenti tasti per controllarlo:

Chiave	Descrizione
Home	Sposta la selezione al primo colore nel pannello dei campioni.
Freccia su	Sposta la selezione di una riga verso l'alto nel pannello dei campioni.
Freccia giù	Sposta la selezione di una riga verso il basso nel pannello dei campioni.
Freccia destra	Sposta la selezione di un colore verso destra nel pannello dei campioni.
Freccia sinistra	Sposta la selezione di un colore verso sinistra nel pannello dei campioni.
Fine	Sposta la selezione all'ultimo colore nel pannello dei campioni.

Parametri del componente ColorPicker

Per ogni istanza ColorPicker potete impostare i parametri di creazione `selectedColor` e `showTextField` nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe ColorPicker nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente ColorPicker

Nell'esempio seguente viene aggiunto un componente ColorPicker a un'applicazione durante la creazione. In questo esempio, ogni volta che cambiate il colore nel componente ColorPicker, la funzione `changeHandler()` chiama la funzione `drawBox()` per disegnare una nuova casella con il colore selezionato in ColorPicker.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente ColorPicker dal pannello Componenti al centro dello stage e assegnategli il nome di istanza **aCp**.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionare Controllo > Prova filmato.
- 5 Fate clic sul componente ColorPicker e selezionate un colore da applicare alla casella.

Creare un componente ColorPicker mediante ActionScript

Nell'esempio seguente vengono usati la funzione di costruzione `ColorPicker()` e il metodo `addChild()` per creare un componente `ColorPicker` sullo stage. Per specificare i colori che saranno visualizzati nel componente `ColorPicker`, viene impostata la proprietà `colors` sui valori di colore rosso (0xFF0000), verde (0x00FF00) e blu (0x0000FF). In questo esempio, viene inoltre creato un componente `TextArea` e ogni volta che selezionate un colore diverso in `ColorPicker`, il colore del testo nel componente `TextArea` viene modificato di conseguenza.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente `ColorPicker` dal pannello Componenti al pannello Libreria.
- 3 Trascinate il componente `TextArea` dal pannello Componenti al pannello Libreria.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

Uso dei componenti UI

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

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

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

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

addChild(aCp);
addChild(aTa);

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

5 Selezionare Controllo > Prova filmato.

Usare il componente ComboBox

Un componente ComboBox consente all'utente di effettuare una sola selezione da un elenco a discesa e può essere statico o modificabile. Un componente ComboBox modificabile consente all'utente di inserire il testo direttamente in un campo di testo presente all'inizio dell'elenco. Se l'elenco a discesa tocca il lato inferiore del documento, viene aperto verso l'alto anziché verso il basso. Il componente ComboBox è costituito da tre sottocomponenti: BaseButton, TextInput e List.

In un componente ComboBox modificabile, solo il pulsante è l'area attiva e non la casella di testo. In un componente ComboBox statico, sia il pulsante che la casella di testo costituiscono l'area attiva. L'area attiva risponde aprendo o chiudendo l'elenco a discesa.

Quando l'utente effettua una selezione nell'elenco, con il mouse o tramite la tastiera, l'etichetta della selezione viene copiata nel campo di testo presente nella parte superiore del componente ComboBox.

Interazione dell'utente con il componente ComboBox

Un componente ComboBox può essere utilizzato in qualsiasi form o applicazione che richiede una singola scelta da un elenco. Ad esempio, è possibile fornire all'utente un elenco a discesa di paesi in un form di indirizzi. Un componente ComboBox modificabile può essere utile per scenari più complessi. Ad esempio, in un'applicazione che fornisce itinerari stradali, potete usare un componente ComboBox modificabile per consentire all'utente di inserire i dati relativi agli indirizzi di partenza e di arrivo. In questo caso l'elenco a discesa conterrebbe gli indirizzi immessi in precedenza dall'utente.

Se il componente ComboBox è modificabile, ovvero la proprietà `editable` è `true`, i tasti elencati di seguito consentono di disattivare la casella di testo di input e lasciare il valore precedente. La sola eccezione è il tasto Invio, che applica il nuovo valore se l'utente ha immesso un testo.

Chiave	Descrizione
Maiusc + Tab	Attiva l'elemento precedente. Se selezionate una nuova voce, viene inviato un evento <code>change</code> .
Tab	Attiva la voce successiva. Se selezionate una nuova voce, viene inviato un evento <code>change</code> .
Freccia giù	Sposta la selezione di una voce verso il basso.
Fine	Sposta la selezione alla fine dell'elenco.
Escape	Chiude l'elenco a discesa e riattiva il componente ComboBox.
Digitare	Chiude l'elenco a discesa e riattiva il componente ComboBox. Se il componente ComboBox è modificabile e l'utente immette un testo, premendo Invio viene impostato il valore corrispondente al testo immesso.
Home	Sposta la selezione all'inizio dell'elenco.
Pagina su	Sposta la selezione di una pagina verso l'alto.
PgGiù	Sposta la selezione di una pagina verso il basso.

Quando aggiungete il componente ComboBox a un'applicazione, potete renderlo accessibile a uno screen reader aggiungendo le seguenti righe di codice ActionScript:

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

Attivare l'accessibilità per il componente una sola volta, indipendentemente dal numero di istanze del componente.

Parametri del componente ComboBox

Per ogni istanza ComboBox potete impostare i seguenti parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `dataProvider`, `editable`, `prompt` e `rowCount`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe ComboBox nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#). Per informazioni sull'uso del parametro `dataProvider`, vedete “[Usare il parametro dataProvider](#)” a pagina 31.

Creare un'applicazione con il componente ComboBox

Nella procedura seguente viene descritto come aggiungere un componente ComboBox a un'applicazione durante la creazione. Il componente ComboBox è modificabile e se digitate **Add** nel campo di testo, in questo esempio viene aggiunta una voce all'elenco a discesa.

- 1 Create un nuovo documento Flash (ActionScript 3.0).

- 2 Trascinate un componente ComboBox nello stage e assegnategli il nome di istanza **aCb**. Nella scheda Parametri, impostate il parametro `editable` su `true`.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice seguente:

```
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 Selezionare Controllo > Prova filmato.

Creare un componente ComboBox mediante ActionScript

Nell'esempio seguente viene creato un componente ComboBox mediante ActionScript, che viene compilato con un elenco di università nell'area di San Francisco in California. La proprietà `width` di ComboBox viene impostata in modo che si adatti alla larghezza del testo di richiesta, mentre la proprietà `dropdownWidth` viene impostata su una larghezza leggermente superiore per adattarla al nome di università più lungo.

Nell'esempio viene creato un elenco di università in un'istanza Array, usando la proprietà `label` per memorizzare i nomi degli istituti e la proprietà `data` per memorizzare l'URL del sito Web di ogni istituto. L'istanza Array viene assegnata al componente ComboBox mediante l'impostazione della relativa proprietà `dataProvider`.

Quando un utente seleziona un'università dall'elenco, attiva un evento `Event.CHANGE` e una chiamata alla funzione `changeHandler()`, che carica la proprietà `data` in una richiesta URL per l'accesso al sito Web della scuola

L'ultima riga imposta la proprietà `selectedIndex` dell'istanza ComboBox su `-1` in modo che la richiesta venga visualizzata di nuovo alla chiusura dell'elenco. In caso contrario, la richiesta verrebbe sostituita dal nome dell'istituto selezionato.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente ComboBox dal pannello Componenti al pannello Libreria.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionare Controllo > Prova filmato.

Potete implementare ed eseguire questo esempio nell'ambiente di creazione Flash, tuttavia se tentate di accedere ai siti Web delle università facendo clic sulle voci presenti nel componente ComboBox, verranno visualizzati dei messaggi di avviso. Per accedere al componente ComboBox completamente funzionale su Internet, visitate il collegamento riportato di seguito:

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

Usare il componente DataGrid

Il componente DataGrid consente di visualizzare i dati in una griglia di righe e colonne, caricando i dati da un array oppure da un file XML esterno che può essere convertito in un array per il DataProvider. Il componente DataGrid include lo scorrimento verticale e orizzontale, il supporto degli eventi (compreso il supporto per le celle modificabili) e le capacità di ordinamento.

In una griglia potete ridimensionare e personalizzare caratteristiche come il carattere, il colore e i bordi delle colonne. È possibile usare un clip filmato personalizzato, ad esempio un renderer di celle, per qualsiasi colonna in una griglia. Un renderer di celle visualizza il contenuto di una cella. Potete disattivare le barre di scorrimento e usare i metodi DataGrid per creare una visualizzazione dello stile della vista di una pagina. Per ulteriori informazioni sulla personalizzazione, vedete la classe DataGridColumn nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Altri argomenti presenti nell’Aiuto

[Creazione, compilazione e ridimensionamento del componente DataGrid](#)

[Personalizzazione e ordinamento del componente DataGrid](#)

[Filtraggio e formattazione dei dati nel componente DataGrid](#)

Interazione dell'utente con il componente DataGrid

È possibile usare il mouse e la tastiera per interagire con un componente DataGrid.

Se le proprietà `sortableColumnse` e `sortable` della colonna hanno entrambe il valore `true` e fate clic su un'intestazione di colonna, i dati vengono ordinati in base ai valori della colonna. Potete disattivare l'ordinamento per una singola colonna impostando la relativa proprietà `sortable` su `false`.

Se il valore della proprietà `resizableColumns` è `true`, potete ridimensionare le colonne trascinando i relativi divisori nella riga di intestazione.

Se fate clic su una cella modificabile, questa viene attivata; se fate clic su una cella non modificabile, non ottenete alcun effetto. Una singola cella è modificabile quando entrambe le proprietà `DataGrid.editable` e `DataGridColumn.editable` hanno valore `true`.

Per ulteriori informazioni, vedete le classi `DataGrid` e `DataGridColumn` nella [Guida di riferimento di ActionScript 3.0](#).

Quando un'istanza `DataGrid` viene attivata facendo clic con il mouse o premendo il tasto `Tab`, è possibile utilizzare i tasti seguenti per controllarla:

Chiave	Descrizione
Freccia giù	Quando è in corso la modifica di una cella, il punto di inserimento si sposta alla fine del testo della cella. Se una cella non è modificabile, il tasto Freccia giù gestisce la selezione nello stesso modo del componente <code>List</code> .
Freccia su	Quando è in corso la modifica di una cella, il punto di inserimento si sposta all'inizio del testo della cella. Se una cella non è modificabile, il tasto Freccia su gestisce la selezione nello stesso modo del componente <code>List</code> .
Maiusc+Freccia su/giù	Se il componente <code>DataGrid</code> non è modificabile e il valore di <code>allowMultipleSelection</code> è <code>true</code> , vengono selezionate righe contigue. L'inversione della direzione con la freccia opposta deselecta le righe selezionate, finché non si oltrepassa la riga iniziale; a quel punto vengono selezionate le righe in quella direzione.
Maiusc+Clic	Se <code>allowMultipleSelection</code> è <code>true</code> , vengono selezionate tutte le righe tra la riga selezionata e la posizione corrente del cursore (la cella evidenziata).
Ctrl+Clic	Se <code>allowMultipleSelection</code> è <code>true</code> , vengono selezionate righe aggiuntive, non necessariamente contigue.
Freccia destra	Quando è in corso la modifica di una cella, il punto di inserimento si sposta di un carattere a destra. Se una cella non è modificabile, il tasto Freccia destra non esegue alcuna operazione.
Freccia sinistra	Quando è in corso la modifica di una cella, il punto di inserimento si sposta di un carattere a sinistra. Se una cella non è modificabile, il tasto Freccia sinistra non esegue alcuna operazione.

Chiave	Descrizione
Home	Seleziona la prima riga nel componente DataGrid.
Fine	Seleziona l'ultima riga nel componente DataGrid.
Pag su	Seleziona la prima riga in una pagina del componente DataGrid. Una pagina è costituita dal numero di righe che il componente DataGrid è in grado di visualizzare senza scorrimento.
Pag giù	Seleziona l'ultima riga in una pagina del componente DataGrid. Una pagina è costituita dal numero di righe che il componente DataGrid è in grado di visualizzare senza scorrimento.
A capo/Invio/Maiusc+Invio	Quando una cella è modificabile, viene eseguito il commit della modifica e il punto di inserimento viene spostato sulla cella della stessa colonna, sulla riga successiva (su o giù, a seconda dell'uso o meno del tasto Maiusc).
Maiusc+Tab/Tab	Se il componente DataGrid è modificabile, l'attivazione si sposta alla voce precedente/successiva finché non viene raggiunta la fine della colonna, quindi alla riga precedente/successiva finché non viene raggiunta la prima o l'ultima cella. Se è selezionata la prima cella, premendo Maiusc+Tab l'attivazione si sposta al controllo precedente. Se è selezionata l'ultima cella, premendo Tab l'attivazione si sposta al controllo successivo. Se il componente DataGrid non è modificabile, l'attivazione si sposta al controllo precedente/successivo.

Il componente DataGrid può essere utilizzato come base per numerosi tipi di applicazioni basate sui dati. Potete visualizzare facilmente una vista in formato tabella dei dati, ma anche usare le funzioni del renderer di celle per costruire elementi più sofisticati e modificabili dell'interfaccia utente. Alcuni usi pratici del componente DataGrid sono i seguenti:

- Un client webmail
- Pagine dei risultati delle ricerche
- Applicazioni per fogli di calcolo, ad esempio per il calcolo delle rate di un prestito e per i moduli fiscali

Quando progettate un'applicazione con il componente DataGrid, è utile conoscere la struttura del componente List, in quanto la classe DataGrid estende la classe SelectableList. Per ulteriori informazioni sulla classe SelectableList e il componente List, vedete le classi SelectableList e List classes nella [Guida di riferimento di ActionScript 3.0](#).

Quando aggiungete un componente DataGrid all'applicazione, potete renderlo accessibile a uno screen reader aggiungendo le seguenti righe di codice ActionScript:

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

Attivare l'accessibilità per il componente una sola volta, indipendentemente dal numero di istanze del componente. Per ulteriori informazioni, vedete il Capitolo 18, "Creazione di contenuto accessibile", nella guida *Uso di Flash*.

Parametri del componente DataGrid

Per ogni istanza del componente DataGrid potete impostare i seguenti parametri di creazione nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: allowMultipleSelection, editable, headerHeight, horizontalLineScrollSize, horizontalPageScrollSize, horizontalScrollPolicy, resizableColumns, rowHeight, showHeaders, verticalLineScrollSize, verticalPageScrollSize e verticalScrollPolicy. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe DataGrid nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente DataGrid

Per creare un'applicazione con il componente DataGrid, è necessario per prima cosa determinare da dove provengono i dati. I dati provengono solitamente da un Array e possono essere inseriti in una griglia mediante l'impostazione della proprietà `dataProvider`. Potete inoltre usare i metodi delle classi DataGrid e DataGridColumn per aggiungere i dati alla griglia.

Usare un fornitore di dati locale con un componente DataGrid:

Nell'esempio viene creato un componente DataGrid per visualizzare la lista dei giocatori di una squadra di softball. La lista viene definita in un Array (`aRoster`) che viene assegnato alla proprietà `dataProvider` di DataGrid.

- 1 In Flash, selezionate File > Nuovo, quindi selezionate File Flash (ActionScript 3.0).
- 2 Trascinate il componente DataGrid dal pannello Componenti allo stage.
- 3 Nella finestra di ispezione Proprietà, inserite **aDg** come nome di istanza.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 funzione `bldRosterGrid()` imposta le dimensioni del componente DataGrid e definisce l'ordine delle colonne e le relative dimensioni.

- 5 Selezionare Controllo > Prova filmato.

Specificare le colonne e aggiungere l'ordinamento di un componente DataGrid in un'applicazione

Potete fare clic sull'intestazione di una colonna per ordinare il contenuto del componente in ordine decrescente in base ai valori di tale colonna.

Nell'esempio seguente viene usato il metodo `addColumn()` per aggiungere le istanze `DataGridColumn` a un componente `DataGrid`. Le colonne rappresentano i nomi dei giocatori e i relativi punteggi. Nell'esempio viene anche impostata la proprietà `sortOptions` per specificare le opzioni di ordinamento per ciascuna colonna:

`Array.CASEINSENSITIVE` per la colonna `Name` e `Array.NUMERIC` per la colonna `Score`. Le dimensioni di `DataGrid` vengono definite nel modo appropriato impostando la lunghezza in base al numero di colonne e la larghezza su 200.

- 1 In Flash, selezionate **File > Nuovo**, quindi selezionate **File Flash (ActionScript 3.0)**.
- 2 Trascinate il componente `DataGrid` dal pannello **Componenti** allo stage.
- 3 Nella finestra di ispezione **Proprietà**, inserite **aDg** come nome di istanza.
- 4 Aprite il pannello **Azioni**, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionare **Controllo > Prova filmato**.

Creare l'istanza di un componente DataGrid mediante ActionScript

Nell'esempio seguente viene creato un componente `DataGrid` mediante ActionScript che viene compilato con un `Array` di nomi di giocatori e punteggi.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente `DataGrid` dal pannello **Componenti** al pannello **Libreria** del documento corrente.
In questo modo, il componente viene aggiunto alla libreria, ma non è visibile nel documento.
- 3 Aprite il pannello **Azioni**, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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);
```

Questo codice crea l'istanza `DataGrid`, quindi ridimensiona e posiziona la griglia.

- 4 Create un array, aggiungete i dati e identificate l'array come fornitore di dati per il componente 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 Selezionare Controllo > Prova filmato.

Caricare un file XML in un componente DataGrid

Nell'esempio seguente viene usata la classe DataGridColumn per creare le colonne del componente DataGrid, che viene compilato passando un oggetto XML come parametro `value` della funzione di costruzione `DataProvider()`.

- 1 L'uso di un editor di testo crea un file XML con i seguenti dati e lo salva come `team.xml` nella stessa cartella in cui verrà salvato il file 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 Create un nuovo documento Flash (ActionScript 3.0).
- 3 Nel pannello Componenti, fare doppio clic sul componente DataGrid per aggiungerlo allo stage.
- 4 Nella finestra di ispezione Proprietà, inserite `aDg` come nome di istanza.
- 5 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionare Controllo > Prova filmato.

Usare il componente Label

Nel componente Label viene visualizzata un'unica riga di testo, solitamente per identificare un altro elemento o un'attività in una pagina Web. Potete specificare che un'etichetta venga formattata in HTML per sfruttare i relativi tag di formattazione del testo e controllarne l'allineamento e le dimensioni. I componenti Label non hanno bordi, non possono essere attivati e non inviano alcun evento.

Nell'anteprima dal vivo di ogni istanza del componente Label vengono visualizzate le modifiche apportate ai parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti durante la creazione. L'etichetta non ha bordi, quindi l'unico modo per visualizzarne l'anteprima consiste nell'impostare il relativo parametro text.

Interazione dell'utente con il componente Label

Usate un componente Label per creare un'etichetta di testo per un altro componente in un form, ad esempio un'etichetta "Nome:" alla sinistra di un campo TextInput che accetta il nome di un utente. Si consiglia di usare un componente Label anziché un campo di testo normale, poiché in questo modo potete usare gli stili per mantenere un aspetto coerente e uniforme nell'intera applicazione.

Per consentire la rotazione di un componente Label, dovete incorporare i caratteri; in caso contrario non verranno visualizzati quando si prova il filmato.

Parametri del componente Label

I seguenti parametri di creazione possono essere impostati per ogni istanza del componente Label: `autoSize`, `condenseWhite`, `selectable`, `text` e `wordWrap`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe Label nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente Label

La procedura seguente indica come aggiungere un componente Label a un'applicazione durante la creazione. In questo esempio, nell'etichetta viene visualizzato semplicemente il testo "Data scadenza."

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente Label dal pannello Componenti allo stage e assegnategli i seguenti valori nella finestra di ispezione Proprietà:
 - Inserite **aLabel** come nome dell'istanza.
 - Inserite **80** per il valore W.
 - Inserite **100** per il valore X.
 - Inserite **100** per il valore Y.
 - Inserite **Data scadenza** per il parametro `text`.
- 3 Trascinate un componente TextArea nello stage e assegnategli i seguenti valori nella finestra di ispezione Proprietà:
 - Inserite **aTa** come nome dell'istanza.
 - Inserite **22** per il valore H.
 - Inserite **200** per il valore X.
 - Inserite **100** per il valore Y.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionare Controllo > Prova filmato.

Creare un'istanza del componente Label mediante ActionScript

Nell'esempio seguente viene creato un componente Label mediante ActionScript. Viene usato per identificare la funzione di un componente ColorPicker e la proprietà `htmlText` per applicare la formattazione al testo di Label.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente Label dal pannello Componenti al pannello Libreria del documento corrente.
- 3 Trascinate il componente ColorPicker dal pannello Componenti al pannello Libreria del documento corrente.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionare Controllo > Prova filmato.

Usare il componente List

Il componente List è un elenco a scorrimento con possibilità di selezione singola o multipla. Un elenco può anche visualizzare immagini, compresi altri componenti. È possibile aggiungere le voci visualizzate nell'elenco tramite la finestra di dialogo Valori che viene visualizzata quando si fa clic sulle etichette o sui campi dei parametri dei dati. Potete anche usare i metodi `List.addItem()` e `List.addItemAt()` per aggiungere le voci all'elenco.

Il componente List usa un indice con base zero, in cui la voce con indice 0 è la prima voce visualizzata. Quando si aggiungono, rimuovono o sostituiscono le voci di elenco usando i metodi e le proprietà della classe List, potrebbe essere necessario specificare l'indice della voce di elenco.

Interazione dell'utente con il componente List

È possibile impostare un elenco affinché gli utenti possano eseguire selezioni singole o multiple. Ad esempio, un utente che visita un sito di commercio elettronico deve poter selezionare l'articolo da acquistare. In un elenco sono presenti 30 elementi che l'utente può scorrere e selezionare facendo clic sull'elemento stesso.

Potete anche creare un componente List che usa clip filmato personalizzati come righe, in modo che sia possibile visualizzare più informazioni all'utente. Ad esempio, in un'applicazione di posta elettronica, ogni cassetta postale può essere un componente List e ogni riga può contenere icone che indicano la priorità e lo stato.

Il componente List si attiva quando si fa clic su di esso o si seleziona con il tasto Tab. Una volta attivato, è possibile usare i tasti seguenti per controllarlo:

Chiave	Descrizione
Tasti alfanumerici	Consentono di passare alla voce successiva nella cui etichetta è presente <code>Key.getAscii()</code> come primo carattere.
Ctrl	Tasto di attivazione/disattivazione che consente di effettuare più selezioni e deselezioni non contigue.
Freccia giù	La selezione viene spostata di una voce verso il basso.
Home	La selezione viene spostata all'inizio dell'elenco.
PgGiù	La selezione viene spostata di una pagina verso il basso.
Pagina su	La selezione viene spostata di una pagina verso l'alto.
Maiusc	Consente di eseguire selezioni contigue.
Freccia su	La selezione viene spostata di una voce verso l'alto.

Nota: le dimensioni per lo scorrimento sono espresse in pixel, non in righe.

Nota: `{\cf0 I}{\cf2}` tasti Pg su e Pg giù scorrono le voci in base al numero delle voci che lo schermo riesce a visualizzare meno una.} Ad esempio, se premete il tasto Pg giù in un elenco a discesa composto da dieci righe, vengono visualizzate le voci 0-9, 9-18, 18-27 e così via, con una voce sovrapposta per pagina.

Per ulteriori informazioni sul controllo dell'attivazione, vedete l'interfaccia `IFocusManager` e la classe `FocusManager` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#) e “Operazioni con `FocusManager`” a pagina 29.

Sullo stage, un'anteprima dal vivo di ogni istanza del componente List consente di visualizzare le modifiche apportate ai parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti durante la creazione.

Quando aggiungete il componente List a un'applicazione, potete renderlo accessibile a uno screen reader aggiungendo le seguenti righe di codice ActionScript:

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

Attivare l'accessibilità per il componente una sola volta, indipendentemente dal numero di istanze del componente. Per ulteriori informazioni, vedete il Capitolo 18, “Creazione di contenuto accessibile”, nella guida *Uso di Flash*.

Parametri del componente List

I seguenti parametri possono essere impostati per ogni istanza del componente List nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `allowMultipleSelection`, `dataProvider`, `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `multipleSelection`, `verticalLineScrollSize`, `verticalPageScrollSize` e `verticalScrollPolicy`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe List nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#). Per informazioni sull'uso del parametro `dataProvider`, vedete “Usare il parametro `dataProvider`” a pagina 31.

Creare un'applicazione con il componente List

Gli esempi seguenti indicano come aggiungere un componente List a un'applicazione durante la creazione.

Aggiungere un semplice componente List a un'applicazione

In questo esempio, List è composto da etichette che identificano i modelli di auto e da campi di dati contenenti i prezzi.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinare un componente List nello stage dal pannello Componenti.
- 3 Nella finestra di ispezione Proprietà, effettuare le operazioni seguenti:
 - Inserite il nome di istanza **aList**.
 - Assegnate il valore **200** per la larghezza (W).
- 4 Usate lo strumento Testo per creare un campo di testo sotto **aList** e assegnategli il nome di istanza **aTf**.
- 5 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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;
}
```

Questo codice utilizza il metodo `addItem()` per compilare l'istanza `aList` con tre voci, assegnando a ciascuna un valore `label`, visualizzato nell'elenco, e un valore `data`. Quando l'utente seleziona una voce nel componente List, il listener di eventi chiama la funzione `showData()`, che visualizza il valore `data` per la voce selezionata.

- 6 Selezionate Controllo > Prova filmato per compilare ed eseguire l'applicazione.

Compilare un'istanza List con un fornitore di dati

Nell'esempio seguente viene creato un componente List per i modelli di auto e i relativi prezzi. Per compilare List viene invece usato un fornitore di dati, anziché il metodo `addItem()`.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinare un componente List nello stage dal pannello Componenti.
- 3 Nella finestra di ispezione Proprietà, effettuare le operazioni seguenti:
 - Inserite il nome di istanza **aList**.
 - Assegnate il valore **200** per la larghezza (W).
- 4 Usate lo strumento Testo per creare un campo di testo sotto **aList** e assegnategli il nome di istanza **aTf**.
- 5 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:


```
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 Selezionate Controllo > Prova filmato per visualizzare il componente List con le relative voci.

Usare un componente List per controllare un'istanza di un clip filmato

Nell'esempio seguente viene creato un componente List di nomi di colore e quando l'utente ne seleziona uno, il colore viene applicato a un clip filmato.

- 1 Create un documento Flash (ActionScript 3.0).
- 2 Trascinate un componente List dal pannello Componenti allo stage e assegnategli i seguenti valori nella finestra di ispezione Proprietà:
 - Inserite **aList** come nome dell'istanza.
 - Inserite **60** per il valore H.
 - Inserite **100** per il valore X.
 - Inserite **150** per il valore Y.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.
- 5 Fate clic sui colori nel componente List per visualizzarli in un clip filmato.

Creare un'istanza del componente List mediante ActionScript

In questo esempio viene creato un elenco usando ActionScript e compilato usando il metodo `addItem()`.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente List dal pannello Componenti al pannello Libreria del documento corrente.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.

Usare il componente NumericStepper

Il componente NumericStepper consente a un utente di incrementare o diminuire un valore all'interno di un set di numeri ordinato. Il componente consiste di un numero in una casella di testo visualizzato accanto a piccoli pulsanti freccia su e giù. Quando un utente fa clic sui pulsanti, il numero aumenta o diminuisce in modo incrementale in base all'unità specificata nel parametro `stepSize`, fino a quando l'utente non rilascia i pulsanti o finché non viene raggiunto il valore minimo o massimo. Il testo contenuto nella casella di testo del componente NumericStepper è modificabile.

Un'anteprima dal vivo di ogni istanza NumericStepper riporta l'impostazione del parametro `value` indicato nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti. Tuttavia, l'anteprima dal vivo non consente di interagire con i pulsanti freccia di NumericStepper utilizzando il mouse o la tastiera.

Interazione dell'utente con il componente NumericStepper

Potete usare il componente NumericStepper ovunque desiderate che un utente selezioni un valore numerico. Ad esempio, potete usare un componente NumericStepper in un form per impostare il giorno, il mese e l'anno di scadenza di una carta di credito. È inoltre possibile utilizzare un componente NumericStepper per consentire a un utente di aumentare o diminuire la dimensione di un carattere.

Il componente NumericStepper è in grado di gestire solo dati numerici. Inoltre è necessario ridimensionare lo stepper in fase di creazione per visualizzare un numero con più di due cifre (ad esempio, i numeri 5246 o 1,34).

Un componente NumericStepper può essere abilitato o disabilitato in un'applicazione. Quando è disattivato, un componente NumericStepper non può ricevere input dal mouse o dalla tastiera. Un componente NumericStepper abilitato viene attivato se l'utente lo seleziona mediante clic o spostandosi su di esso mediante il tasto Tab e l'attivazione interna del componente è impostata sulla casella di testo. Quando un'istanza NumericStepper è attiva, potete utilizzare i seguenti tasti per controllarla:

Chiave	Descrizione
Freccia giù	Il valore cambia di un'unità.
Freccia sinistra	Sposta il punto di inserimento a sinistra all'interno della casella di testo.
Freccia destra	Sposta il punto di inserimento a destra all'interno della casella di testo.
Maiusc+Tab	Attiva l'oggetto precedente.
Tab	Attiva l'oggetto successivo.
Freccia su	Il valore cambia di un'unità.

Per ulteriori informazioni sul controllo dell'attivazione, vedete la classe FocusManager nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#) e “Operazioni con FocusManager” a pagina 29.

Parametri del componente NumericStepper

I seguenti parametri possono essere impostati per ogni istanza del componente NumericStepper nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `maximum`, `minimum`, `stepSize` e `value`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe NumericStepper nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente NumericStepper

Nella procedura seguente viene descritto come aggiungere un componente NumericStepper a un'applicazione durante la creazione. Nell'esempio vengono collocati sullo stage un componente NumericStepper e un componente Label e viene creato un listener per un evento `Event.CHANGE` sull'istanza NumericStepper. Quando il valore dello stepper numerico cambia, il nuovo valore viene visualizzato nella proprietà `text` dell'istanza Label.

- 1 Trascinare un componente NumericStepper dal pannello Componenti allo stage.
- 2 Nella finestra di ispezione Proprietà, inserite **aNs** come nome di istanza.
- 3 Trascinare un componente Label dal pannello Componenti allo stage.
- 4 Nella finestra di ispezione Proprietà, inserite **aLabel** come nome di istanza.
- 5 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
import flash.events.Event;

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

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

In questo esempio, la proprietà `text` dell'etichetta viene impostata sul valore di NumericStepper. La funzione `changeHandler()` aggiorna la proprietà `text` dell'etichetta ogni volta che cambia il valore nell'istanza NumericStepper.

- 6 Selezionare Controllo > Prova filmato.

Creare un componente NumericStepper mediante ActionScript

Nell'esempio seguente vengono creati tre componenti NumericStepper mediante il codice ActionScript, per inserire rispettivamente il giorno, il mese e l'anno della data di nascita dell'utente. Per ogni componente NumericStepper, vengono inoltre aggiunti componenti Label per un messaggio di richiesta e per gli identificatori.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente Label nel pannello Libreria.
- 3 Trascinate un componente NumericStepper nel pannello Libreria.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.

Usare il componente ProgressBar

Il componente ProgressBar visualizza l'avanzamento del caricamento di un contenuto, che rassicura l'utente quando il contenuto è di grandi dimensioni e può ritardare l'esecuzione dell'applicazione. Questo componente risulta utile per visualizzare l'avanzamento del caricamento di immagini e parti di un'applicazione. Il processo di caricamento può essere di tipo determinato o indeterminato. Una barra di avanzamento *determinata* è una rappresentazione lineare dell'avanzamento di un'attività nel tempo ed è usata quando è nota la quantità di contenuto da caricare. Una barra di avanzamento *indeterminata* viene usata quando la quantità di contenuto da caricare non è nota. Potete aggiungere un componente Label per visualizzare l'avanzamento del caricamento sotto forma di percentuale.

Il componente ProgressBar usa la modifica in scala a 9 porzioni e dispone di uno skin per la barra, uno per la traccia e uno skin indeterminato.

Interazione dell'utente con il componente ProgressBar

Le modalità possibili in cui usare il componente ProgressBar sono tre. Le modalità più comunemente usate sono event e polled, che specificano un processo di caricamento che genera gli eventi `progress` e `complete` (modalità event e polled) oppure espone le proprietà `bytesLoaded` e `bytesTotal` (modalità polled). Inoltre, potete usare il componente ProgressBar in modalità manuale impostando le proprietà `maximum`, `minimum` e `value` congiuntamente a chiamate del metodo `ProgressBar.setProgress()`. Potete impostare la proprietà `indeterminate` per indicare se il componente ProgressBar ha un riempimento a strisce e un'origine di caricamento di dimensione sconosciuta (`true`) oppure un riempimento uniforme e un'origine di caricamento di dimensione nota (`false`).

Per specificare la modalità del componente ProgressBar, impostate la relativa proprietà `mode` mediante il parametro `mode` nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti, oppure mediante ActionScript.

Se usate il componente ProgressBar per visualizzare lo stato di elaborazione, ad esempio per analizzare 100.000 voci, e se il componente si trova in un ciclo di un solo fotogramma, gli aggiornamenti del componente non saranno visibili in quanto lo schermo non viene ridisegnato.

Parametri del componente ProgressBar

I seguenti parametri possono essere impostati per ogni istanza del componente ProgressBar nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `direction`, `mode` e `source`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome.

È possibile creare il codice ActionScript per controllare queste e altre opzioni del componente ProgressBar utilizzando le proprietà, i metodi e gli eventi. Per ulteriori informazioni, vedete la classe ProgressBar nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente ProgressBar

Nella procedura seguente viene illustrato come aggiungere un componente ProgressBar a un'applicazione durante la creazione. In questo esempio, il componente ProgressBar viene usato in modalità event. In modalità event, il contenuto in fase di caricamento genera eventi `progress` e `complete` che il componente invia per indicare l'avanzamento. Nell'esempio, quando si verifica l'evento `progress` viene aggiornata un'etichetta per visualizzare la percentuale di contenuto caricata. Nell'esempio, quando si verifica l'evento `complete` vengono visualizzati un messaggio di caricamento completato e il valore della proprietà `bytesTotal` che corrisponde alle dimensioni del file.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente ProgressBar dal pannello Componenti allo stage.
 - Nella finestra di ispezione Proprietà, inserite **aPb** come nome di istanza.
 - Nella sezione Parametri, inserite **200** per il valore X.
 - Inserite **260** per il valore Y.
 - Selezionate `event` per il parametro `mode`.
- 3 Trascinare il componente Button dal pannello Componenti allo stage.
 - Nella finestra di ispezione Proprietà, immettete **loadButton** come nome di istanza.
 - Inserite **220** per il parametro X.
 - Inserite **290** per il parametro Y.
 - Inserite **Load Sound** per il parametro `label`.
- 4 Trascinate il componente Label nello stage e assegnate il nome di istanza **progLabel**.
 - Inserite **150** per il valore W.
 - Inserite **200** per il parametro X.
 - Inserite **230** per il parametro Y.
 - Nella sezione Parametri, cancellate il valore del parametro `text`.
- 5 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi inserite il codice ActionScript seguente, che carica un file 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 Selezionare Controllo > Prova filmato.

Creare un'applicazione con il componente ProgressBar in modalità polled

Nell'esempio seguente il componente ProgressBar viene impostato in modalità polled. In modalità polled, l'avanzamento è determinato mediante l'ascolto degli eventi `progress` nel contenuto caricato e usando le relative proprietà `bytesLoaded` e `bytesTotal` per calcolare l'avanzamento. In questo esempio viene caricato un oggetto `Sound`, vengono ascoltati gli eventi `progress` e viene calcolata la percentuale caricata usando le proprietà `bytesLoaded` e `bytesTotal`. La percentuale caricata viene visualizzata sia in un'etichetta che nel pannello Output.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente ProgressBar dal pannello Componenti allo stage e assegnategli i seguenti valori nella finestra di ispezione Proprietà:
 - Inserite **aPb** come nome dell'istanza.
 - Inserite **185** per il valore X.
 - Inserite **225** per il valore Y.
- 3 Trascinate un componente Label nello stage e immettete i seguenti valori nella finestra di ispezione Proprietà:
 - Inserite **progLabel** come nome dell'istanza.
 - Inserite **180** per il valore X.

- Inserite **180** per il valore Y.
 - Nella sezione Parametri, cancellate il valore del parametro text.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale e immettete il codice ActionScript seguente, che crea un oggetto Sound (aSound) e chiama il metodo loadSound() per caricare un suono nell'oggetto 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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.

Creare un'applicazione con il componente ProgressBar in modalità manual

Nell'esempio seguente il componente ProgressBar viene impostato in modalità manual. In modalità manual, dovete impostare l'avanzamento manualmente chiamando il metodo setProgress() e specificando i valori corrente e massimo per determinare il livello di avanzamento. In modalità manual, la proprietà source non viene impostata. In questo esempio, per incrementare il componente ProgressBar, viene usato un componente NumericStepper con un valore massimo di 250. Quando il valore nel componente NumericStepper cambia e attiva un evento CHANGE, il gestore di eventi (nsChangeHandler) chiama il metodo setProgress() per l'avanzamento di ProgressBar. Viene inoltre visualizzata la percentuale di avanzamento completata, in base al valore massimo.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente ProgressBar dal pannello Componenti allo stage e assegnategli i seguenti valori nella finestra di ispezione Proprietà:
 - Inserite **aPb** come nome dell'istanza.
 - Inserite **180** per il valore X.
 - Inserite **175** per il valore Y.
- 3 Trascinate un componente NumericStepper nello stage e immettete i seguenti valori nella finestra di ispezione Proprietà:
 - Inserite **aNs** come nome dell'istanza.
 - Inserite **220** per il valore X.
 - Inserite **215** per il valore Y.

- Nella sezione Parametri, inserite **250** per il parametro maximum, **0** per il valore minimum, **1** per il parametro stepSize e **0** per il parametro value.
- 4 Trascinate un componente Label nello stage e immettete i seguenti valori nella finestra di ispezione Proprietà:
 - Inserite **progLabel** come nome dell'istanza.
 - Inserite **150** per il valore W.
 - Inserite **180** per il valore X.
 - Inserite **120** per il valore Y.
 - Nella sezione Parametri, cancellate il valore Label del parametro text.
 - 5 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice seguente:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.
- 7 Fate clic sulla freccia su del componente NumericStepper per determinare l'avanzamento di ProgressBar.

Creare un componente ProgressBar mediante ActionScript

In questo esempio viene creato un componente ProgressBar mediante ActionScript. Inoltre, viene duplicata la funzionalità dell'esempio precedente, in cui viene creato un componente ProgressBar in modalità manual.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente ProgressBar nel pannello Libreria.
- 3 Trascinate il componente NumericStepper nel pannello Libreria.
- 4 Trascinate il componente Label nel pannello Libreria.
- 5 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice seguente:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.
- 7 Fate clic sulla freccia su del componente NumericStepper per determinare l'avanzamento di ProgressBar.

Usare il componente RadioButton

Il componente RadioButton consente all'utente di selezionare una singola opzione tra una serie di possibilità. Questo componente deve essere usato in un gruppo di almeno due istanze RadioButton. Solo un membro del gruppo può essere selezionato in un determinato momento. La selezione di un pulsante di scelta in un gruppo deselecta il pulsante di scelta attualmente selezionato nel gruppo stesso. Potete impostare il parametro `groupName` per indicare a quale gruppo appartiene un pulsante di scelta.

Un pulsante di scelta è una parte essenziale di molte applicazioni basate su form presenti sul Web. È possibile utilizzare pulsanti di scelta ovunque si desideri che l'utente effettui una scelta da un gruppo di opzioni. Ad esempio, è possibile utilizzare pulsanti di scelta in un form per chiedere quale tipo di carta di credito il cliente intende utilizzare.

Interazione dell'utente con il componente RadioButton

Un pulsante di scelta può essere abilitato o disabilitato. Se è disabilitato, un pulsante di scelta non può ricevere input dal mouse o dalla tastiera. Quando l'utente fa clic o preme il tasto Tab per spostarsi in un gruppo di componenti RadioButton, solo il pulsante di scelta selezionato viene reso attivo. Una volta attivato, potete usare i tasti seguenti per controllarlo:

Chiave	Descrizione
Freccia su/freccia sinistra	La selezione si sposta sul pulsante di scelta precedente all'interno del gruppo di pulsanti di scelta.
Freccia giù/freccia destra	La selezione si sposta sul pulsante di scelta successivo all'interno del gruppo di pulsanti di scelta.
Tab	Sposta la selezione dal gruppo di pulsanti di scelta al componente successivo.

Per ulteriori informazioni sul controllo dell'attivazione, vedete l'interfaccia IFocusManager e la classe FocusManager nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#) e [“Operazioni con FocusManager”](#) a pagina 29.

Sullo stage, un'anteprima dal vivo di ogni istanza del componente RadioButton consente di visualizzare le modifiche apportate ai parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti durante la creazione. Tuttavia, la reciproca esclusione di selezione non viene visualizzata nell'anteprima dal vivo. Se impostate il parametro `selected` su `true` per due pulsanti di scelta dello stesso gruppo, entrambi sono visualizzati come selezionati anche se solo l'ultima istanza creata appare selezionata in fase di runtime. Per ulteriori informazioni, vedete [“Parametri del componente RadioButton”](#) a pagina 80.

Quando aggiungete il componente RadioButton a un'applicazione, potete renderlo accessibile a uno screen reader aggiungendo le seguenti righe di codice ActionScript:

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

Attivare l'accessibilità per il componente una sola volta, indipendentemente dal numero di istanze del componente. Per ulteriori informazioni, vedete il Capitolo 18, [“Creazione di contenuto accessibile”](#), nella guida [Uso di Flash](#).

Parametri del componente RadioButton

I seguenti parametri di creazione possono essere impostati per ogni istanza del componente RadioButton nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `groupName`, `label`, `LabelPlacement`, `selected` e `value`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe RadioButton nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

È possibile creare codice ActionScript per impostare opzioni aggiuntive per le istanze RadioButton utilizzando i metodi, le proprietà e gli eventi della classe RadioButton.

Creare un'applicazione con il componente RadioButton

La procedura seguente indica come aggiungere componenti RadioButton a un'applicazione durante la creazione. In questo esempio, i componenti RadioButton vengono usati per presentare una domanda affermativa/negativa. I dati dal componente RadioButton vengono visualizzati in un componente TextArea.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinare due componenti RadioButton dal pannello Componenti allo stage.
- 3 Selezionate il primo pulsante di scelta. Nella finestra di ispezione Proprietà, assegnate al pulsante il nome di istanza `yesRb` e il nome di gruppo `rbGroup`.

- 4 Selezionate il secondo pulsante di scelta. Nella finestra di ispezione Proprietà, assegnate al pulsante il nome di istanza **noRb** e il nome di gruppo **rbGroup**.
- 5 Trascinate un componente TextArea dal pannello Componenti allo stage e assegnategli il nome di istanza **aTa**.
- 6 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.

Creare un componente RadioButton mediante ActionScript

Nell'esempio seguente viene usato ActionScript per creare tre componenti RadioButton per i colori rosso, blu e verde e per disegnare una casella grigia. Nella proprietà `value` di ogni componente RadioButton è specificato il valore esadecimale per il colore associato al pulsante. Quando un utente fa clic su uno dei componenti RadioButton, la funzione `clickHandler()` chiama il metodo `drawBox()`, passando il colore dalla proprietà `value` di RadioButton da applicare alla casella.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente RadioButton nel pannello Libreria.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.

Per ulteriori informazioni, vedete la classe `RadioButton` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Usare il componente ScrollPane

Potete utilizzare il componente `ScrollPane` per visualizzare qualsiasi contenuto di dimensioni maggiori rispetto all'area in cui viene caricato. Ad esempio, se disponete di un'immagine di grandi dimensioni per la quale è disponibile solo poco spazio in un'applicazione, potete caricarla in un componente `ScrollPane`. Questo componente supporta i clip filmato e i file JPEG, PNG, GIF e SWF.

I componenti come `ScrollPane` e `UILoader` dispongono di eventi `complete` che consentono di determinare l'avvenuto caricamento del contenuto. Se desiderate impostare delle proprietà per il contenuto di un componente `ScrollPane` o `UILoader`, impostate l'ascolto dell'evento `complete` e la proprietà nel gestore di eventi. Ad esempio, il codice seguente crea un listener per l'evento `Event.COMPLETE` e un gestore di eventi che imposta la proprietà `alpha` del contenuto di `ScrollPane` su 0,5:

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

Se specificate una posizione quando caricate del contenuto nel componente `ScrollPane`, dovete specificare il valore 0, 0 (coordinate X e Y). Ad esempio, il codice seguente carica correttamente il componente `ScrollPane` perché la casella è disegnata in corrispondenza delle coordinate 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
```

Per ulteriori informazioni, vedete la classe `ScrollPane` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Interazione dell'utente con il componente ScrollPane

Un componente `ScrollPane` può essere abilitato o disabilitato. Se è disabilitato, non può ricevere input dal mouse o dalla tastiera. Per controllare un componente `ScrollPane` attivo, un utente può utilizzare i tasti seguenti:

Chiave	Descrizione
Freccia giù	Il contenuto si sposta verticalmente di una riga verso l'alto.
Freccia su	Il contenuto si sposta verticalmente di una riga verso il basso.
Fine	Il contenuto si sposta nella parte inferiore del componente <code>ScrollPane</code> .
Freccia sinistra	Il contenuto si sposta orizzontalmente di una riga verso destra.
Freccia destra	Il contenuto si sposta orizzontalmente di una riga verso sinistra.
Home	Il contenuto si sposta all'inizio del componente <code>ScrollPane</code> .
Fine	Il contenuto si sposta nella parte inferiore del componente <code>ScrollPane</code> .
Pag giù	Il contenuto si sposta verticalmente di una pagina verso l'alto.
Pag su	Il contenuto si sposta verticalmente di una pagina verso il basso.

Per interagire con il contenuto e con le barre di scorrimento verticale e orizzontale del componente `ScrollPane`, l'utente può usare il mouse. Potete trascinare il contenuto con il mouse quando la proprietà `scrollDrag` è impostata su `true`. La visualizzazione di un puntatore a forma di mano sopra il contenuto indica che questo può essere trascinato. A differenza di molti altri controlli, le azioni si verificano quando viene premuto il pulsante del mouse e continuano finché non viene rilasciato. Se il contenuto presenta tabulazioni valide, dovete impostare `scrollDrag` su `false`, altrimenti ogni interazione del mouse con il contenuto richiamerà un'azione di `scrollDrag`.

Parametri del componente ScrollPane

Potete impostare i parametri seguenti per ogni istanza del componente ScrollPane nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `horizontalLineScrollSize`, `horizontalPageScrollSize`, `horizontalScrollPolicy`, `scrollDrag`, `source`, `verticalLineScrollSize`, `verticalPageScrollSize` e `verticalScrollPolicy`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe ScrollPane nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

È possibile creare codice ActionScript per controllare queste e altre opzioni per un componente ScrollPane utilizzando le proprietà, i metodi e gli eventi.

Creare un'applicazione con il componente ScrollPane

La procedura seguente indica come aggiungere un componente ScrollPane a un'applicazione durante la creazione. In questo esempio, il componente ScrollPane carica un'immagine da un percorso specificato dalla proprietà `source`.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente ScrollPane dal pannello Componenti allo stage e assegnategli il nome di istanza **aSp**.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.

Creare un'istanza ScrollPane mediante ActionScript

Nell'esempio viene creato un componente ScrollPane, ne vengono impostate le dimensioni e viene caricata un'immagine nel componente usando la proprietà `source`. Vengono inoltre creati due listener: il primo rileva un evento `scroll` e visualizza la posizione dell'immagine quando questa viene fatta scorrere in verticale o in orizzontale; il secondo rileva l'evento `complete` e visualizza un messaggio nel pannello Output in cui è indicato che il caricamento dell'immagine è completato.

In questo esempio viene creato un componente ScrollPane mediante ActionScript e viene inserito un clip filmato (una casella rossa), largo 150 pixel e alto 300 pixel.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente ScrollPane dal pannello Componenti al pannello Libreria del documento corrente.

- 3 Trascinate il componente DataGrid dal pannello Componenti al pannello Libreria del documento corrente.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.

Usare il componente Slider

Il componente Slider consente all'utente di selezionare un valore mediante lo spostamento di un *cursore* grafico tra le estremità di una traccia che corrisponde a un intervallo di valori. Potete usare un componente Slider per consentire, ad esempio, all'utente di scegliere un valore quale un numero o una percentuale. Inoltre, potete usare ActionScript per fare in modo che il valore del componente influisca sul comportamento di un secondo oggetto, associando ad esempio il componente a un'immagine e riducendolo o ingrandendolo in base alla posizione relativa, o valore, della casella di scorrimento.

Il valore corrente del componente Slider è determinato dalla posizione relativa della casella di scorrimento rispetto alle due estremità della traccia, ovvero i valori minimo e massimo.

Il componente Slider consente un intervallo continuo di valori tra i relativi valori minimo e massimo, tuttavia potete impostare il parametro `snapInterval` per specificare degli intervalli tra il valore minimo e massimo. Lungo la traccia di scorrimento di un componente, a intervalli specificati, possono essere presenti delle tacche, che sono indipendenti dai valori assegnati del componente.

Per impostazione predefinita il componente Slider ha un orientamento orizzontale, tuttavia potete impostare un orientamento verticale impostando il valore del parametro `direction` su `vertical` . La traccia di scorrimento si estende da un'estremità all'altra e le tacche sono posizionate da sinistra a destra sopra la traccia.

Interazione dell'utente con il componente Slider

Quando un'istanza Slider è attiva, potete utilizzare i seguenti tasti per controllarla:

Chiave	Descrizione
Freccia destra	Aumenta il valore associato per un componente Slider orizzontale.
Freccia su	Aumenta il valore associato per un componente Slider verticale.
Freccia sinistra	Diminuisce il valore associato per un componente Slider orizzontale.
Freccia giù	Diminuisce il valore associato per un componente Slider verticale.
Maiusc+Tab	Attiva l'oggetto precedente.
Tab	Attiva l'oggetto successivo.

Per ulteriori informazioni sul controllo dell'attivazione, vedete l'interfaccia IFocusManager e la classe FocusManager nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#) e “Operazioni con FocusManager” a pagina 29.

Nell'anteprima dal vivo di ogni istanza del componente Slider vengono visualizzate le modifiche apportate durante la creazione ai parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti.

Parametri del componente Slider

I seguenti parametri di creazione possono essere impostati per ogni istanza del componente Slider nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `direction`, `liveDragging`, `maximum`, `minimum`, `snapInterval`, `tickInterval` e `value`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe Slider nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente Slider

Nell'esempio seguente viene creata un'istanza Slider per consentire all'utente di esprimere il proprio livello di soddisfazione per un evento ipotetico. L'utente sposta il componente Slider verso destra o verso sinistra per indicare un livello di soddisfazione maggiore o minore.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente Label dal pannello Componenti al centro dello stage.
 - Assegnate all'istanza il nome **valueLabel**.
 - Assegnate il valore **0percent** al parametro `text`.
- 3 Trascinate un componente Slider dal pannello Componenti e posizionate al centro sotto `value_lbl`.
 - Assegnate all'istanza il nome **aSlider**.
 - Assegnate il valore **200** per la larghezza (W:).
 - Assegnate il valore **10** per l'altezza (H:).
 - Assegnate il valore **100** al parametro `maximum`.
 - Assegnate il valore **10** ai parametri `snapInterval` e `tickInterval`.
- 4 Trascinate un'altra istanza del componente Label dal pannello Libreria e posizionate al centro sotto `aSlider`.
 - Assegnate all'istanza il nome **promptLabel**.
 - Assegnate il valore 250 per la larghezza (W:).
 - Assegnate il valore 22 per l'altezza (H:).
 - Inserite **Indicare il proprio livello di soddisfazione** per il parametro `text`.

- 5 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionare Controllo > Prova filmato.

In questo esempio, quando spostate la casella di scorrimento da un intervallo all'altro, un listener dell'evento `SliderEvent.CHANGE` aggiorna la proprietà `text` di `valueLabel` per visualizzare la percentuale corrispondente alla posizione della casella di scorrimento.

Creare un'applicazione con il componente Slider mediante ActionScript

Nell'esempio seguente viene creato un componente mediante ActionScript. Viene scaricata l'immagine di un fiore e il componente Slider viene usato per consentire all'utente di applicare un effetto di dissolvenza o di rendere più brillante l'immagine mediante la modifica della proprietà `alpha` che corrisponde al valore del componente.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate i componenti Label e Slider dal pannello Componenti al pannello Libreria del documento corrente.
In questo modo, i componenti vengono aggiunti alla libreria ma non sono visibili nell'applicazione.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi inserite il codice seguente per creare e posizionare le istanze dei componenti:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.
- 5 Spostate la casella di scorrimento del componente a sinistra per sbiadire l'immagine e a destra per renderla più luminosa.

Usare il componente TextArea

Il componente TextArea è un componente di testo a riga singola, wrapper dell'oggetto TextField nativo di ActionScript. Potete usare il componente TextArea per visualizzare del testo e per modificare e ricevere input di testo se la proprietà `editable` è impostata su `true`. Il componente è in grado di visualizzare o ricevere più righe di testo e mandare a capo automaticamente le righe di testo lunghe se la proprietà `wordWrap` è impostata su `true`. La proprietà `restrict` consente di limitare i caratteri che l'utente può inserire, mentre `maxChars` consente di specificare il numero massimo di caratteri consentiti. Se il testo supera i limiti orizzontale e verticale dell'area di testo, vengono visualizzate automaticamente le barre di scorrimento orizzontale e verticale, a meno che le proprietà associate `horizontalScrollPolicy` e `verticalScrollPolicy` non siano impostate su `off`.

È possibile utilizzare un componente `TextArea` ogni volta che è necessario un campo di testo multiriga. Ad esempio, è possibile utilizzare un componente `TextArea` come campo di commento in un form. Potete impostare un listener che verifica se il campo è vuoto quando un utente preme il tasto `Tab` per uscire dal campo. Questo listener potrebbe visualizzare un messaggio di errore che indica che è necessario inserire un commento nel campo.

Se è necessario un campo di testo a riga singola, utilizzare il componente `TextInput`, componente.

Per cambiare lo stile del testo visualizzato in un'istanza `TextArea`, impostate lo stile `textFormat` usando il metodo `setStyle()`. A un componente `TextArea` potete inoltre applicare il formato HTML usando la proprietà `htmlText` in ActionScript. Inoltre, potete impostare la proprietà `displayAsPassword` su `true` per mascherare il testo mediante asterischi. Se impostate la proprietà `condenseWhite` su `true`, viene rimosso automaticamente lo spazio vuoto supplementare contenuto nel nuovo testo (spazi, interruzioni di riga e così via). Non ha effetto sul testo già presente nel controllo.

Interazione dell'utente con il componente `TextArea`

Un componente `TextArea` può essere abilitato o disabilitato in un'applicazione: Quando è disattivato, non può ricevere input dal mouse o dalla tastiera. Quando è abilitato, segue le stesse regole di attivazione e di navigazione di un oggetto `TextField` di ActionScript. Quando un'istanza del componente `TextArea` è attiva, potete usare i seguenti tasti per controllarla:

Chiave	Descrizione
Tasti freccia	Spostano il punto di inserimento in alto, in basso, a sinistra o a destra all'interno del testo, se il testo è modificabile.
PgGiù	Sposta il punto di inserimento alla fine del testo, se il testo è modificabile.
Pagina su	Sposta il punto di inserimento all'inizio del testo, se il testo è modificabile.
Maiusc+Tab	Attiva l'oggetto precedente nel ciclo di tabulazioni.
Tab	Attiva l'oggetto successivo nel ciclo di tabulazioni.

Per ulteriori informazioni sul controllo dell'attivazione, vedete la classe `FocusManager` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#) e “Operazioni con `FocusManager`” a pagina 29.

Parametri del componente `TextArea`

Potete impostare i parametri di creazione seguenti per ogni istanza del componente `TextArea` nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `condenseWhite`, `editable`, `horizontalScrollPolicy`, `maxChars`, `restrict`, `text`, `verticalScrollPolicy` e `wordwrap`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe `TextArea` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Nell'anteprima dal vivo di ogni istanza di `TextArea` vengono visualizzate le modifiche apportate ai parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti durante la creazione. Se è necessaria una barra di scorrimento, nell'anteprima dal vivo viene visualizzata ma non funziona. Il testo non è selezionabile nell'anteprima dal vivo e non è possibile immetterlo nell'istanza del componente sullo stage.

È possibile creare codice ActionScript per controllare queste e altre opzioni per il componente `TextArea` utilizzandone le proprietà, i metodi e gli eventi. Per ulteriori informazioni, vedete la classe `TextArea` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente TextArea

La procedura seguente indica come aggiungere un componente TextArea a un'applicazione durante la creazione. L'esempio imposta sull'istanza TextArea un gestore di eventi `focusOut` che verifica che l'utente abbia digitato dei caratteri nel campo di testo prima di attivare una parte diversa dell'interfaccia.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente TextArea dal pannello Componenti allo stage e assegnategli il nome di istanza **aTa**. Lasciate invariate le impostazioni predefinite dei parametri.
- 3 Trascinate un secondo componente TextArea dal pannello Componenti allo stage, posizionate lo sotto il primo e assegnategli il nome di istanza **bTa**. Lasciate invariate le impostazioni predefinite dei parametri.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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();
}
```

In questo esempio i caratteri che potete inserire nell'area di testo `aTa` vengono limitati a caratteri minuscoli, virgola, apostrofo e spazi. Vengono inoltre impostati i gestori di eventi per gli eventi `change`, `KEY_FOCUS_CHANGE` e `MOUSE_FOCUS_CHANGE` dell'area di testo `aTa`. La funzione `changeHandler()` causa la visualizzazione automatica del testo inserito nell'area di testo `aTa` nell'area di testo `bTa`, mediante l'assegnazione di `aTa.text` a `bTa.text` per ogni evento `change`. La funzione `k_m_fHandler()` per gli eventi `KEY_FOCUS_CHANGE` e `MOUSE_FOCUS_CHANGE` impedisce all'utente di premere il tasto Tab per passare al campo successivo senza avere inserito del testo. Questa impostazione impedisce l'esecuzione del comportamento predefinito.

- 5 Selezionare Controllo > Prova filmato.

Se premete il tasto Tab per attivare la seconda area di testo senza inserire del testo, viene visualizzato un messaggio di errore e viene riattivata la prima area di testo. Se inserite del testo nella prima area di testo, questo viene duplicato nella seconda area di testo.

Creare un'istanza TextArea mediante ActionScript

Nell'esempio seguente viene creato un componente TextArea mediante ActionScript. La proprietà `condenseWhite` viene impostata su `true` per comprimere lo spazio vuoto e il testo viene assegnato alla proprietà `htmlText` per avvalersi degli attributi di formattazione del testo HTML.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente TextArea nel pannello Libreria.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
import fl.controls.TextArea;

var aTa:TextArea = new TextArea();

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

In questo esempio viene usata la proprietà `htmlText` per applicare gli attributi di sottolineatura e grassetto HTML a un blocco di testo e visualizzarlo nell'area di testo `a_ta`. Viene inoltre impostata la proprietà `condenseWhite` su `true` per comprimere gli spazi all'interno del blocco di testo. Il metodo `setSize()` consente di impostare l'altezza e la larghezza dell'area di testo, mentre il metodo `move()` ne imposta la posizione. Il metodo `addChild()` aggiunge l'istanza `TextArea` allo stage.

4 Selezionare Controllo > Prova filmato.

Usare il componente TextInput

`TextInput` è un componente di testo a riga singola, wrapper dell'oggetto `TextField` nativo di `ActionScript`. Se è necessario un campo di testo multiriga, utilizzate il componente `TextArea`. Ad esempio, è possibile utilizzare un componente `TextInput` come campo password in un form. Inoltre, potete impostare un listener che verifica se nel campo è presente un numero sufficiente di caratteri quando un utente preme il tasto `Tab` per uscire dal campo. Tale listener potrebbe visualizzare un messaggio di errore che indica che deve essere immesso il numero di caratteri appropriato.

Per cambiare lo stile del testo visualizzato in un'istanza `TextInput`, potete impostare la proprietà `textFormat` usando il metodo `setStyle()`. Un componente `TextInput` può anche essere formattato in HTML o come un campo password che nasconde il testo.

Interazione dell'utente con il componente TextInput

In un'applicazione, un componente `TextInput` può essere abilitato o disabilitato. se disabilitato, non può ricevere input dal mouse o dalla tastiera. Quando è abilitato, segue le stesse regole di attivazione e di navigazione di un oggetto `TextField` di `ActionScript`. Quando un'istanza del componente `TextInput` è attiva, è anche possibile utilizzare i tasti seguenti per controllarla:

Chiave	Descrizione
Tasti freccia	Spostano il punto di inserimento di un carattere a sinistra o a destra.
Maiusc+Tab	Attiva l'oggetto precedente.
Tab	Attiva l'oggetto successivo.

Per ulteriori informazioni sul controllo dell'attivazione, vedete l'interfaccia `FocusManager` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#) e “Operazioni con `FocusManager`” a pagina 29.

Nell'anteprima dal vivo di ogni istanza del componente TextInput vengono visualizzate le modifiche apportate ai parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti durante la creazione. Il testo non è selezionabile nell'anteprima dal vivo e non è possibile immetterlo nell'istanza del componente sullo stage.

Quando si aggiunge un componente TextInput a un'applicazione, è possibile utilizzare il pannello Accessibilità per renderlo accessibile agli screen reader.

Parametri del componente TextInput

Potete impostare i seguenti parametri di creazione per ogni istanza del componente TextInput finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `editable`, `displayAsPassword`, `maxChars`, `restrict` e `text`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe TextInput nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

È possibile creare codice ActionScript per controllare queste e altre opzioni per il componente TextInput utilizzandone le proprietà, i metodi e gli eventi. Per ulteriori informazioni, vedete la classe TextInput nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente TextInput

Nella procedura seguente viene illustrato come aggiungere un componente TextInput a un'applicazione. Nell'esempio vengono usati due campi TextInput per ricevere e confermare la password. Viene usato un listener di eventi per verificare che sia stato inserito un numero minimo di otto caratteri e che il testo dei due campi corrisponda.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente Label dal pannello Componenti allo stage e assegnategli i seguenti valori nella finestra di ispezione Proprietà:
 - Inserite il nome di istanza **pwdLabel**.
 - Inserite il valore **100** per W.
 - Inserite il valore **50** per X.
 - Inserite il valore **150** per Y.
 - Nella sezione Parametri, immettete il valore **Password:** per il parametro text.
- 3 Trascinate un secondo componente Label dal pannello Componenti allo stage e assegnategli i seguenti valori:
 - Inserite il nome di istanza **confirmLabel**.
 - Inserite il valore **100** per W.
 - Inserite il valore **50** per X.
 - Inserite il valore **200** per Y.
 - Nella sezione Parametri, immettete il valore **ConfirmPassword:** per il parametro text.
- 4 Trascinate un componente TextInput dal pannello Componenti allo stage e assegnategli i seguenti valori:
 - Inserite il nome di istanza **pwdTi**.
 - Inserite il valore **150** per W.
 - Inserite il valore **190** per X.
 - Inserite il valore **150** per Y.

- Nella sezione Parametri, fate doppio clic sul valore del parametro `displayAsPassword` e selezionate **true**. Il valore inserito nel campo di testo verrà mascherato con degli asterischi.
- 5 Trascinate un secondo componente Label dal pannello Componenti allo stage e assegnategli i seguenti valori:
- Inserite il nome di istanza **confirmTi**.
 - Inserite il valore **150** per W.
 - Inserite il valore **190** per X.
 - Inserite il valore **200** per Y.
 - Nella sezione Parametri, fate doppio clic sul valore del parametro `displayAsPassword` e selezionate **true**. Il valore inserito nel campo di testo verrà mascherato con degli asterischi.
- 6 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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);
```

Questo codice imposta un gestore di eventi `enter` sull'istanza `TextInput` chiamata `confirmTi`. Se le due password non corrispondono o l'utente digita meno di otto caratteri, nell'esempio viene visualizzato il messaggio: "Password is incorrect. Please reenter it." Se le password sono costituite da un minimo di otto caratteri e corrispondono, nell'esempio viene visualizzato il valore inserito nel pannello Output.

- 7 Selezionare Controllo > Prova filmato.

Creare un'istanza `TextInput` mediante `ActionScript`

Nell'esempio seguente viene creato un componente `TextInput` mediante `ActionScript`. Viene inoltre creato un componente `Label` usato per richiedere all'utente di inserire il nome. Viene impostata la proprietà `restrict` del componente per consentire solo lettere maiuscole e minuscole, un punto e uno spazio. Viene creato un oggetto `TextFormat` usato per formattare il testo nei componenti `Label` e `TextInput`.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente `TextInput` dal pannello Componenti allo stage.
- 3 Trascinate un componente `Label` dal pannello Componenti allo stage.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.

Usare il componente TileList

Il componente `TileList` è un elenco costituito da righe e colonne in cui vengono inseriti dati mediante un fornitore di dati. Una *voce* è un'unità di dati memorizzata in una cella del componente `TileList`. Una voce viene generata nel fornitore di dati e dispone di una proprietà `label` e di una proprietà `source`. La proprietà `label` identifica il contenuto da visualizzare in una cella, mentre la proprietà `source` fornisce un valore.

Potete creare un'istanza `Array` o recuperarne una da un `server`. Al componente `TileList` sono associati metodi che vengono inviati tramite `proxy` al relativo fornitore di dati, ad esempio, i metodi `addItem()` e `removeItem()`. Se all'elenco non vengono associati dei fornitori di dati esterni, questi metodi creano automaticamente un'istanza di fornitore di dati che viene elaborata con `List.dataProvider`.

Interazione dell'utente con il componente TileList

Un componente `TileList` esegue il rendering di ogni riga mediante uno `Sprite` che implementa l'interfaccia `ICellRenderer`. Potete specificare il `renderer` mediante la proprietà `cellRenderer` di `TileList`. L'oggetto `CellRenderer` predefinito del componente `TileList` è `ImageCell`, che visualizza un'immagine (classe, `bitmap`, istanza o `URL`) e un'etichetta facoltativa. L'etichetta è costituita da una sola riga di testo che viene sempre allineata nella parte inferiore della cella. Potete scorrere un componente `TileList` in una sola direzione.

Quando un'istanza del componente `TileList` è attiva, potete anche usare i tasti seguenti per accedere alle voci in esso contenute:

Chiave	Descrizione
Freccia su e Freccia giù	Consentono di spostarsi verso l'alto o verso il basso in una colonna. Se la proprietà <code>allowMultipleSelection</code> è impostata su <code>true</code> , potete usare questi tasti in combinazione con il tasto Maiusc per selezionare più celle.
Freccia sinistra e Freccia destra	Consentono di spostarsi a sinistra o a destra in una riga. Se la proprietà <code>allowMultipleSelection</code> è impostata su <code>true</code> , potete usare questi tasti in combinazione con il tasto Maiusc per selezionare più celle.
Home	Seleziona la prima cella nel componente <code>TileList</code> . Se la proprietà <code>allowMultipleSelection</code> è impostata su <code>true</code> , potete selezionare tutte le celle della selezione corrente, fino alla prima cella, tenendo premuto Maiusc e premendo il tasto Home.
Fine	Seleziona l'ultima cella nel componente <code>TileList</code> . Se la proprietà <code>allowMultipleSelection</code> è impostata su <code>true</code> , potete selezionare tutte le celle della selezione corrente, fino all'ultima cella, tenendo premuto Maiusc e premendo il tasto Fine.
Ctrl	Se la proprietà <code>allowMultipleSelection</code> è impostata su <code>true</code> , potete selezionare più celle, senza un ordine specifico.

Quando aggiungete il componente `TitleList` a un'applicazione, potete renderlo accessibile a uno screen reader aggiungendo le seguenti righe di codice `ActionScript`:

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

Attivare l'accessibilità per il componente una sola volta, indipendentemente dal numero di istanze del componente. Per ulteriori informazioni, vedete il Capitolo 18, "Creazione di contenuto accessibile", nella guida *Uso di Flash*.

Parametri del componente `TileList`

I seguenti parametri di creazione possono essere impostati per ogni istanza del componente `TileList` nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `allowMultipleSelection`, `columnCount`, `columnWidth`, `dataProvider`, `direction`, `horizontalScrollLineSize`, `horizontalScrollPageSize`, `labels`, `rowCount`, `rowHeight`, `ScrollPolicy`, `verticalScrollLineSize` e `verticalScrollPageSize`. Ciascuno di questi parametri ha una proprietà `ActionScript` corrispondente con lo stesso nome. Per informazioni sull'uso del parametro `dataProvider`, vedete "Usare il parametro `dataProvider`" a pagina 31.

Potete creare codice `ActionScript` per impostare opzioni aggiuntive per le istanze `TileList` usandone le proprietà, i metodi e gli eventi. Per ulteriori informazioni, vedete la classe `TileList` nella *Guida di riferimento di ActionScript 3.0 per Flash Professional*.

Creare un'applicazione con il componente `TileList`

In questo esempio viene usato un clip filmato per compilare un componente `TileList` con un array di colori.

- 1 Create un nuovo documento Flash (`ActionScript 3.0`).
- 2 Trascinate nello stage un componente `TileList` e assegnategli il nome di istanza `aTL`.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice `ActionScript` seguente:

```
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 Selezionate Controllo > Prova filmato per provare l'applicazione.

Creare un componente TileList mediante ActionScript

Nell'esempio viene creata dinamicamente un'istanza TileList e vi vengono aggiunte istanze dei componenti ColorPicker, ComboBox, NumericStepper e CheckBox. Viene creato un array contenente le etichette e i nomi del componente per visualizzare e assegnare l'array (dp) alla proprietà dataProvider di TileList. Vengono usate le proprietà columnWidth e rowHeight e il metodo setSize() per il layout di TileList, il metodo move() per posizionare il componente sullo stage e lo stile contentPadding per inserire dello spazio tra i bordi dell'istanza TileList e il relativo contenuto e il metodo sortItemsOn() per ordinare il contenuto in base alle etichette.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate i seguenti componenti dal pannello Componenti al pannello Libreria: ColorPicker, ComboBox, NumericStepper, CheckBox e TileList.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
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 Selezionate Controllo > Prova filmato per provare l'applicazione.

Usare il componente UI Loader

Il componente UI Loader è un contenitore che consente la visualizzazione di file SWF, JPEG, JPEG progressivi, PNG e GIF. Potete usare un componente UI Loader ogni volta che dovete recuperare contenuto da una posizione remota per inserirlo in un'applicazione Flash. Ad esempio, potete usare un componente UI Loader per aggiungere un logo aziendale (file JPEG) a un form oppure inserirlo in un'applicazione per visualizzare fotografie. Usate il metodo `load()` per caricare il contenuto, la proprietà `percentLoaded` per determinare la quantità di contenuto caricata e l'evento `complete` per determinare quando il caricamento è completato.

Potete modificare in scala il contenuto del componente UI Loader o ridimensionarlo per adattarlo alla dimensione del contenuto. Per impostazione predefinita, il contenuto viene modificato in scala per adattarsi al componente UI Loader. Potete anche caricare il contenuto in fase di runtime e monitorare l'avanzamento del caricamento (dopo il primo caricamento, tuttavia, il contenuto viene inserito nella cache, per cui l'avanzamento raggiunge rapidamente il 100%). Se specificate una posizione quando caricate contenuto nel componente UI Loader, dovete specificare il valore 0, 0 (coordinate X e Y).

Interazione dell'utente con il componente UI Loader

Un componente UI Loader non può ricevere l'attivazione. Il contenuto caricato nel componente UI Loader può invece essere attivato e avere proprie interazioni di attivazione. Per ulteriori informazioni sul controllo dell'attivazione, vedete la classe `FocusManager` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#) e "Operazioni con `FocusManager`" a pagina 29.

Parametri del componente UILoader

I seguenti parametri di creazione possono essere impostati per ogni istanza del componente UILoader nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `autoLoad`, `maintainAspectRatio`, `source` e `scaleContent`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome.

Nell'anteprima dal vivo di ogni istanza del componente UILoader vengono visualizzate le modifiche apportate ai parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti durante la creazione.

Potete creare codice ActionScript per impostare opzioni aggiuntive per le istanze di UILoader usandone le proprietà, i metodi e gli eventi. Per ulteriori informazioni, vedete la classe UILoader nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente UILoader

Nella procedura seguente viene descritto come aggiungere un componente UILoader a un'applicazione durante la creazione. In questo esempio, il loader carica l'immagine GIF di un logo.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente UILoader dal pannello Componenti allo stage.
- 3 Nella finestra di ispezione Proprietà, inserite **aUI** come nome di istanza.
- 4 Selezionate il loader nello stage e della finestra di ispezione dei componenti, quindi inserite <http://www.helpexamples.com/images/logo.gif> come valore del parametro `source`.

Creare l'istanza di un componente UILoader mediante ActionScript

In questo esempio viene creato un componente UILoader mediante ActionScript e viene caricata l'immagine JPEG di un fiore. Quando si verifica l'evento `complete`, viene visualizzato il numero di byte caricati nel pannello Output.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente UILoader dal pannello Componenti al pannello Libreria del documento corrente.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice ActionScript seguente:

```
import fl.containers.UILoader;

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

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

- 4 Selezionare Controllo > Prova filmato.

Usare il componente UIScrollBar

Il componente UIScrollBar consente di aggiungere una barra di scorrimento a un campo di testo, durante la creazione oppure in fase di runtime con ActionScript. Per usare il componente UIScrollBar, create un campo di testo sullo stage e trascinate il componente UIScrollBar dal pannello Componenti in un'area qualsiasi del riquadro di delimitazione del campo di testo.

Se la lunghezza della barra di scorrimento è inferiore alle dimensioni complessive delle relative frecce di scorrimento, la barra non viene visualizzata correttamente. Uno dei pulsanti freccia viene infatti coperto dall'altro. In Flash non è disponibile una funzione di controllo per questo tipo di errore, quindi, in questo caso, è consigliabile nascondere la barra di scorrimento utilizzando ActionScript. Se le dimensioni della barra di scorrimento sono insufficienti per contenere la casella di scorrimento, questa viene nascosta automaticamente.

Il funzionamento del componente UIScrollBar è analogo a qualsiasi altra barra di scorrimento. Contiene pulsanti freccia a entrambe le estremità, separati da una traccia di scorrimento e una casella di scorrimento. Può essere associato a un bordo qualsiasi di un campo di testo e utilizzato sia in verticale che in orizzontale.

Per ulteriori informazioni, vedete la classe TextField nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Interazione dell'utente con il componente UIScrollBar

A differenza di molti altri componenti, il componente UIScrollBar non richiede l'esecuzione di clic ripetuti, ma può ricevere l'input continuo del mouse, ad esempio quando l'utente tiene premuto il pulsante del mouse.

Il componente UIScrollBar non supporta alcuna interazione mediante la tastiera.

Parametri del componente UIScrollBar

I seguenti parametri di creazione possono essere impostati per ogni istanza del componente UIScrollBar nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: `direction` e `scrollTargetName`. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome.

Potete creare codice ActionScript per impostare opzioni aggiuntive per le istanze di UIScrollBar usandone le proprietà, i metodi e gli eventi. Per ulteriori informazioni, vedete la classe UIScrollBar nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Creare un'applicazione con il componente UIScrollBar

Nella procedura seguente viene descritto come aggiungere un componente UIScrollBar a un'applicazione durante la creazione.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Create un campo di testo dinamico la cui altezza sia sufficiente per contenere una o più righe di testo e assegnategli il nome di istanza `myText` nella finestra di ispezione Proprietà.
- 3 Nella finestra di ispezione Proprietà, impostare il tipo di riga del campo di immissione di testo su Multiriga, oppure su Multiriga senza a capo auto se si prevede di utilizzare la barra di scorrimento in orizzontale.
- 4 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale e immettete il codice ActionScript seguente per compilare la proprietà `text` in modo che sia necessario scorrere il testo per visualizzarlo 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:** Assicurarsi che il campo di testo sullo stage sia sufficientemente piccolo da richiedere lo scorrimento per visualizzare tutto il testo. In caso contrario, la barra di scorrimento non è visibile o appare sotto forma di due righe prive del riquadro da trascinare per scorrere il contenuto.*

- 5 Verificate che la funzione di aggancio agli oggetti sia attivata (Visualizza > Aggancio > Aggancia agli oggetti).
- 6 Trascinare un'istanza di `UIScrollBar` dal pannello Componenti al campo di testo di input accanto al lato di aggancio. Quando si rilascia il pulsante del mouse, il componente deve essere sovrapposto al campo di testo per agganciarsi correttamente al campo. Assegnate all'istanza il nome **mySb**.

La proprietà `scrollTargetName` del componente viene compilata automaticamente con il nome di istanza del campo di testo nella finestra di ispezione Proprietà e nella finestra di ispezione dei componenti. Se non viene visualizzata nella scheda Parametri, è possibile che non sia stata sufficientemente sovrapposta l'istanza `UIScrollBar`.

- 7 Selezionare Controllo > Prova filmato.

Creare l'istanza di un componente `UIScrollBar` mediante `ActionScript`

Potete inoltre creare un'istanza del componente `UIScrollBar` mediante `ActionScript` e associarla a un campo di testo in fase di runtime. Nell'esempio seguente viene creata un'istanza `UIScrollBar` con orientamento orizzontale che viene agganciata al lato inferiore di un'istanza di campo di testo chiamata **myTxt**, caricata con il testo da un URL. Nell'esempio vengono inoltre impostate le dimensioni della barra di scorrimento in modo che corrisponda alle dimensioni del campo di testo:

- 1 Create un nuovo documento Flash (`ActionScript 3.0`).
- 2 Trascinate il componente `ScrollBar` nel pannello Libreria.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale, quindi immettete il codice `ActionScript` seguente:


```
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 Selezionare Controllo > Prova filmato.

Capitolo 5: Personalizzazione dei componenti UI

Personalizzazione dei componenti dell'interfaccia utente

Potete personalizzare l'aspetto dei componenti nelle applicazioni modificando i due elementi seguenti:

Stili Ogni componente dispone di stili che possono essere impostati per specificare quali valori saranno usati da Flash per il rendering dell'aspetto del componente. Gli stili consentono in genere di specificare gli skin e le icone da usare per i diversi stati di un componente e i valori da applicare per la formattazione del testo e il riempimento.

Skin Uno skin è costituito da un insieme di simboli che compongono l'aspetto grafico di un componente in un determinato stato. Mentre uno stile specifica quale skin usare, uno skin è un elemento grafico usato da Flash per disegnare il componente. L'*associazione di skin* rappresenta il processo di modifica dell'aspetto di un componente attraverso il cambiamento o la sostituzione della sua immagine di origine.

Nota: l'aspetto predefinito dei componenti ActionScript 3.0 può essere considerato un tema (Aeon Halo), ma questi skin sono incorporati nei componenti. A differenza dei componenti ActionScript 2.0, i componenti ActionScript 3.0 non supportano i file dei temi esterni.

Impostazione di stili

Gli stili di un componente specificano in genere i valori per gli skin, le icone, la formattazione del testo e il riempimento usati da Flash per disegnare i vari stati del componente. Ad esempio, il componente Button viene disegnato con uno skin diverso per mostrare lo stato premuto (che corrisponde a quando si fa clic con il mouse sul componente), rispetto allo skin usato per mostrare lo stato non premuto o normale. Viene usato uno skin diverso anche quando lo stato del componente è disattivato, ovvero quando la proprietà `enabled` è impostata su `false`.

Gli stili per i componenti possono essere impostati a livello di documento, classe e istanza. Inoltre, alcune proprietà di stile possono essere ereditate da un componente principale. Ad esempio, il componente List eredita gli stili di ScrollBar, poiché eredita dalla classe BaseScrollPane.

Gli stili consentono di personalizzare un componente nei modi seguenti:

- Impostando gli stili in un'istanza di un componente. Potete modificare le proprietà di colore e testo di un'unica istanza di componente. Questo metodo può risultare efficace in alcune situazioni, tuttavia può richiedere un certo dispendio di tempo nei casi in cui sia necessario impostare le singole proprietà di tutti i componenti presenti in un documento.
- Impostando gli stili per tutti i componenti di un dato tipo in un documento. Per applicare un aspetto coerente a tutti i componenti di un dato tipo, ad esempio tutti i componenti CheckBox o Button di un documento, potete impostare gli stili a livello di componente.

I valori delle proprietà di stile impostati per i contenitori vengono ereditati dai componenti in essi contenuti.

Le modifiche apportate alle proprietà di stile non vengono visualizzate in Flash quando si visualizzano i componenti nello stage con la funzione Anteprima dal vivo.

Nozioni fondamentali sulle impostazioni di stile

Di seguito vengono descritti alcuni punti chiave sull'uso degli stili:

Ereditarietà Per impostazione predefinita, l'elemento secondario di un componente eredita uno stile dal componente principale. Non è possibile impostare l'ereditarietà degli stili in ActionScript.

Precedenza Se lo stile di un componente viene impostato in più modi, viene usato automaticamente il primo stile rilevato da Flash in base all'ordine di precedenza. Gli stili vengono cercati fino a quando non viene trovato un valore, secondo l'ordine seguente:

- 1 Viene cercata una proprietà di stile nell'istanza di componente.
- 2 Se lo stile è ereditario, viene cercato nella gerarchia principale un valore ereditato.
- 3 Viene cercato lo stile nel componente.
- 4 Viene cercata un'impostazione globale in StyleManager.
- 5 Se la proprietà non è ancora definita, assume il valore `undefined`.

Accesso agli stili predefiniti di un componente

Potete accedere agli stili predefiniti di un componente usando il metodo statico `getStyleDefinition()` per la classe del componente. Ad esempio, il codice seguente consente di recuperare gli stili predefiniti del componente `ComboBox` e di visualizzare i valori predefiniti delle proprietà `buttonWidth` e `downArrowDownSkin`:

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

Impostazione e recupero degli stili in un'istanza di componente

Qualsiasi istanza di componente UI può chiamare direttamente i metodi `setStyle()` e `getStyle()` per impostare o recuperare uno stile. La sintassi seguente imposta uno stile e il valore di un'istanza di componente:

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

Questa sintassi recupera uno stile per un'istanza di componente:

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

Il metodo `getStyle()` restituisce il tipo `Object` poiché è in grado di restituire più stili con tipi di dati diversi. Ad esempio, il codice seguente imposta lo stile di carattere per un'istanza `TextArea` (`aTa`), quindi lo recupera usando il metodo `getStyle()`. In questo esempio, il valore restituito viene inserito in un oggetto `TextFormat` per assegnarlo a una variabile `TextFormat`. Senza l'inserimento, il compilatore genera un errore a causa del tentativo di assegnare forzatamente una variabile `Object` a una variabile `TextFormat`.

```
import flash.text.TextFormat;

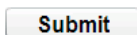
var tf:TextFormat = new TextFormat();
tf.font = "Georgia";
aTa.setStyle("textFormat", tf);
aTa.text = "Hello World!";
var aStyle:TextFormat = aTa.getStyle("textFormat") as TextFormat;
trace(aStyle.font);
```

Usare l'oggetto `TextFormat` per impostare le proprietà di testo

L'oggetto `TextFormat` consente di formattare il testo di un'istanza di componente. L'oggetto `TextFormat` dispone di proprietà che consentono di specificare le caratteristiche del testo quali `bold`, `bullet`, `color`, `font`, `italic`, `size` e molte altre. Potete impostare queste proprietà nell'oggetto `TextFormat`, quindi chiamare il metodo `setStyle()` per applicarle a un'istanza del componente. Ad esempio, il codice seguente imposta le proprietà `font`, `size` e `bold` di un oggetto `TextFormat` e le applica a un'istanza del componente `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);
```

Nella figura seguente è illustrato l'effetto di queste impostazioni su un pulsante con l'etichetta `Submit`:



Le proprietà di stile impostate in un'istanza di componente mediante il metodo `setStyle()` hanno la priorità più elevata e sostituiscono tutte le altre impostazioni di stile. L'impostazione di un numero elevato di proprietà mediante il metodo `setStyle()` per un'unica istanza di componente causa tuttavia il rallentamento del rendering del componente in fase di runtime.

Impostazione di uno stile per tutte le istanze di un componente

Potete accedere agli stili predefiniti per tutte le istanze di un componente usando il metodo statico `setComponentStyle()` per la classe `StyleManager`. Ad esempio, potete impostare il colore del testo su rosso per tutte le istanze `Button` trascinando prima un componente `Button` nello stage, quindi aggiungendo il codice `ActionScript` seguente al pannello Azioni sul fotogramma 1 della linea temporale:

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

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

Tutte le istanze di `Button` aggiunte successivamente allo stage avranno delle etichette rosse.

Impostare uno stile per tutti i componenti

Potete impostare uno stile per tutti i componenti usando il metodo statico `setStyle()` della classe `StyleManager`.

- 1 Trascinate nello stage un componente `List` e assegnategli il nome di istanza **aList**.
- 2 Trascinate nello stage un componente `Button` e assegnategli il nome di istanza **aButton**.
- 3 Premete **F9** o selezionate Azioni dal menu Finestra per aprire il pannello Azioni (se non è già aperto), quindi immettete il codice seguente nel fotogramma 1 della linea temporale per impostare il colore del testo su rosso per tutti i componenti.

```
import fl.managers.StyleManager;

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

- 4 Aggiungete il codice seguente nel pannello Azioni per inserire il testo nel 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 Selezionate Controllo > Prova filmato o premete Ctrl+Invio per compilare il codice e provare il contenuto. Il testo nell'etichetta del pulsante e nell'elenco sarà di colore rosso.

Informazioni sugli skin

L'aspetto di un componente è costituito da elementi grafici, ad esempio un contorno, un colore di riempimento, delle icone ed eventualmente altri componenti. Un componente ComboBox, ad esempio, contiene un componente List, mentre un componente List contiene un componente ScrollBar. Insieme, gli elementi grafici compongono l'aspetto del componente ComboBox. L'aspetto di un componente, tuttavia, cambia in base al relativo stato corrente. Un componente CheckBox senza etichetta, ad esempio, ha un aspetto analogo a quello illustrato di seguito quando viene visualizzato nell'applicazione:



Componente CheckBox nello stato normale non premuto

Se fate clic con il pulsante del mouse e lo tenete premuto sul componente CheckBox, il relativo aspetto viene modificato come illustrato di seguito:



Componente CheckBox nello stato premuto

Quando rilasciate il pulsante del mouse, viene ripristinato l'aspetto originale di CheckBox, ma al suo interno è ora visualizzato un segno di spunta per indicare che è selezionato.



Componente CheckBox nello stato selezionato

Le icone che rappresentano il componente nei diversi stati vengono collettivamente definite *skin*. Potete modificare l'aspetto di un componente in uno o tutti gli stati disponibili modificando i relativi skin in Flash, esattamente come qualsiasi altro simbolo di Flash. Potete accedere agli skin di un componente in due modi. Il modo più semplice consiste nel trascinare il componente nello stage e fare doppio clic su di esso. Viene aperta una tavolozza degli skin del componente che, per un componente CheckBox, è analoga alla seguente.



Skin di un componente CheckBox

Potete inoltre accedere ai singoli skin di un componente dal pannello Libreria. Quando trascinate un componente nello stage, questo viene copiato anche nella libreria insieme a una cartella con le relative risorse e gli eventuali altri componenti in esso contenuti. Ad esempio, se trascinate un componente ComboBox nello stage, il pannello Libreria contiene anche i componenti List, ScrollBar e TextInput, che sono incorporati in ComboBox, insieme a una cartella di skin per ognuno di essi e una cartella Shared contenente gli elementi condivisi da questi componenti. Potete modificare gli skin per uno qualsiasi di questi componenti aprendo la relativa cartella di skin (ComboBoxSkins, ListSkins, ScrollBarSkins o TextInputSkins) e facendo doppio clic sull'icona dello skin che desiderate modificare. Se ad esempio fate doppio clic su ComboBox_downSkin, lo skin viene aperto in modalità di modifica dei simboli, come illustrato nella figura seguente:



ComboBox_downSkin

Creazione di un nuovo skin

Se desiderate creare un nuovo aspetto per un componente presente nel documento, modificate gli skin del componente. Per accedere agli skin di un componente, fate semplicemente doppio clic sul componente nello stage per aprire la relativa tavolozza di skin. Quindi, fate doppio clic sullo skin che desiderate modificare per aprirlo in modalità di modifica dei simboli. Ad esempio, fate doppio clic sul componente TextArea nello stage per aprirne le risorse in modalità di modifica dei simboli. Impostate la percentuale di ingrandimento su 400% (o superiore, se necessario), quindi modificate il simbolo per cambiarne l'aspetto. Al termine, la modifica avrà effetto su tutte le istanze del componente nel documento. In alternativa, potete fare doppio clic su uno skin particolare nel pannello Libreria per aprirlo nello stage in modalità di modifica dei simboli.

Per modificare gli skin di un componente, potete:

- Creare un nuovo skin per tutte le istanze
- Creare nuovi skin per alcune istanze

Creare uno skin per tutte le istanze

Quando modificate lo skin di un componente, per impostazione predefinita viene modificato l'aspetto di tutte le istanze del componente contenute nel documento. Se desiderate creare aspetti diversi dello stesso componente, dovete duplicare gli skin da modificare, assegnarvi nomi diversi, modificarli, quindi impostare gli stili appropriati per applicarli. Per ulteriori informazioni, vedete “[Creare skin per alcune istanze](#)” a pagina 107.

In questo capitolo viene descritto come modificare uno o più skin per ogni componente UI. Se seguite una di queste procedure per modificare uno o più skin dei componenti UI, la modifica viene applicata a tutte le istanze contenute nel documento.

Creare skin per alcune istanze

Potete creare uno skin per alcune istanze di un componente usando la seguente procedura generale:

- Selezionate lo skin nella cartella Assets del componente nel pannello Libreria.
- Duplicate lo skin e assegnategli un nome di classe univoco.
- Modificate lo skin applicandovi l'aspetto desiderato.
- Chiamate il metodo `setStyle()` per l'istanza del componente per assegnare il nuovo skin allo stile di skin.

Nella seguente procedura viene creato un nuovo skin `selectedDownSkin` per una delle due istanze di `Button`.

- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate due componenti `Button` dal pannello Componenti nello stage e assegnate loro i nomi di istanza **aButton** e **bButton**.
- 3 Aprite il pannello Libreria e, al suo interno, le cartelle Component Assets e ButtonSkins.
- 4 Fate clic sullo skin `selectedDownSkin` per selezionarlo.
- 5 Fate clic con il pulsante destro del mouse per aprire il menu di scelta rapida, quindi selezionate `Duplica`.
- 6 Nella finestra di dialogo `Duplica simbolo`, assegnate al nuovo skin un nome univoco, ad esempio **Button_mySelectedDownSkin**. Fare clic su `OK`.
- 7 Nella cartella Component Assets > ButtonSkins del pannello Libreria, selezionate `Button_mySelectedDownSkin` e fate clic con il pulsante destro del mouse per aprire il menu di scelta rapida. Selezionate `Concatenamento` per aprire la finestra di dialogo `Proprietà del concatenamento`.
- 8 Selezionate la casella di controllo `Esporta per ActionScript`. Lasciate selezionata la casella di controllo `Esporta` nel primo fotogramma e assicuratevi che il nome di classe sia univoco. Fate clic su `OK`, quindi di nuovo su `OK` in risposta al messaggio di avviso in cui è specificato che non è possibile trovare una definizione di classe e che ne verrà creata una.
- 9 Fate doppio clic sullo skin `Button_mySelectedDownSkin` nel pannello Libreria per aprirlo in modalità di modifica dei simboli.
- 10 Fate clic sul riempimento blu al centro dello skin per visualizzare il colore nel selettore `Colore riempimento` nella finestra di ispezione `Proprietà`. Fate clic sul selettore del colore e scegliete il colore `#00CC00` per il riempimento dello skin.
- 11 Fate clic sul pulsante `Indietro` sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 12 Nella finestra di ispezione `Proprietà`, fate clic sulla scheda `Parametri` per ogni pulsante e impostate il parametro `toggle` su `true`.
- 13 Aggiungere il codice seguente nel pannello Azioni al fotogramma 1 nella linea temporale:

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

14 Selezionate Controllo > Prova filmato.

15 Fate clic su ogni pulsante. Come noterete, per lo skin down (selezionato e non selezionato) dell'oggetto Button viene usato il nuovo simbolo di skin.

Personalizzare il componente Button

Potete trasformare un componente Button modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi Elabora > Trasforma. In fase di runtime, usate il metodo `setSize()` o le proprietà valide della classe Button, ad esempio `height` e `width`, `scaleX` e `scaleY`.

Il ridimensionamento del pulsante non cambia le dimensioni dell'icona o dell'etichetta. Il riquadro di delimitazione di componente Button corrisponde al bordo del componente e determina inoltre l'area attiva dell'istanza. Se si aumentano le dimensioni dell'istanza, vengono aumentate anche le dimensioni dell'area attiva. Se il riquadro di delimitazione è troppo piccolo per contenere l'etichetta, questa viene troncata.

Se il componente Button dispone di un'icona e le dimensioni di quest'ultima sono maggiori di quelle del componente, l'icona si estende oltre i bordi del pulsante.

Usare gli stili con il componente Button

Gli stili di un componente Button specificano in genere i valori per gli skin, le icone, la formattazione del testo e il riempimento usati da Flash per disegnare i vari stati del componente.

Nella procedura seguente vengono inseriti due componenti Button nello stage e la relativa proprietà `emphasized` viene impostata su `true` per entrambi i componenti quando l'utente fa clic su uno di essi. Viene inoltre impostato lo stile `emphasizedSkin` su `selectedOverSkin` per il secondo componente Button, in modo che quando l'utente fa clic su di esso vengono visualizzati skin diversi per lo stesso stato dei due componenti Button.

- 1 Create un file Flash (ActionScript 3.0).
- 2 Trascinate due componenti Button nello stage, uno alla volta, e assegnate loro i nomi di istanza **aBtn** e **bBtn**. Nella scheda Parametri della finestra di ispezione Proprietà, assegnate alle istanze le etichette Button A e Button B.
- 3 Aggiungere il codice seguente nel pannello Azioni al fotogramma 1 nella linea temporale:

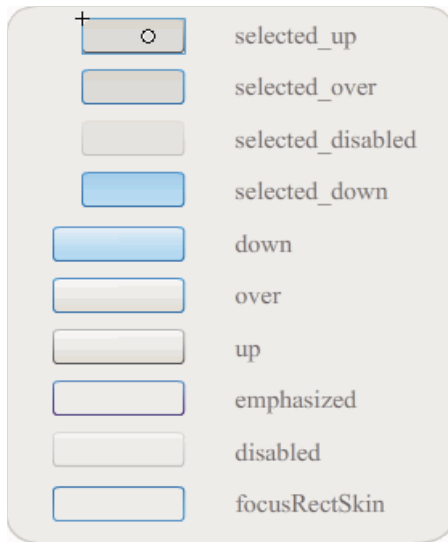
```
bBtn.emphasized = true;  
aBtn.emphasized = true;  
bBtn.addEventListener(MouseEvent.CLICK, Btn_handler);  
function Btn_handler(evt:MouseEvent):void {  
    bBtn.setStyle("emphasizedSkin", "Button_selectedOverSkin");  
}
```

4 Selezionare Controllo > Prova filmato.

5 Fate clic su uno dei pulsanti per verificare l'effetto dello stile `emphasizedSkin` su ogni pulsante.

Usare gli skin con il componente Button

Per rappresentare i diversi stati del componente Button vengono usati gli skin descritti di seguito. Per modificare uno o più skin al fine di cambiare l'aspetto del componente, fate doppio clic sull'istanza Button nello stage per aprire la relativa tavolozza di skin, come illustrato nella figura seguente:

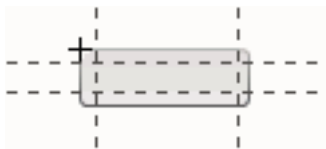


Skin del componente Button

Se un pulsante è abilitato, quando il puntatore passa sopra di esso viene visualizzato lo stato sopra. Quando viene selezionato, il pulsante viene attivato e viene visualizzato lo stato giù. Quando si rilascia il mouse, il pulsante torna allo stato precedente. Se spostate il puntatore dal pulsante mentre il pulsante del mouse è premuto, il pulsante torna allo stato originale. Se il parametro `toggle` viene impostato su `true`, lo stato premuto del pulsante viene visualizzato con lo stile `selectedDownSkin`, lo stato Up con lo stile `selectedUpSkin` e lo stato Over con lo stile `selectedOverSkin`.

Se un componente Button è disabilitato, tale stato viene visualizzato indipendentemente dall'azione dell'utente.

Per modificare uno skin, fate doppio clic su di esso per aprirlo in modalità di modifica dei simboli, come illustrato nella figura seguente:



Componente Button in modalità di modifica dei simboli

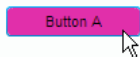
A questo punto potete usare gli strumenti di creazione di Flash per modificare lo skin secondo le esigenze.

La procedura seguente consente di cambiare il colore dello skin `selected_over` del componente Button.

- 1 Create un nuovo file Flash (ActionScript 3.0).
- 2 Trascinate un componente Button dal pannello Componenti allo stage. Nella scheda Parametri, impostate il parametro `toggle` su `true`.
- 3 Fate doppio clic sul componente Button per aprire la tavolozza dei relativi skin.
- 4 Fate doppio clic sullo skin `selected_over` per aprirlo nella modalità di modifica dei simboli.
- 5 Impostate la percentuale di ingrandimento su 400% per aumentare le dimensioni dell'icona per la modifica.
- 6 Fate doppio clic sullo sfondo per visualizzare il colore nel selettore Colore riempimento nella finestra di ispezione Proprietà.
- 7 Selezionate il colore `#CC0099` dal selettore Colore riempimento per applicarlo allo sfondo dello skin `selected_over`.

- Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- Selezionare Controllo > Prova filmato.
- Fate clic sul pulsante per applicarvi lo stato selezionato.

Quando spostate il puntatore del mouse sul componente Button, lo stato `selected_over` dovrebbe apparire come nella figura seguente.



Pulsante con lo skin `selected_over` con il colore modificato

Personalizzare il componente CheckBox

Potete trasformare un componente CheckBox modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi **Elabora > Trasforma**. In fase di runtime, usate il metodo `setSize()` o le proprietà valide della classe CheckBox. Ad esempio, potete modificare le dimensioni del componente CheckBox impostando le relative proprietà `height`, `width`, `scaleX` e `scaleY`. Il ridimensionamento del componente CheckBox non cambia le dimensioni dell'etichetta o dell'icona della casella di controllo, ma solo quelle del riquadro di delimitazione.

Il riquadro di delimitazione di un'istanza CheckBox è invisibile e determina anche l'area attiva dell'istanza. Se si aumentano le dimensioni dell'istanza, vengono aumentate anche le dimensioni dell'area attiva. Se il riquadro di delimitazione è troppo piccolo per contenere l'etichetta, questa viene troncata.

Usare gli stili con il componente CheckBox

L'impostazione delle proprietà di stile consente di modificare l'aspetto di un'istanza CheckBox. La procedura riportata di seguito, ad esempio, consente di modificare le dimensioni e il colore dell'etichetta di un componente CheckBox.

- Trascinate un componente CheckBox dal pannello Componenti allo stage e assegnategli il nome di istanza **myCb**.
- Fate clic sulla scheda Parametri della finestra di ispezione Proprietà e inserite il seguente valore per il parametro label: **Less than \$500?**
- Nel fotogramma 1 della linea temporale principale, immettete il codice seguente nel pannello Azioni:

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

Per ulteriori informazioni, vedete [“Impostazione di stili”](#) a pagina 102. Per informazioni sull'impostazione delle proprietà di stile per modificare le icone e gli skin del componente, vedete [“Creazione di un nuovo skin”](#) a pagina 106 e [“Usare gli skin con il componente CheckBox”](#) a pagina 110.

Usare gli skin con il componente CheckBox

Il componente CheckBox dispone dei seguenti stili che possono essere modificati per cambiarne l'aspetto.



Skin del componente CheckBox

Nell'esempio vengono modificati il colore del contorno e il colore dello sfondo del componente negli stati `up` e `selectedUp`. Per cambiare altri stati degli skin, procedete in modo analogo.

- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate il componente CheckBox nello stage; in questo modo il componente viene inserito anche nel pannello Libreria con la relativa cartella di risorse.
- 3 Fate doppio clic sul componente CheckBox nello stage per aprire il relativo pannello di icone di skin.
- 4 Fate doppio clic sullo skin `selected_up` per aprirlo in modalità di modifica dei simboli.
- 5 Impostate la percentuale di ingrandimento su 800% per aumentare le dimensioni dell'icona per la modifica.
- 6 Fate clic sul bordo del componente CheckBox per selezionarlo. Utilizzate il selettore Colore riempimento nella finestra di ispezione Proprietà per selezionare il colore #0033FF e applicarlo al bordo.
- 7 Fate doppio clic sullo sfondo del componente CheckBox per selezionarlo e usate di nuovo il selettore Colore riempimento per impostare il colore di sfondo su #00CCFF.
- 8 Ripetete i passaggi da 4 a 8 per lo skin `up` del componente CheckBox.
- 9 Selezionare Controllo > Prova filmato.

Personalizzare il componente ColorPicker

Il ridimensionamento di un componente ColorPicker è possibile solo mediante i relativi stili: `swatchWidth`, `swatchHeight`, `backgroundPadding`, `textFieldWidth` e `textFieldHeight`. Se tentate di modificare le dimensioni del componente ColorPicker con lo strumento Trasformazione o mediante ActionScript usando il metodo `setSize()` o la proprietà `width`, `height`, `scaleX` o `scaleY`, i valori impostati vengono ignorati quando create il file SWF e ColorPicker viene visualizzato con le dimensioni predefinite. Lo sfondo della tavolozza viene ridimensionato in base al numero di colonne impostate usando `setStyle()` per lo stile `columnCount`. Il numero di colonne predefinito è 18. Potete impostare i colori personalizzati su 1024; la tavolozza viene ridimensionata in verticale in modo che corrisponda al numero di campioni.

Usare gli stili con il componente ColorPicker

Potete impostare numerose proprietà di stile per modificare l'aspetto di un componente ColorPicker. Nella procedura seguente, ad esempio, il numero di colonne (`columnCount`) nel componente ColorPicker viene impostato su 12, vengono modificate l'altezza (`swatchHeight`) e la larghezza (`swatchWidth`) dei campioni di colore e il riempimento per il campo di testo (`textPadding`) e lo sfondo (`backgroundPadding`).

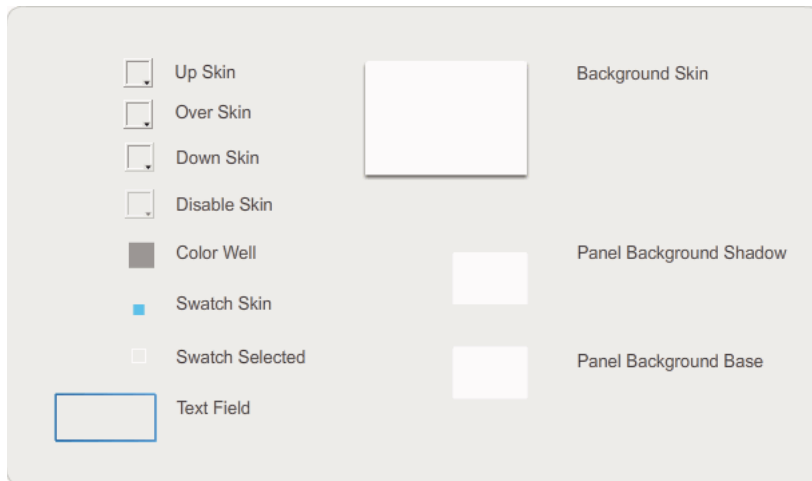
- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate il componente ColorPicker nello stage e assegnategli il nome di istanza **aCp**.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale e immettete il codice seguente:

```
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 Selezionare Controllo > Prova filmato.
- 5 Fate clic sul componente ColorPicker per aprirlo e verificare se l'aspetto è stato modificato da queste impostazioni.

Usare gli skin con il componente ColorPicker

Il componente ColorPicker usa i seguenti skin per rappresentare i relativi stati visivi.



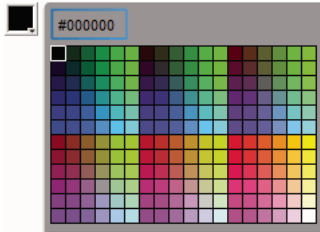
Skin del componente ColorPicker

Potete cambiare il colore dello skin di sfondo per modificare il colore dello sfondo della tavolozza.

- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate il componente ColorPicker nello stage.
- 3 Fate doppio clic sul componente per aprire la tavolozza dei relativi skin.
- 4 Fate doppio clic sullo skin Background per selezionarlo; viene visualizzato il selettore Colore riempimento nella finestra di ispezione Proprietà.
- 5 Selezionate il colore #999999 usando il selettore Colore riempimento per applicarlo allo skin di sfondo.

- 6 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 7 Selezionare Controllo > Prova filmato.

Quando fate clic sul componente ColorPicker, lo sfondo della tavolozza dovrebbe essere grigio, come nella figura illustrata di seguito.



ColorPicker con skin Background grigio

Personalizzare il componente ComboBox

Potete trasformare un componente ComboBox modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi **Elabora > Trasforma**. In fase di runtime, usate il metodo `setSize()` o le proprietà valide della classe `ComboBox` quali `height`, `width`, `scaleX` e `scaleY`.

Il componente `ComboBox` viene ridimensionato in base alla larghezza e all'altezza specificate. L'elenco viene ridimensionato in base alla larghezza del componente, a meno che sia stata impostata la proprietà `dropdownWidth`.

Se il testo è troppo lungo, viene troncato per essere adattato al componente `ComboBox`. È necessario ridimensionare `ComboBox` e impostare la proprietà `dropdownWidth` per adattare il testo.

Usare gli stili con il componente ComboBox

L'impostazione delle proprietà degli stili consente di modificare l'aspetto di un componente `ComboBox`. Mediante gli stili potete specificare i valori per gli skin, il renderer di celle, il riempimento e la larghezza del pulsante relativi al componente. Nell'esempio seguente vengono impostati gli stili `buttonWidth` e `textPadding`. Lo stile `buttonWidth` consente di impostare la larghezza dell'area attiva del pulsante, ed è disponibile quando il componente `ComboBox` è modificabile ed è possibile solo premere il pulsante per aprire l'elenco a discesa. Lo stile `textPadding` consente di specificare la quantità di spazio tra il bordo esterno del campo di testo e il testo. È utile per centrare verticalmente il testo nel campo di testo se aumentate l'altezza del componente `ComboBox`, in caso contrario il testo potrebbe essere visualizzato nella parte superiore del campo di testo.

- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate il componente `ComboBox` nello stage e assegnategli il nome di istanza **aCb**.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale e immettete il codice seguente:

```
import fl.data.DataProvider;

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

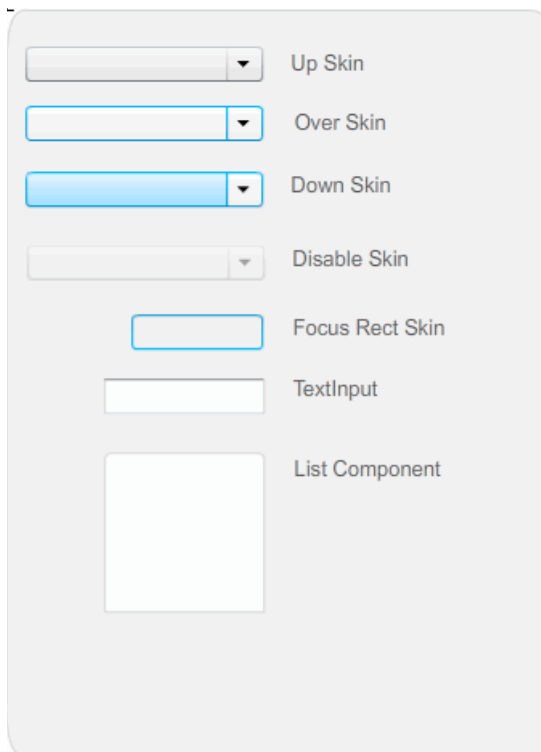
var items:Array = [
    {label:"San Francisco", data:"601 Townsend St."},
    {label:"San Jose", data:"345 Park Ave."},
    {label:"San Diego", data:"10590 West Ocean Air Drive, Suite 100"},
    {label:"Santa Rosa", data:"2235 Mercury Way, Suite 105"},
    {label:"San Luis Obispo", data:"3220 South Higuera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

4 Selezionare Controllo > Prova filmato.

L'area del pulsante su cui potete fare clic per aprire l'elenco a discesa è solamente un'area sottile presente sul lato destro e il testo è centrato verticalmente nel campo di testo. Potete provare a eseguire l'esempio senza le due istruzioni `setStyle()` per verificarne l'effetto.

Usare gli skin con il componente ComboBox

Il componente ComboBox usa i seguenti skin per rappresentare i relativi stati visivi:



Skin del componente ComboBox

Potete cambiare il colore dello skin Up per modificare il colore del componente nello stato inattivo sullo stage.

1 Create un nuovo documento File Flash (ActionScript 3.0).

- 2 Trascinate il componente ComboBox nello stage.
- 3 Fate doppio clic sul componente per aprire la tavolozza dei relativi skin.
- 4 Fate doppio clic sullo skin Up per selezionarlo e aprirlo per la modifica.
- 5 Impostate la percentuale di ingrandimento su 400%.
- 6 Fate clic sull'area centrale dello skin per visualizzare il colore nel selettore Colore riempimento nella finestra di ispezione Proprietà.
- 7 Selezionate il colore #33FF99 usando il selettore Colore riempimento per applicarlo allo skin Up.
- 8 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 9 Selezionare Controllo > Prova filmato.

Il componente ComboBox dovrebbe essere visualizzato nello stage come nella figura illustrata di seguito.



ComboBox con colore personalizzato per lo skin Background

Personalizzare il componente DataGrid

Potete trasformare un componente DataGrid modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi *Elabora > Trasforma*. In fase di runtime, usate il metodo `setSize()` o le proprietà valide della classe `Button` quali `width`, `height`, `scaleX` e `scaleY`. Se non è presente alcuna barra di scorrimento orizzontale, le larghezze delle colonne si adattano in maniera proporzionale. Se le colonne si adattano automaticamente, e quindi anche le celle, il testo delle celle potrebbe essere troncato.

Usare gli stili con il componente DataGrid

È possibile impostare le proprietà degli stili per modificare l'aspetto di un componente DataGrid. Il componente DataGrid eredita gli stili dal componente List. (Vedete [“Usare gli stili con il componente List”](#) a pagina 121.)

Impostare gli stili per una singola colonna

Un oggetto DataGrid può avere più colonne ed è possibile specificare diversi renderer di celle per ogni colonna. Ogni colonna di un oggetto DataGrid è rappresentata da un oggetto `DataGridColumn` e la classe `DataGridColumn` include una proprietà `cellRenderer` per cui viene definito l'oggetto `CellRenderer` per la colonna.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente DataGrid nel pannello Libreria.
- 3 Aggiungete il codice seguente nel pannello Azioni al fotogramma 1 della linea temporale. Questo codice consente di creare un componente DataGrid con una lunga stringa di testo nella terza colonna. Al termine, la proprietà `cellRenderer` della colonna viene impostata sul nome di un renderer di celle che visualizza una cella multiriga.

```
/* 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 Salvate il file FLA come MultiLineGrid fla.
- 5 Create un nuovo file ActionScript.
- 6 Copiate il codice ActionScript seguente nella finestra 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 Salvate il file ActionScript come MultiLineCell.as nella stessa cartella in cui è stato salvato il file MultiLineGrid fla.
- 8 Tornate all'applicazione MultiLineGrid fla e selezionate Controllo > Prova filmato.

Il componente DataGrid dovrebbe essere simile al seguente:

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 per l'applicazione MultiLineGrid fla

Impostare gli stili dell'intestazione

Potete impostare lo stile del testo per una riga di intestazione usando lo stile `headerTextFormat`. Nell'esempio seguente viene usato l'oggetto `TextFormat` per impostare lo stile `headerTextFormat` affinché vengano usati il carattere Arial, il colore rosso, la dimensione del carattere 14 e lo stile corsivo.

- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate il componente DataGrid nello stage e assegnategli il nome di istanza **aDg**.
- 3 Aprite il pannello Azioni, selezionate il fotogramma 1 nella linea temporale principale e immettete il codice seguente:

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

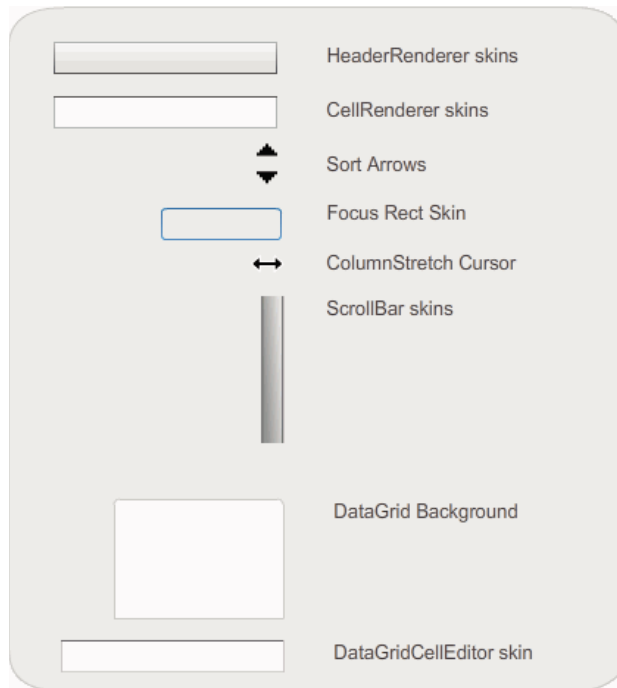
var myDP:Array = new Array();
myDP = [{FirstName:"Winston", LastName:"Elstad"},
        {FirstName:"Ric", LastName:"Dietrich"},
        {FirstName:"Ewing", LastName:"Canepa"},
        {FirstName:"Kevin", LastName:"Wade"},
        {FirstName:"Kimberly", LastName:"Dietrich"},
        {FirstName:"AJ", LastName:"Bilow"},
        {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 Selezionate Controllo > Prova filmato per eseguire l'applicazione.

Usare gli skin con il componente DataGrid

Il componente DataGrid usa i seguenti skin per rappresentare i relativi stati visivi:



Skin del componente DataGrid

Lo skin CellRenderer viene usato per le celle del corpo del componente DataGrid, mentre lo skin HeaderRenderer viene usato per la riga di intestazione. Nella procedura seguente viene modificato il colore di sfondo della riga di intestazione, ma potete seguire lo stesso processo per modificare il colore di sfondo delle celle del corpo di DataGrid mediante la modifica dello skin CellRenderer.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente DataGrid nello stage e assegnategli il nome di istanza **aDg**.
- 3 Fate doppio clic sul componente per aprire la tavolozza dei relativi skin.
- 4 Impostate la percentuale di ingrandimento su 400% per aumentare le dimensioni delle icone per la modifica.
- 5 Fate doppio clic sullo skin HeaderRenderer per aprire la tavolozza degli skin HeaderRenderer.
- 6 Fate doppio clic sullo skin Up_Skin per aprirlo in modalità di modifica dei simboli, quindi fate clic sul suo sfondo per selezionarlo; viene visualizzato il selettore Colore riempimento nella finestra di ispezione Proprietà.
- 7 Selezionate il colore #00CC00 dal selettore Colore riempimento per applicarlo allo sfondo dello skin HeaderRenderer Up_Skin.
- 8 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 9 Aggiungete il codice seguente nel pannello Azioni al fotogramma 1 nella linea temporale per aggiungere i dati 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 Selezionate Controllo > Prova filmato per provare l'applicazione.

Il componente DataGrid dovrebbe apparire come nella figura seguente con lo sfondo della riga di intestazione 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 sfondo personalizzato della riga di intestazione

Personalizzare il componente Label

Potete trasformare un componente Label modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi **Elabora > Trasforma**. Inoltre, potete impostare il parametro di creazione `autoSize`; in questo caso il riquadro di delimitazione non viene modificato nell'anteprima dal vivo, mentre il componente List viene effettivamente ridimensionato. Il ridimensionamento di Label dipende dal parametro `wordwrap`. Se il parametro è impostato su `true`, il componente Label viene ridimensionato verticalmente per adattarvi il testo. Se il parametro è impostato su `false`, il componente Label viene ridimensionato orizzontalmente. In fase di runtime, usate il metodo `setSize()`. Per ulteriori informazioni, vedete il metodo `Label.setSize()` e la proprietà `Label.autoSize` nella *Guida di riferimento ActionScript 3.0 per Flash Professional*. Vedete anche [“Creare un'applicazione con il componente Label”](#) a pagina 65.

Usare gli stili con il componente Label

È possibile impostare le proprietà degli stili per cambiare l'aspetto di un'istanza Label. Tutto il testo in un'istanza del componente Label deve condividere lo stesso stile. Il componente Label dispone di uno stile `textFormat` con gli stessi attributi dell'oggetto `TextFormat` e consente di impostare le stesse proprietà per il contenuto di `Label.text` come per un normale oggetto `TextField` di Flash. Nell'esempio seguente, il colore del testo in un'etichetta viene impostato su rosso.

- 1 Trascinate il componente Label dal pannello Componenti allo stage e assegnategli il nome di istanza `a_label`.
- 2 Fate clic sulla scheda Parametri e sostituite il valore della proprietà `text` con il testo seguente:

Color me red

- 3 Selezionare il fotogramma 1 nella linea temporale principale, aprire il pannello Azioni e immettere il codice seguente:

```
/* 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 Selezionate Controllo > Prova filmato.

Per ulteriori informazioni sugli stili del componente Label, vedete la classe Label nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Skin e componente Label

Nel componente Label non sono disponibili elementi visivi da associare a uno skin.

Personalizzare il componente List

Potete trasformare un componente List modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi *Elabora > Trasforma*. In fase di runtime, usate il metodo `setSize()` e le proprietà valide della classe List quali `height`, `width`, `scaleX` e `scaleY`.

Quando un elenco viene ridimensionato, le righe dell'elenco si riducono in orizzontale impedendo la completa visualizzazione del testo. In verticale, vengono aggiunte o rimosse le righe secondo le necessità. Le barre di scorrimento vengono posizionate automaticamente secondo le esigenze.

Usare gli stili con il componente List

È possibile impostare le proprietà di stile per modificare l'aspetto di un componente List. Mediante gli stili potete specificare i valori per gli skin e il riempimento del componente quando viene disegnato.

I diversi stili dello skin consentono di specificare le diverse classi da usare per lo skin. Per ulteriori informazioni sull'uso degli stili degli skin, vedete ["Informazioni sugli skin"](#) a pagina 105.

Nella procedura seguente viene impostato il valore dello stile `contentPadding` per il componente `List`. Il valore di questa impostazione viene sottratto dalle dimensioni di `List` per ottenere il riempimento intorno al contenuto, quindi potrebbe essere necessario aumentare le dimensioni del componente `List` per impedire che il testo in esso contenuto venga ritagliato.

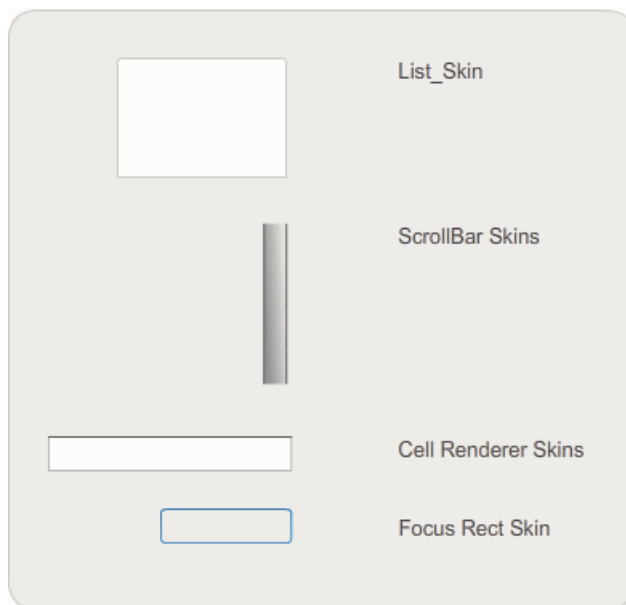
- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate il componente `List` dal pannello Componenti allo stage e assegnategli il nome di istanza **aList**.
- 3 Selezionate il fotogramma 1 nella linea temporale principale, aprite il pannello Azioni e immettete il codice seguente, che imposta lo stile `contentPadding` e aggiunge i dati a `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 Selezionare Controllo > Prova filmato.

Usare gli skin con il componente List

Il componente `List` usa i seguenti skin per rappresentare i relativi stati visivi:

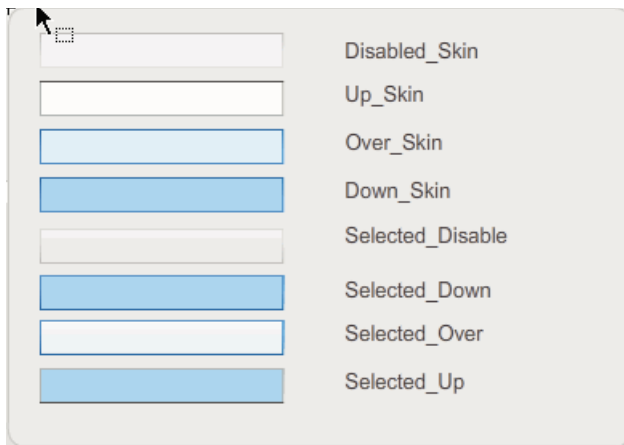


Skin del componente List

Per ulteriori informazioni sull'associazione di skin al componente `ScrollBar`, vedete “[Personalizzare il componente UI ScrollBar](#)” a pagina 137. Per ulteriori informazioni sull'associazione di skin a Focus Rect, vedete “[Personalizzare il componente TextArea](#)” a pagina 131.

Nota: la modifica dello skin `ScrollBar` in un componente viene applicata a tutti gli altri componenti che usano `ScrollBar`.

Fate doppio clic sullo skin `Cell Renderer` per aprire una seconda tavolozza di skin per stati diversi di una cella `List`.



Skin Cell Renderer del componente List

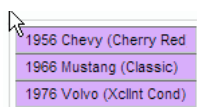
L'aspetto delle celle del componente List può essere cambiato mediante la modifica di questi skin. Nella procedura seguente viene cambiato il colore dello skin Up per modificare l'aspetto del componente List nel suo normale stato inattivo.

- 1 Create un nuovo documento File Flash (ActionScript 3.0).
- 2 Trascinate il componente List dal pannello Componenti allo stage e assegnategli il nome di istanza **aList**.
- 3 Fate doppio clic sul componente List per aprire la tavolozza dei relativi skin.
- 4 Fate doppio clic sullo skin Cell Renderer per aprire la tavolozza dei relativi skin.
- 5 Fate doppio clic sullo skin Up_Skin per aprirlo e modificarlo.
- 6 Fate clic sull'area di riempimento dello skin per selezionarlo. Nella finestra di ispezione Proprietà dovrebbe essere visualizzato un selettore Colore riempimento con il colore di riempimento corrente dello skin.
- 7 Selezionate il colore #CC66FF dal selettore Colore riempimento per applicarlo al riempimento dello skin Up_Skin.
- 8 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 9 Aggiungete il codice seguente nel pannello Azioni al fotogramma 1 nella linea temporale per aggiungere dati 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 Selezionare Controllo > Prova filmato.

Il componente List dovrebbe essere visualizzato come illustrato nella figura seguente:



Celle List con colore Up_Skin personalizzato

La cornice viene generata dall'impostazione dello stile `contentPadding`.

Personalizzare il componente NumericStepper

Potete trasformare un componente NumericStepper modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi **Elabora > Trasforma**. In fase di runtime, usate il metodo `setSize()` o le proprietà e i metodi validi della classe NumericStepper quali `width`, `height`, `scaleX` e `scaleY`.

Il ridimensionamento del componente NumericStepper non modifica la larghezza dei pulsanti freccia giù e freccia su. Se lo stepper viene ridimensionato con un'altezza maggiore rispetto a quella predefinita, i pulsanti freccia vengono ancorati nella parte superiore e inferiore del componente per impostazione predefinita. In caso contrario, la modifica in scala a 9 porzioni determina il modo in cui vengono disegnati i pulsanti. Essi vengono sempre visualizzati a destra della casella di testo.

Stili e componente NumericStepper

Potete impostare le proprietà di stile del componente NumericStepper per modificarne l'aspetto. Mediante gli stili potete specificare i valori per gli skin, il riempimento e il formato del testo del componente quando viene disegnato. Lo stile `textFormat` consente di modificare le dimensioni e l'aspetto del valore di NumericStepper. I diversi stili dello skin consentono di specificare le diverse classi da usare per lo skin. Per ulteriori informazioni sull'uso degli stili degli skin, vedete ["Informazioni sugli skin"](#) a pagina 105.

In questa procedura viene usato lo stile `textFormat` per modificare l'aspetto del valore visualizzato dal componente NumericStepper.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente NumericStepper dal pannello Componenti allo stage e assegnategli il nome di istanza **myNs**.
- 3 Aggiungete il codice seguente nel pannello Azioni al fotogramma 1 della linea temporale principale:

```
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 Selezionare **Controllo > Prova filmato**.

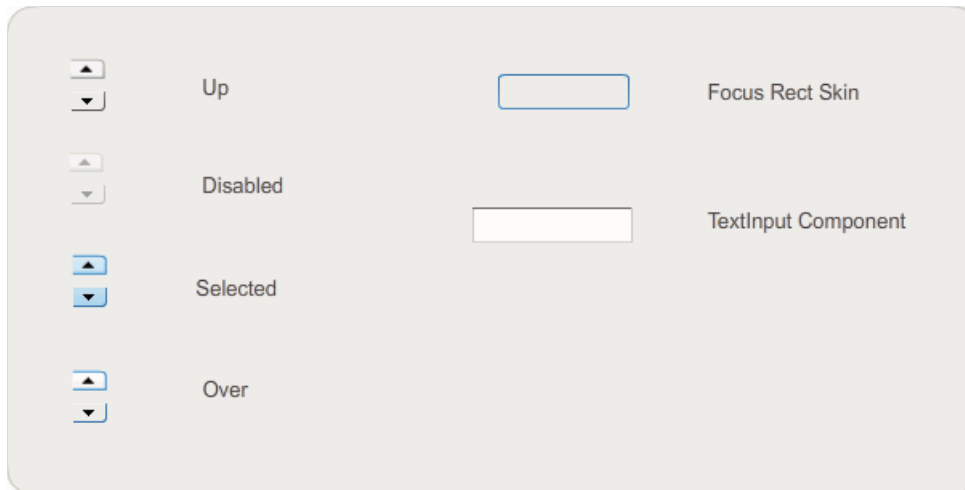
Skin e componente NumericStepper

Il componente NumericStepper dispone di skin per rappresentare gli stati su, giù, disattivato e selezionato dei relativi pulsanti.

Se uno stepper è attivo, il pulsante giù e il pulsante su visualizzano lo stato sopra quando il puntatore passa su di essi. I pulsanti visualizzano lo stato giù quando vengono premuti e tornano allo stato sopra quando il pulsante del mouse viene rilasciato. Se il puntatore si sposta dai pulsanti mentre il pulsante del mouse è premuto, i pulsanti tornano allo stato originale.

Se uno stepper è disabilitato, visualizza tale stato indipendentemente dall'azione dell'utente.

Un componente NumericStepper dispone dei seguenti skin:



Skin del componente NumericStepper

- 1 Creare un nuovo file FLA.
- 2 Trascinate il componente NumericStepper nello stage.
- 3 Impostate la percentuale di ingrandimento su 400% per aumentare le dimensioni dell'immagine per la modifica.
- 4 Fate doppio clic sullo sfondo dello skin TextInput nel pannello degli skin e accedete al livello relativo al gruppo finché non viene visualizzato il colore di sfondo nel selettore Colore riempimento della finestra di ispezione Proprietà.
- 5 Mediante il selettore Colore riempimento della finestra di ispezione Proprietà, selezionate il colore #9999FF per applicarlo allo sfondo dello skin TextInput.
- 6 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 7 Fate di nuovo doppio clic sul componente NumericStepper per riaprire la tavolozza degli skin.
- 8 Fate doppio clic sullo sfondo del pulsante freccia su nel gruppo Up per selezionarlo; il relativo colore viene visualizzato nel selettore Colore riempimento della finestra di ispezione Proprietà.
- 9 Selezionate il colore #9966FF per applicarlo allo sfondo del pulsante freccia su.
- 10 Ripetete i punti 8 e 9 per il pulsante freccia giù nel gruppo Up.
- 11 Selezionare Controllo > Prova filmato.

L'istanza NumericStepper dovrebbe apparire come mostrato nella figura seguente:



Personalizzare il componente ProgressBar

Potete trasformare un componente ProgressBar modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi Elabora > Trasforma. In fase di runtime, usate il metodo `setSize()` o le proprietà valide della classe ProgressBar quali `height`, `width`, `scaleX` e `scaleY`.

Il componente ProgressBar è dotato di tre skin: a uno skin per la traccia, uno skin per la barra e uno skin indeterminato. Per ridimensionare le risorse, viene usata la modifica in scala a 9 porzioni.

Stili e componente ProgressBar

L'impostazione delle proprietà di stile consente di modificare l'aspetto dell'istanza ProgressBar. Mediante gli stili di ProgressBar potete specificare i valori per gli skin e il riempimento quando il componente viene disegnato. Nell'esempio seguente vengono aumentate le dimensioni dell'istanza ProgressBar e viene impostato il relativo stile `barPadding`.

- 1 Creare un nuovo file FLA.
- 2 Trascinate un componente ProgressBar dal pannello Componenti allo stage e assegnategli il nome di istanza **myPb**.
- 3 Nel fotogramma 1 della linea temporale principale, immettete il codice seguente nel pannello Azioni:

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

- 4 Selezionare Controllo > Prova filmato.

Per informazioni sull'impostazione degli stili di skin, vedete ["Informazioni sugli skin"](#) a pagina 105.

Skin e componente ProgressBar

Per rappresentare la traccia della barra di avanzamento, la barra completata e una barra indeterminata per il componente ProgressBar, come nella figura illustrata di seguito, vengono usati degli skin.



Skin del componente ProgressBar

La barra viene collocata sopra lo skin per la traccia usando `barPadding` per determinare la posizione. Le risorse vengono ridimensionate utilizzando la modifica in scala a 9 porzioni.

La barra indeterminata viene usata quando la proprietà `indeterminate` dell'istanza ProgressBar è impostata su `true`. Lo skin viene ridimensionato in verticale e in orizzontale per adattarsi alle dimensioni del componente ProgressBar.

Potete modificare questi skin per cambiare l'aspetto del componente ProgressBar. L'esempio seguente consente ad esempio di modificare il colore della barra indeterminata:

- 1 Creare un nuovo file FLA.
- 2 Trascinate un componente ProgressBar nello stage e fate doppio clic sul componente per aprire il relativo pannello di icone di skin.
- 3 Fate doppio clic sullo skin della barra indeterminata.
- 4 Impostate la percentuale di ingrandimento su 400% per aumentare le dimensioni dell'icona per la modifica.

- 5 Fate doppio clic su una delle barre diagonali, quindi tenete premuto il tasto Maiusc e fate clic su ognuna delle altre barre. Il colore corrente viene visualizzato nel selettore Colore riempimento della finestra di ispezione Proprietà.
- 6 Fate clic sul selettore Colore riempimento nella finestra di ispezione Proprietà per aprirlo e selezionare il colore #00CC00 per applicarlo alle barre diagonali selezionate.
- 7 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 8 Selezionare Controllo > Prova filmato.

Il componente ProgressBar dovrebbe essere visualizzato come illustrato nella figura seguente.



Personalizzare il componente RadioButton

Potete trasformare un componente RadioButton modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi Elabora > Trasforma. In fase di runtime, usate il metodo `setSize()`.

Il riquadro di delimitazione di un componente RadioButton è invisibile e determina anche l'area attiva del componente. Se si aumentano le dimensioni del componente, vengono aumentate anche le dimensioni dell'area attiva.

Se il riquadro di delimitazione del componente è troppo piccolo per adattarvi l'etichetta del componente, questa viene troncata.

Usare gli stili con il componente RadioButton

È possibile impostare le proprietà degli stili per modificare l'aspetto di un componente RadioButton. Le proprietà degli stili di ScrollPane consentono di specificare i valori per gli skin, le icone, la formattazione del testo e il riempimento quando il componente viene disegnato. Mediante gli stili di RadioButton potete specificare i valori per gli skin e il riempimento per il layout quando il componente viene disegnato.

Nell'esempio seguente viene recuperato lo stile `textFormat` da un componente CheckBox e viene applicato al componente RadioButton per rendere identico lo stile delle etichette.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate nello stage un componente CheckBox e assegnategli il nome di istanza **my_Ch** nella finestra di ispezione Proprietà.
- 3 Trascinate nello stage un componente RadioButton e assegnategli il nome di istanza **my_Rb** nella finestra di ispezione Proprietà.
- 4 Aggiungere il codice seguente nel pannello Azioni al fotogramma 1 nella linea temporale.

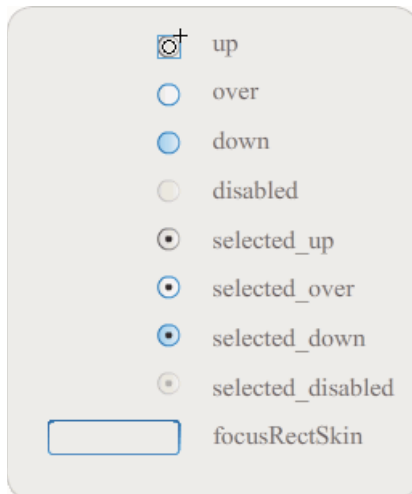
```
var tf:TextFormat = new TextFormat();
tf.color = 0x00FF00;
tf.font = "Georgia";
tf.size = 18;
myCh.setStyle("textFormat", tf);
myRb.setStyle("textFormat", myCh.getStyle("textFormat"));
```

Con questo codice viene impostato lo stile `textFormat` per `CheckBox`, che viene quindi applicato al componente `RadioButton` mediante una chiamata al metodo `getStyle()` sul componente `CheckBox`.

5 Selezionare Controllo > Prova filmato.

Skin e componente RadioButton

Il componente `RadioButton` dispone dei seguenti stili che possono essere modificati per cambiarne l'aspetto:



Skin del componente RadioButton

Se un componente `RadioButton` è attivato e non selezionato, quando un utente vi sposta sopra il puntatore, viene visualizzato il relativo skin `over`. Quando un utente fa clic su un componente `RadioButton`, il pulsante si attiva e viene visualizzato il relativo skin `selected_down`. Quando il pulsante del mouse viene rilasciato, il componente `RadioButton` viene visualizzato con il relativo skin `selected_up`. Se un utente allontana il puntatore dall'area attiva del componente `RadioButton` mentre tiene premuto il pulsante del mouse, il componente `RadioButton` viene rivisualizzato con lo skin `up`.

Se un componente `RadioButton` è disabilitato, visualizza tale stato indipendentemente dall'azione dell'utente.

Nell'esempio seguente viene sostituito lo skin `selected_up` che indica lo stato selezionato.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate un componente `RadioButton` nello stage e fate doppio clic sul componente per aprire la relativa tavolozza di skin.
- 3 Impostate la percentuale di ingrandimento su 800% per aumentare le dimensioni dell'icona per la modifica.
- 4 Fate doppio clic sullo skin `selected_up` skin per selezionarlo e premete il tasto Canc per eliminarlo.
- 5 Nel pannello Strumenti, selezionate lo strumento Rettangolo.
- 6 Nella finestra di ispezione Proprietà, impostate il colore della linea su rosso (`#FF0000`) e il colore del riempimento su nero (`#000000`).
- 7 Iniziando dal mirino che indica il punto di registrazione del simbolo (detto anche *punto di origine* o *punto zero*), trascinate il puntatore per disegnare un rettangolo.
- 8 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 9 Selezionare Controllo > Prova filmato.

10 Fate clic sul componente RadioButton per selezionarlo.

Il componente RadioButton nello stato selezionato dovrebbe essere simile a quello illustrato nella figura di seguito.



Personalizzare il componente ScrollPane

Potete trasformare un componente ScrollPane modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi *Elabora > Trasforma*. In fase di runtime, usate il metodo `setSize()` o le proprietà e i metodi validi della classe ScrollPane quali `height`, `width`, `scaleX` e `scaleY`.

Il componente ScrollPane ha le seguenti caratteristiche grafiche:

- Il punto di registrazione (detto anche *punto di origine* o *punto zero*) del contenuto si trova nell'angolo superiore sinistro del riquadro.
- Quando la barra di scorrimento orizzontale è disattivata, la barra di scorrimento verticale viene visualizzata dall'alto verso il basso lungo il lato destro del riquadro di scorrimento. Quando invece è disattivata la barra di scorrimento verticale, la barra di scorrimento orizzontale viene visualizzata da sinistra a destra lungo il lato inferiore del riquadro di scorrimento. È anche possibile disattivare entrambe le barre di scorrimento.
- Se il riquadro di scorrimento è di dimensioni troppo piccole, il contenuto potrebbe non essere visualizzato correttamente.
- Quando il riquadro di scorrimento viene ridimensionato, la traccia di scorrimento e la casella di scorrimento si espandono o si comprimono e le aree attive vengono ridimensionate. Le dimensioni dei pulsanti rimangono invariate.

Usare gli stili con il componente ScrollPane

Mediante le proprietà degli stili del componente ScrollPane potete specificare i valori per gli skin e il riempimento per il layout quando il componente viene disegnato. I diversi stili dello skin consentono di specificare le diverse classi da usare per gli skin del componente. Per ulteriori informazioni sull'uso degli stili degli skin, vedete “[Informazioni sugli skin](#)” a pagina 105.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate nello stage un componente ScrollPane e assegnategli il nome di istanza **mySp**.
- 3 Fate clic sulla scheda Parametri della finestra di ispezione Proprietà e inserite il seguente valore per il parametro `source`: **`http://www.helpexamples.com/flash/images/image1.jpg`**.

- 4 Nel fotogramma 1 della linea temporale principale, aggiungete il codice seguente nel pannello Azioni.

```
mySp.setStyle("contentPadding", 5);
```

Il riempimento viene applicato tra il bordo del componente e il relativo contenuto, all'esterno delle barre di scorrimento.

- 5 Selezionare *Controllo > Prova filmato*.

Skin e componente ScrollPane

Il componente ScrollPane usa il bordo e le barre di scorrimento per scorrere gli elementi. Per informazioni sull'associazione degli skin alle barre di scorrimento, vedete [“Usare gli skin con il componente UIScrollBar”](#) a pagina 137.

Personalizzare il componente Slider

Potete trasformare un componente Slider modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi *Elabora > Trasforma*. In fase di runtime, usate il metodo `setSize()` o le proprietà valide della classe Slider quali `width` e `scaleX`.

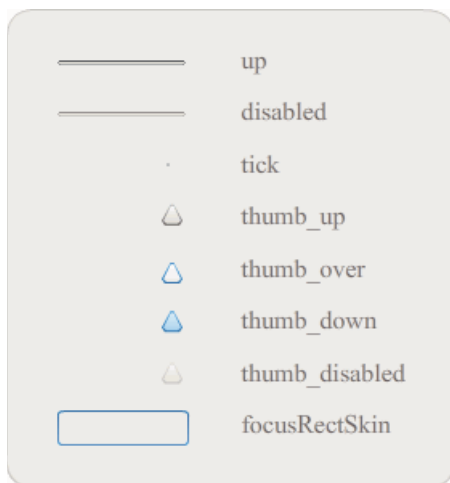
Potete aumentare la lunghezza di un componente Slider, ma non l'altezza. La proprietà `height` e il parametro `height` del metodo `setSize()` vengono ignorati da Flash. Tuttavia, potete creare un cursore verticale e modificarne la lunghezza in verticale.

Stili e componente Slider

Gli stili del componente Slider consentono di specificare solo le classi per i relativi skin e un valore per `FocusRectPadding`, che specifica il numero di pixel da usare per il riempimento tra il riquadro di delimitazione e il bordo esterno del componente. Per ulteriori informazioni sull'uso degli stili degli skin, vedete [“Informazioni sugli skin”](#) a pagina 105.

Skin e componente Slider

Il componente Slider usa i seguenti skin, che possono essere modificati per cambiarne l'aspetto.



Skin del componente Slider

Nell'esempio seguente viene modificato lo skin della traccia Up applicando il colore blu.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente Slider dal pannello Componenti allo stage.
- 3 Fate doppio clic sul componente Slider per aprire il pannello dei relativi skin.

- 4 Fate doppio clic sul contrassegno di registrazione della traccia up per aprire lo skin nella modalità di modifica dei simboli.
- 5 Impostate la percentuale di ingrandimento su 800% per aumentare le dimensioni dell'icona per la modifica. La traccia del componente Slider è costituita da tre barre.
- 6 Fate clic sulla barra superiore per selezionarla. Quando è selezionata, il relativo colore viene visualizzato nel selettore Colore riempimento della finestra di ispezione Proprietà.
- 7 Mediante il selettore Colore riempimento della finestra di ispezione Proprietà, selezionate il colore #000066 per applicarlo alla barra superiore della traccia di Slider.
- 8 Fate clic sulla barra centrale della traccia di Slider per selezionarla. Quando è selezionata, il relativo colore viene visualizzato nel selettore Colore riempimento della finestra di ispezione Proprietà.
- 9 Mediante il selettore Colore riempimento della finestra di ispezione Proprietà, selezionate il colore #0066FF per applicarlo alla barra centrale della traccia di Slider.
- 10 Fate clic sulla barra inferiore della traccia di Slider per selezionarla. Quando è selezionata, il relativo colore viene visualizzato nel selettore Colore riempimento della finestra di ispezione Proprietà.
- 11 Mediante il selettore Colore riempimento della finestra di ispezione Proprietà, selezionate il colore #00CCFF per applicarlo alla barra inferiore della traccia di Slider.
- 12 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 13 Selezionare Controllo > Prova filmato.

Il componente Slider dovrebbe apparire come nella figura seguente.



Personalizzare il componente TextArea

Potete trasformare un componente TextArea modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi **Elabora > Trasforma**. In fase di runtime, usate il metodo `setSize()` o le proprietà valide della classe TextArea quali `height`, `width`, `scaleX` e `scaleY`.

Quando un componente TextArea viene ridimensionato, il bordo viene adattato al nuovo riquadro di delimitazione. Le barre di scorrimento, se necessarie, vengono posizionate sui bordi inferiore e destro. L'area di testo viene quindi ridimensionata in base all'area rimanente; in un componente TextArea non esistono elementi a dimensioni fisse. Se la larghezza del componente TextArea è troppo limitata per consentire la visualizzazione della larghezza del testo, il testo viene troncato.

Stili e componente TextArea

Gli stili del componente TextArea consentono di specificare i valori per gli skin, il riempimento e il formato del testo del componente quando viene disegnato. Gli stili `textFormat` e `disabledTextFormat` determinano lo stile del testo visualizzato nel componente TextArea. Per ulteriori informazioni sulle proprietà degli stili degli skin, vedete [“Usare gli skin con il componente TextArea”](#) a pagina 132.

Nell'esempio seguente viene impostato lo stile `disabledTextFormat` per modificare l'aspetto del testo quando `TextArea` è disattivato, ma lo stesso processo si applica per l'impostazione dello stile `textFormat` per un componente `TextArea` attivato.

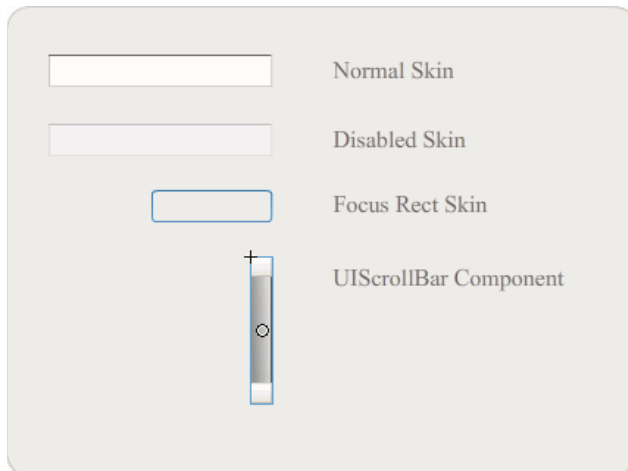
- 1 Create un nuovo file Flash.
- 2 Trascinate nello stage un componente `TextArea` e assegnategli il nome di istanza `myTa`.
- 3 Aggiungete il codice seguente nel pannello Azioni al fotogramma 1 della linea temporale principale.

```
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 Selezionare Controllo > Prova filmato.

Usare gli skin con il componente `TextArea`

Il componente `TextArea` usa i seguenti skin, che possono essere modificati per cambiarne l'aspetto.



Skin del componente `TextArea`

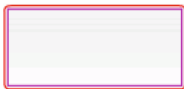
Nota: la modifica dello skin `ScrollBar` in un componente viene applicata a tutti gli altri componenti che usano `ScrollBar`.

Nella procedura seguente vengono modificati il colore del bordo dello skin `Focus Rect`, che viene visualizzato quando il componente `TextArea` è attivo, e dello skin `Normal`.

- 1 Create un nuovo file Flash.
- 2 Trascinate un componente `TextArea` nello stage e fate doppio clic sul componente per aprire il relativo pannello di icone di skin.
- 3 Fate doppio clic sullo skin `Focus Rect`.
- 4 Fate clic sul bordo dello skin `Focus Rect` per selezionarlo. Quando è selezionata, il relativo colore viene visualizzato nel selettore Colore riempimento della finestra di ispezione Proprietà.

- 5 Fate clic sul selettore Colore riempimento nella finestra di ispezione Proprietà per aprirlo e selezionare il colore #CC0000 per applicarlo al bordo.
- 6 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 7 Fate doppio clic sul componente TextArea per aprire il relativo pannello di icone di skin.
- 8 Fate doppio clic sullo skin Normal.
- 9 Selezionate ogni lato del bordo dello skin Up uno alla volta, e impostate il relativo colore su #990099.
- 10 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 11 Selezionare Controllo > Prova filmato.

Quando selezionate il componente TextArea per iniziare a inserire del testo, il bordo dovrebbe essere visualizzato come illustrato nella figura seguente:



Il bordo esterno è costituito dallo skin Focus Rect, quello interno dallo skin Normal.

Per informazioni sulla modifica dello skin del componente UIScrollBar, vedete [“Personalizzare il componente UIScrollBar”](#) a pagina 137.

Personalizzare il componente TextInput

Potete modificare le dimensioni di un'istanza TextInput sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi *Elabora > Trasforma*. In fase di runtime, usate il metodo `setSize()` o le proprietà valide della classe TextInput quali `height`, `width`, `scaleX` e `scaleY`.

Quando un componente TextInput viene ridimensionato, il bordo viene adattato al nuovo riquadro di delimitazione. Il componente TextInput non utilizza le barre di scorrimento ma il punto di inserimento scorre automaticamente quando l'utente interagisce con il testo. Il campo di testo viene quindi ridimensionato in base all'area rimanente; in un componente TextInput non esistono elementi a dimensioni fisse. Se il componente TextInput è troppo piccolo per consentire la visualizzazione del testo, il testo viene troncato.

Stili e componente TextInput

Gli stili del componente TextInput consentono di specificare i valori per gli skin, il riempimento e il formato del testo del componente quando viene disegnato. Gli stili `textFormat` e `disabledTextFormat` determinano lo stile del testo visualizzato nel componente. Per ulteriori informazioni sulle proprietà degli stili degli skin, vedete [“Skin e componente TextInput”](#) a pagina 134.

Nell'esempio seguente viene impostato lo stile `textFormat` per specificare il carattere, la dimensione e il colore del testo visualizzato nel componente TextInput. Lo stesso processo consente di impostare lo stile `disabledTextFormat` che viene applicato quando il componente è disattivato.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate nello stage un componente TextInput e assegnategli il nome di istanza **myTi**.

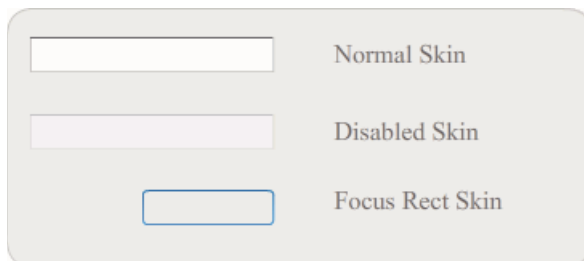
- 3 Aggiungete il codice seguente nel pannello Azioni al fotogramma 1 della linea temporale principale.

```
var tf:TextFormat = new TextFormat();  
tf.color = 0x0000FF;  
tf.font = "Verdana";  
tf.size = 30;  
tf.align = "center";  
tf.italic = true;  
myTi.setStyle("textFormat", tf);  
myTi.text = "Enter your text here";  
myTi.setSize(350, 50);  
myTi.move(100, 50);
```

- 4 Selezionare Controllo > Prova filmato.

Skin e componente TextInput

Il componente TextInput usa i seguenti skin, che possono essere modificati per cambiarne l'aspetto.



Didascalia del componente TextInput

La procedura riportata di seguito consente di modificare i colori del bordo e dello sfondo di un componente TextInput.

- 1 Create un nuovo file Flash.
- 2 Trascinate un componente TextInput nello stage e fate doppio clic sul componente per aprire il relativo pannello di skin.
- 3 Fate doppio clic sullo skin Normal.
- 4 Impostate la percentuale di ingrandimento su 800% per aumentare le dimensioni dell'icona per la modifica.
- 5 Selezionate ogni lato del bordo dello Normal, uno alla volta, e impostate il relativo colore su #993399 per applicarlo.
- 6 Fate doppio clic sullo sfondo per visualizzare il colore nel selettore Colore riempimento nella finestra di ispezione Proprietà. Selezionate il colore #99CCCC per applicarlo allo sfondo.
- 7 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 8 Selezionare Controllo > Prova filmato.

Il componente TextInput dovrebbe apparire come nella figura seguente:



Personalizzare il componente TileList

Potete trasformare un componente `TileList` modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi *Elabora > Trasforma*. In fase di runtime, usate il metodo `setSize()` o le proprietà valide quali `width`, `height`, `columnCount`, `rowCount`, `scaleX` e `scaleY`. Lo skin `ScrollBar`, contenuto nel componente `TileList`, viene ridimensionato con la casella di riepilogo.

Stili e componente TileList

Gli stili del componente `TileList` consentono di specificare i valori per gli skin, il riempimento e il formato del testo del componente quando viene disegnato. Gli stili `textFormat` e `disabledTextFormat` determinano lo stile del testo visualizzato nel componente. Per ulteriori informazioni sugli stili degli skin, vedete [“Usare gli skin con il componente TileList”](#) a pagina 135.

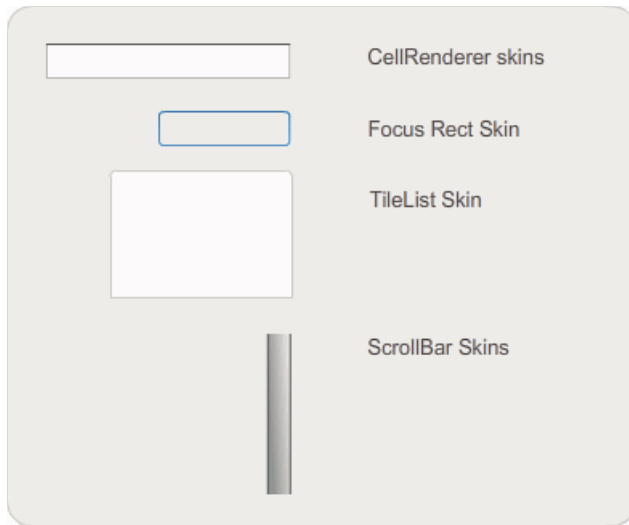
Nell'esempio seguente viene chiamato il metodo `setRendererStyle()` mediante lo stile `textFormat` per specificare il carattere, la dimensione e gli attributi del testo delle etichette visualizzate in un'istanza `TileList`. Lo stesso processo consente di impostare lo stile `disabledTextFormat` che viene applicato quando la proprietà `enabled` è impostata su `false`.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente `TileList` nello stage e assegnategli il nome di istanza **myTl**.
- 3 Aggiungere il codice seguente nel pannello Azioni al fotogramma 1 nella linea temporale.

```
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);
```

Usare gli skin con il componente TileList

Il componente `TileList` dispone degli skin `TileList`, `CellRenderer` e `ScrollBar`. Potete modificare questi skin per cambiare l'aspetto del componente `TileList`:



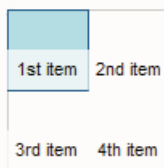
Skin del componente TileList

Nota: la modifica dello skin ScrollBar in un componente viene applicata a tutti gli altri componenti che usano il componente ScrollBar.

La procedura seguente consente di cambiare il colore dello skin CellRenderer Selected_Up del componente TileList.

- 1 Create un documento Flash (ActionScript 3.0).
- 2 Trascinate il componente TileList nello stage e fate doppio clic sul componente per aprire il relativo pannello di skin.
- 3 Fate doppio clic sullo skin CellRenderer, quindi fate doppio clic sullo skin Selected_Up e infine sullo sfondo rettangolare.
- 4 Selezionate il colore #99FFFF usando il selettore Colore riempimento nella finestra di ispezione Proprietà per applicarlo allo skin Selected_Up.
- 5 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 6 Nella scheda Parametri della finestra di ispezione Proprietà, fate doppio clic sulla seconda colonna della riga dataProvider per aprire la finestra di dialogo Valori. Aggiungete degli elementi con le etichette seguenti: 1st item, 2nd item, 3rd item, 4th item.
- 7 Selezionare Controllo > Prova filmato.
- 8 Fate clic su una cella nel componente TileList per selezionarla, quindi allontanate il puntatore del mouse dalla cella selezionata.

La cella selezionata dovrebbe essere visualizzata come nella figura seguente:



Componente TileList con colore skin Selected_Up modificato

Personalizzare il componente UI Loader

Potete trasformare un componente UI Loader modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi **Elabora > Trasforma**. In fase di runtime, usate il metodo `setSize()` o le proprietà valide quali `width`, `height`, `scaleX` e `scaleY`.

Il comportamento di ridimensionamento del componente UI Loader viene controllato dalla proprietà `scaleContent`. Se `scaleContent` è impostato su `true`, il contenuto viene modificato in scala per essere adattato ai limiti del loader e viene di nuovo modificato in scala quando si chiama `setSize()`. Se la proprietà `scaleContent` è impostata su `false`, la dimensione del componente è fissa in base alla dimensione del contenuto e il metodo `setSize()` e le proprietà di ridimensionamento non hanno effetto.

Il componente UI Loader non dispone di elementi dell'interfaccia utente a cui potete applicare stili o skin.

Personalizzare il componente UI ScrollBar

Potete trasformare un componente UI ScrollBar modificandone le dimensioni orizzontale e verticale sia durante la creazione che in fase di runtime. Tuttavia, un componente UI ScrollBar verticale non consente di modificare la larghezza, mentre un componente UI ScrollBar orizzontale non permette di modificare l'altezza. Durante la creazione, selezionate il componente nello stage e usate lo strumento Trasformazione libera o uno dei comandi **Elabora > Trasforma**. In fase di runtime, usate il metodo `setSize()` o le proprietà valide della classe UI ScrollBar quali `width`, `height`, `scaleX` e `scaleY`.

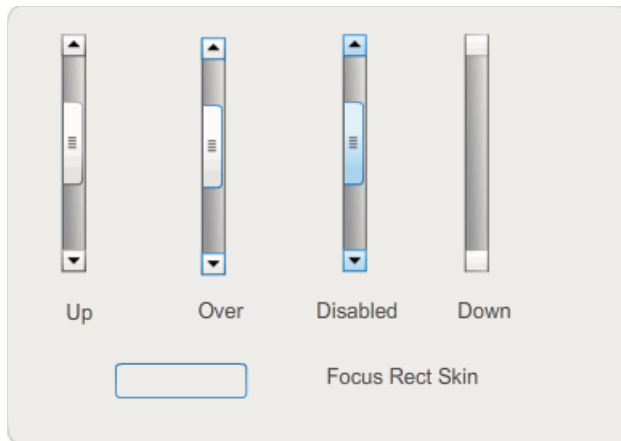
***Nota:** se utilizzate il metodo `setSize()`, potete modificare solo la larghezza di una barra di scorrimento orizzontale o l'altezza di una barra di scorrimento verticale. In fase di creazione potete impostare l'altezza di una barra di scorrimento orizzontale o la larghezza di una barra di scorrimento verticale, ma i valori vengono ripristinati al momento della pubblicazione del filmato. È possibile modificare solo le dimensioni di una barra di scorrimento corrispondente alla sua lunghezza.*

Usare gli stili con il componente UI ScrollBar

Gli stili del componente UI ScrollBar consentono di specificare solo le classi per i relativi skin e un valore per `FocusRectPadding`, che specifica il numero di pixel da usare per il riempimento tra il riquadro di delimitazione e il bordo esterno del componente. Per ulteriori informazioni sull'uso degli stili degli skin, vedete [“Informazioni sugli skin”](#) a pagina 105.

Usare gli skin con il componente UI ScrollBar

Il componente UI ScrollBar usa i seguenti skin:



Skin del componente UI ScrollBar

Le barre orizzontale e verticale usano gli stessi skin, che vengono ruotati automaticamente dal componente UI ScrollBar quando viene visualizzata una barra di scorrimento orizzontale.

Nota: la modifica dello skin ScrollBar in un componente viene applicata a tutti gli altri componenti che usano il componente ScrollBar.

L'esempio seguente illustra come cambiare il colore della casella di scorrimento e dei pulsanti freccia del componente UI ScrollThumb.

- 1 Create un nuovo documento Flash (ActionScript 3.0).
- 2 Trascinate il componente UI ScrollBar nello stage e assegnategli il nome di istanza **mySb**. Nella scheda Parametri, impostate la direzione su orizzontale.
- 3 Fate doppio clic sulla barra di scorrimento per aprire il relativo pannello di skin.
- 4 Fate clic sullo skin Up per selezionarlo.
- 5 Impostate la percentuale di ingrandimento su 400% per aumentare le dimensioni dell'icona per la modifica.
- 6 Fate doppio clic sullo sfondo del pulsante freccia destra (o freccia su per una barra di scorrimento verticale) per selezionarlo; il relativo colore viene visualizzato nel selettore Colore riempimento della finestra di ispezione Proprietà.
- 7 Selezionate il colore #CC0033 per applicarlo allo sfondo del pulsante.
- 8 Fate clic sul pulsante Indietro sul lato sinistro della barra di modifica, visualizzata sopra lo stage, per tornare alla modalità di modifica del documento.
- 9 Ripetete i punti 6, 7 e 8 per gli elementi casella di scorrimento e freccia sinistra (o freccia giù per una barra di scorrimento verticale).
- 10 Aggiungete il codice seguente nel pannello Azioni al fotogramma 1 della linea temporale per associare la barra di scorrimento a un componente TextField.

```
var tf:TextField = new TextField();
addChild(tf);
tf.x = 150;
tf.y = 100;
mySb.width = tf.width = 200;
tf.height = 22;
tf.text = "All work and no play makes Jack a dull boy. All work and no play makes Jack a
dull boy. All . . .";
mySb.y = tf.y + tf.height;
mySb.x = tf.x + tf.width;x
mySb.scrollTarget = tf;
```

11 Selezionare Controllo > Prova filmato.

Il componente UIScrollBar dovrebbe apparire come nella figura seguente.



ScrollBar orizzontale con casella di scorrimento e frecce sinistra e destra in rosso

Capitolo 6: Uso del componente FLVPlayback

Il componente FLVPlayback consente di includere con facilità un lettore video in un'applicazione CS5 Professional per la riproduzione di file a scaricamento progressivo via HTTP o per la riproduzione di file in streaming da Macromedia Flash Media Server di Adobe o da Flash Video Streaming Service (FVSS).

Con il rilascio di Adobe Flash Player 9 Update 3 (versione 9.0.115.0 o successiva) sono stati incorporati dei miglioramenti significativi alla riproduzione dei contenuti video in Flash Player. L'aggiornamento comprende delle modifiche al FLVPlayback che sfruttano l'hardware video del sistema dell'utente finale per fornire una migliore riproduzione video. Le modifiche apportate al componente FLVPlayback aumentano tra l'altro la fedeltà dei file video riprodotti in modalità a schermo intero.

Inoltre, Flash Player 9 Update 3 migliora la funzionalità del componente FLVPlayback grazie all'aggiunta del supporto per i formati video MPEG-4 in alta definizione che utilizzano la codifica H.264 standard di settore, tra cui: MP4, M4A, MOV, MP4V, 3GP e 3G2.

***Nota:** i file MP4 protetti (ad esempio, quelli scaricati da Apple® iTunes® o con crittografia digitale FairPlay®) non sono supportati.*

Il componente FLVPlayback è di facile utilizzo e presenta le caratteristiche e i vantaggi seguenti:

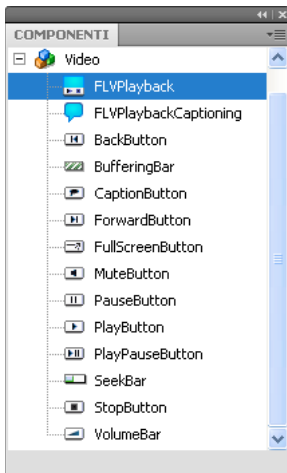
- Può essere trascinato nello stage e implementato in modo rapido ed efficace
- Supporta la dimensione a schermo intero
- Offre una raccolta di *skin* predefiniti che consentono di personalizzare l'aspetto dei controlli di riproduzione
- Consente di selezionare i valori di colore e alfa per gli *skin* predefiniti
- Consente agli utenti più esperti di creare i propri *skin*
- Fornisce l'anteprima dal vivo durante la creazione
- Fornisce proprietà del layout per mantenere centrato il file video durante il ridimensionamento
- Consente di iniziare la riproduzione dopo che è stata scaricata una quantità sufficiente di un file video a scaricamento progressivo
- Fornisce cue point per permettono di sincronizzare il video con testo, grafica e animazioni
- Consente di ottenere file SWF di dimensioni relativamente ridotte

Usare il componente FLVPlayback

Per utilizzare il componente FLVPlayback è sufficiente posizionarlo sullo stage e specificare il file video per avviare la riproduzione. Inoltre, potete impostare vari parametri per regolarne il comportamento e per descrivere il file video.

Il componente FLVPlayback include anche un'API (Application Programming Interface) ActionScript. L'API include le classi seguenti, che sono completamente descritte nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#): CuePointType, FLVPlayback, FLVPlaybackCaptioning, NCManager, NCManagerNative, VideoAlign, VideoError, VideoPlayer, VideoState, and several event classes - AutoLayoutEvent, LayoutEvent, MetadataEvent, SkinErrorEvent, SoundEvent, VideoEvent e VideoProgressEvent.

Il componente FLVPlayback include i componenti UI FLV di riproduzione personalizzati (FLV Playback Custom UI) ed è una combinazione dell'area di visualizzazione, o lettore video, nella quale viene visualizzato il file video e dei controlli che ne consentono il funzionamento. I componenti UI FLV di riproduzione personalizzati (FLV Playback Custom UI) dispongono di pulsanti e meccanismi di controllo che permettono di riprodurre, interrompere, mettere in pausa e controllare in altro modo il file video. Questi controlli includono i componenti BackButton, BufferingBar, CaptionButton (per FLVPlaybackCaptioning), ForwardButton, FullScreenButton, MuteButton, PauseButton, PlayButton, PlayPauseButton, SeekBar, StopButton e VolumeBar. Il componente FLVPlayback e i controlli UI FLV di riproduzione personalizzati sono disponibili nel pannello Componenti, come illustrato nella figura seguente:



Componenti FLVPlayback nel pannello Componenti

La possibilità di aggiungere i controlli della riproduzione al componente FLVPlayback viene detta *associazione di skin*. Il componente FLVPlayback presenta lo skin iniziale predefinito SkinOverAll.swf, che fornisce i controlli per le funzioni di riproduzione, interruzione, riavvolgimento, avanzamento, ricerca, volume disattivato, volume attivato, schermo intero e inserimento didascalie. Per modificare questo skin, potete:

- Selezionare da una raccolta di skin predefiniti
- Creare uno skin personalizzato e aggiungerlo alla raccolta di skin predefiniti
- Selezionare e personalizzare i singoli controlli dei componenti UI FLV di riproduzione personalizzati

Quando selezionate uno skin predefinito, potete scegliere separatamente i valori alfa e di colore dello skin, sia in fase di creazione che di runtime. Per ulteriori informazioni, vedete [“Selezionare uno skin predefinito”](#) a pagina 159.

Una volta selezionato lo skin, questo diventa il nuovo skin predefinito.

Per ulteriori informazioni sulla selezione e la creazione di uno skin per il componente FLVPlayback, vedete [“Personalizzare il componente FLVPlayback”](#) a pagina 159.

Creare un'applicazione con il componente FLVPlayback

Il componente FLVPlayback può essere inserito in un'applicazione nei seguenti modi:

- Trascinando il componente FLVPlayback dal pannello Componenti nello stage e specificando un valore per il parametro `source`.
- Usando la procedura guidata Importa video per creare il componente sullo stage e personalizzarlo con uno skin.
- Usando la funzione di costruzione `FLVPlayback()` per creare un'istanza FLVPlayback sullo stage in modo dinamico, supponendo che il componente sia nella libreria.

Uso del componente FLVPlayback

***Nota:** se create un'istanza FLVPlayback con ActionScript, dovete anche assegnargli uno skin impostando la proprietà `skin` con ActionScript. Uno skin applicato in questo modo non viene pubblicato automaticamente con il file SWF. Dovete copiare sia il file SWF dell'applicazione che il file SWF dello skin sul server di applicazioni, altrimenti il file SWF dello skin non sarà disponibile quando si esegue l'applicazione.*

Trascinare il componente FLVPlayback dal pannello Componenti

- 1 Nel pannello Componenti, fate clic sul segno più (+) per aprire la voce video.
- 2 Trascinate il componente FLVPlayback nello stage.
- 3 Nella scheda Parametri della finestra di ispezione dei componenti, con il componente FLVPlayback selezionato nello stage, individuate la cella Valore corrispondente al parametro `source` e inserite una stringa che specifichi uno dei dati seguenti:

- Un percorso locale a un file video
- Un URL a un file video
- Un URL a un file SMIL (Synchronized Multimedia Integration Language) che descrive come riprodurre un file video.

Per informazioni su come creare un file SMIL utilizzato per descrivere uno o più file FLV, vedete “[Usare un file SMIL](#)” a pagina 170.

- 4 Nella scheda Parametri della finestra di ispezione dei componenti, con il componente FLVPlayback selezionato nello stage, fate clic sulla cella Valore del parametro `skin`.
- 5 Fate clic sull'icona della lente di ingrandimento per aprire la finestra di dialogo Seleziona skin.
- 6 Selezionate una delle seguenti opzioni:

- Dall'elenco a discesa Skin, selezionate uno degli skin predefiniti per associare al componente una serie di controlli di riproduzione.
- Se è stato creato uno skin personalizzato, selezionate URL skin personalizzato dal menu a comparsa, quindi immettete l'URL del file SWF che contiene lo skin nella casella URL.
- Per aggiungere i controlli di riproduzione, selezionate Nessuno e trascinate i singoli componenti UI FLV di riproduzione personalizzati nello stage.

***Nota:** in entrambi i casi, nel riquadro di visualizzazione sopra il menu a comparsa viene visualizzata un'anteprima dello skin. Potete utilizzare il selettore di colore per cambiare il colore dello skin.*

Per cambiare il colore di un controllo UI personalizzato, dovete personalizzarlo. Per ulteriori informazioni sull'uso dei controlli UI personalizzati, vedete “[Associare skin ai singoli componenti UI FLV di riproduzione personalizzati](#)” a pagina 160.

- 7 Fate clic su OK per chiudere la finestra di dialogo Seleziona skin.
- 8 Selezionate Controllo > Prova filmato per eseguire il file SWF e avviare il video.
Nella procedura riportata di seguito viene usata la procedura guidata Importa video per aggiungere un componente FLVPlayback:

Usare la procedura guidata Importa video:

- 1 Selezionate File > Importa > Importa video.
- 2 Indicate la posizione del file video selezionando una delle seguenti opzioni:
 - Sul computer locale
 - Già distribuito su un server Web, Flash Video Streaming Service o Flash Media Server

- 3 A seconda della selezione effettuata, immettete il percorso del file o l'URL che specifica la posizione del file video, quindi fate clic su Avanti.
- 4 Se avete selezionato il percorso del file, viene visualizzata la finestra di dialogo Distribuzione in cui potete selezionare una delle opzioni elencate per specificare le modalità di distribuzione del video:
 - Scaricamento progressivo da un server Web standard
 - In streaming da Flash Video Streaming Service
 - In streaming da Flash Media Server
 - Incorpora video in SWF e riproduci nella linea temporale

***Importante:** non selezionate l'opzione Incorpora Video. Il componente FLVPlayback consente di riprodurre solo video in streaming. Questa opzione non consente di posizionare il componente FLVPlayback nello stage.*
- 5 Fate clic su Next (Avanti).
- 6 Selezionate una delle seguenti opzioni:
 - Dall'elenco a discesa Skin, selezionate uno degli skin predefiniti per associare al componente una serie di controlli di riproduzione.
 - Se è stato creato uno skin personalizzato per il componente, selezionate URL skin personalizzato dal menu a comparsa, quindi immettete l'URL del file SWF che contiene lo skin nella casella URL.
 - Per aggiungere i controlli di riproduzione, selezionate Nessuno e trascinate i singoli componenti UI FLV di riproduzione personalizzati nello stage.

***Nota:** in entrambi i casi, nel riquadro di visualizzazione sopra il menu a comparsa viene visualizzata un'anteprima dello skin.*
- 7 Fate clic su OK per chiudere la finestra di dialogo Seleziona skin.
- 8 Leggete il contenuto della finestra di dialogo Esegui importazione video per informazioni sui passaggi successivi, quindi fate clic su Fine.
- 9 Se il file FLA non è ancora stato salvato, viene visualizzata la finestra di dialogo Salva con nome.
- 10 Selezionate Controllo > Prova filmato per eseguire il SWF e avviare il video.

Nella procedura seguente viene aggiunto il componente FLVPlayback mediante ActionScript.

Creare un'istanza in modo dinamico mediante ActionScript:

- 1 Trascinate il componente FLVPlayback dal pannello Componenti al pannello Libreria (Finestra > Libreria).
- 2 Aggiungete il codice seguente nel pannello Azioni al fotogramma 1 della linea temporale. Sostituite *install_drive* con l'unità in cui avete installato Flash e modificate il percorso in modo che rifletta la posizione della cartella Skins nell'installazione locale:

Su un computer 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";
```

Su un computer Macintosh:

Uso del componente FLVPlayback

```
import fl.video.*;
var my_FLVPlaybk = new FLVPlayback();
my_FLVPlaybk.x = 100;
my_FLVPlaybk.y = 100;
addChild(my_FLVPlaybk);
my_FLVPlaybk.skin = "file:///Macintosh HD:Applications:Adobe Flash
CS5:Configuration:FLVPlayback Skins:ActionScript 3.0SkinOverPlaySeekMute.swf"
my_FLVPlaybk.source = "http://www.helpexamples.com/flash/video/water.flv";
```

Nota: se non impostate le proprietà *source* e *skin*, il clip filmato generato apparirà vuoto.

3 Selezionate Controllo > Prova filmato per eseguire il file SWF e avviare il file video.

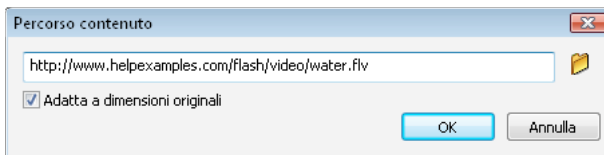
Parametri del componente FLVPlayback

Per ogni istanza del componente FLVPlayback potete impostare i seguenti parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti: *align*, *autoPlay*, *cuePoints*, *preview*, *scaleMode*, *skin*, *skinAutoHide*, *skinBackgroundAlpha*, *skinBackgroundColor*, *source* e *volume*. Ciascuno di questi parametri ha una proprietà ActionScript corrispondente con lo stesso nome. Quando assegnate un valore a questi parametri, impostate lo stato iniziale della proprietà nell'applicazione. Se impostate la proprietà in ActionScript, il valore impostato nel parametro viene ignorato. Per informazioni sui valori che potete impostare per questi parametri, vedete la classe FLVPlayback nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Specificare il parametro source di FLVPlayback

Il parametro *source* consente di specificare il nome e il percorso del file video per indicare a Flash come riprodurre il file.

Aprire la finestra di dialogo Percorso contenuto facendo doppio clic sulla cella Valore per attivare il parametro *source* nella finestra di ispezione dei componenti.



Finestra di dialogo Percorso contenuto del componente FLVPlayback

La finestra di dialogo Percorso contenuto comprende la casella di controllo Adatta a dimensioni FLV originali, che specifica se l'istanza di FLVPlayback nello stage corrisponde alle dimensioni del file video di origine. Il file video di origine contiene le dimensioni relative alla larghezza e all'altezza desiderate per la riproduzione. Se scegliete questa opzione, l'istanza FLVPlayback viene ridimensionata in base alle misure desiderate.

Il parametro source

Immettete l'URL o il percorso locale di un file video oppure di un file XML che descriva come riprodurre un file video. Se non conoscete il percorso esatto di un file video, fate clic sull'icona della cartella per aprire una finestra di dialogo Sfoglia, mediante la quale potete individuare il percorso corretto. Quando utilizzate la finestra di dialogo Sfoglia per individuare un file video, se questo si trova nello stesso percorso o a un livello inferiore rispetto al file SWF di destinazione, Flash rende il percorso automaticamente relativo al percorso indicato, in modo che il file possa essere distribuito da un server Web. In caso contrario, si tratta di un percorso di file assoluto di Windows o Macintosh. Per inserire il nome di un file XML locale dovete digitare il percorso e il nome del file.

Se specificate un URL HTTP, il file FLV è a scaricamento progressivo. Se specificate un URL di tipo RMTP, il file video viene distribuito in streaming da Flash Media Server o da FVSS. Anche l'URL di un file XML potrebbe essere un file video distribuito in streaming da un server Flash Media Server o da FVSS.

Importante:

Potete inoltre specificare la posizione di un file SMIL che descrive come riprodurre flussi di file video multipli per larghezze di banda differenti. Il file utilizza il linguaggio SMIL (Synchronized Multimedia Integration Language) per descrivere i file FLV. Per una descrizione del file SMIL, vedete “[Usare un file SMIL](#)” a pagina 170.

Potete anche specificare il nome e la posizione del file video mediante il componente FLVPlayback di ActionScript,utilizzandone la proprietà `source` e i metodi `FLVPlayback.play()` e `FLVPlayback.load()`. Queste tre alternative hanno la precedenza sul parametro `source` nella finestra di ispezione dei componenti. Per ulteriori informazioni, vedete le voci `FLVPlayback.source`, `FLVPlayback.play()` e `FLVPlayback.load()` per la classe FLVPlayback nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Supporto della modalità a schermo intero

La versione ActionScript 3.0 del componente FLVPlayback supporta la modalità a schermo intero, che richiede Flash Player 9.0.28.0 o versione successiva e che il codice sia impostato correttamente per la visualizzazione a schermo intero. Alcuni skin predefiniti includono un pulsante di tipo premuto/non premuto per attivare e disattivare la modalità a schermo intero. L'icona di FullScreenButton viene visualizzata sul lato destro della barra di controllo come illustrato nella figura seguente.



Icona schermo intero nella barra di controllo

Il supporto della modalità a schermo intero è presente solo se la proprietà `fullScreenTakeOver` è impostata su `true` (impostazione predefinita).

Il supporto della modalità a schermo intero è utilizzabile con o senza il supporto dell'accelerazione hardware. Per informazioni sul supporto dell'accelerazione hardware, vedete “[Accelerazione hardware](#)” a pagina 148.

Per implementare il supporto della modalità a schermo intero per FLVPlayback:

- 1 Aggiungete il componente FLVPlayback all'applicazione e assegnategli un file video.
- 2 Selezionate uno skin per il componente FLVPlayback che includa il pulsante per lo schermo intero (ad esempio, `SkinUnderPlaySeekFullscreen.swf`) o aggiungete il componente `FullScreenButton` dell'interfaccia utente al componente FLVPlayback dalla sezione Video del pannello dei componenti.
- 3 Selezionate File > Impostazioni pubblicazione.
- 4 Nella finestra di dialogo Impostazioni pubblicazione, fate clic sulla scheda HTML.
- 5 Nella scheda HTML, selezionate Flash con supporto schermo intero dal menu a comparsa Modello.
- 6 Sempre nella scheda HTML, selezionate la casella di controllo Rileva versione di Flash e specificate la versione 9.0.28 o una versione successiva, a seconda della versione di Flash Player in uso.
- 7 Selezionate la scheda Formati e assicuratevi che le opzioni Flash (.swf) e HTML (.html) siano entrambe selezionate. Potete sostituire i nomi di file predefiniti.
- 8 Fate clic su Pubblica, quindi su OK.

In alternativa al punto 7, potete fare clic su OK e selezionare File > Anteprima pubblicazione > Predefinito - (HTML) per aprire automaticamente nel browser predefinito il file HTML esportato. Altrimenti, aprite il file HTML esportato con il browser per provare l'opzione schermo intero.

Per aggiungere il componente FLVPlayback con supporto della modalità a schermo intero alla pagina Web, aprite il file HTML esportato e copiate il codice che incorpora il file SWF nel file HTML della pagina Web. Il codice dovrebbe essere simile a quello riportato nell'esempio seguente:

```
//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>
```

In alternativa, potete usare il file HTML esportato come modello per la pagina Web e aggiungervi altro contenuto. In tal caso, dovete tuttavia cambiare il nome del file HTML per evitare di sovrascriverlo accidentalmente nel caso decidiate in un secondo momento di esportare nuovamente il file HTML di FLVPlayback da Flash.

In ogni caso, dovete anche caricare sul server Web il file AC_RunActiveContent.js che viene esportato nella stessa cartella del file HTML.

Il supporto ActionScript della modalità a schermo intero include le proprietà `fullScreenBackgroundColor`, `fullScreenSkinDelay` e `fullScreenTakeOver` e il metodo `enterFullScreenDisplayState()`. Per informazioni su questi elementi ActionScript, vedete la [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Usare `enterFullScreenDisplayState()`

Potete richiamare la modalità a schermo intero anche chiamando il metodo `enterFullScreenDisplayState()` di ActionScript, come mostrato nell'esempio seguente.

```
function handleClick(e:MouseEvent):void {  
    myFLVPlybk.enterFullScreenDisplayState();  
}  
myButton.addEventListener(MouseEvent.CLICK, handleClick);
```

Nell'esempio, la modalità a schermo intero *non* viene richiamata facendo clic sul pulsante di attivazione/disattivazione della modalità dello skin FLVPlayback, ma facendo clic su un pulsante (MyButton) che il creatore della pagina Web ha incluso per richiamare la modalità a schermo intero. Facendo clic sul pulsante si attiva il gestore di eventi `handleClick`, che chiama il metodo `enterFullScreenDisplayState()`.

Il metodo `enterFullScreenDisplayState()` imposta la proprietà `Stage.displayState` su `StageDisplayState.FULL_SCREEN` e pertanto ha le stesse limitazioni della proprietà `displayState`. Per ulteriori informazioni sul metodo `enterFullScreenDisplayState()` e sulla proprietà `Stage.displayState`, vedete la [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Accelerazione hardware

Flash Player versione 9.0.115.0 e successive include del codice che sfrutta l'hardware video disponibile per migliorare le prestazioni e la fedeltà dei file FLV riprodotti da FLVPlayback in modalità a schermo intero. Se i prerequisiti sono soddisfatti e la proprietà `fullScreenTakeOver` è impostata su `true`, Flash Player si serve dell'accelerazione hardware per modificare in scala il file video anziché utilizzare il software. Se il componente FLVPlayback viene eseguito in una versione precedente di Flash Player o se non sussistono i prerequisiti per l'accelerazione hardware, è Flash Player stesso a modificare in scala il file video, come accadeva in precedenza.

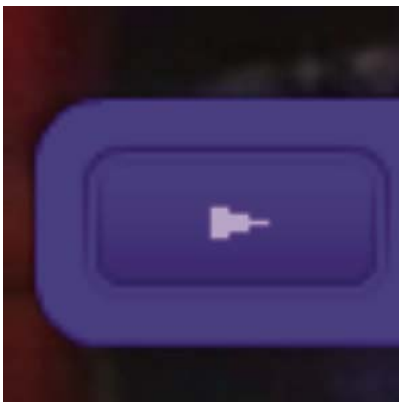
Per sfruttare l'accelerazione hardware e supportare la modalità a schermo intero, il computer deve essere dotato di una scheda video compatibile con DirectX 7 con almeno 4 MB di memoria VRAM (video RAM). Tale supporto hardware è disponibile in Windows 2000 o Mac OS X 10.2 e nelle versioni successive di questi sistemi operativi. Direct X® fornisce delle API che comprendono un'interfaccia tra il software e l'hardware video che, tra le altre cose, accelera la grafica tridimensionale e bidimensionale.

Per sfruttare la modalità di accelerazione hardware, dovete richiamare la modalità a schermo intero in uno dei modi seguenti:

- Mediante il pulsante di attivazione/disattivazione della modalità a schermo intero di uno skin di FLVPlayback
- Mediante il controllo video `FullScreenButton`
- Mediante il metodo `enterFullScreenDisplayState()` di ActionScript. Per ulteriori informazioni, vedete “[Usare enterFullScreenDisplayState\(\)](#)” a pagina 147.

Se richiamate la modalità a schermo intero impostando la proprietà `Stage.displayState` su `StageDisplayState.FULLSCREEN`, FLVPlayback non utilizza l'accelerazione hardware, anche se sono disponibili l'hardware e la memoria video.

Una conseguenza dell'uso dell'accelerazione hardware per il supporto della modalità a schermo intero consiste nel fatto che gli skin di FLVPlayback vengono modificati in scala insieme al lettore video e al file video. L'immagine seguente mostra l'effetto della modalità a schermo intero con l'accelerazione hardware sullo skin di FLVPlayback, un dettaglio del quale viene riprodotto con risoluzione piena.



Modalità a schermo intero su un monitor 1600 x 1200 con video di 320 x 240 pixel

L'immagine mostra il risultato dell'uso della modalità a schermo intero su un monitor 1600 x 1200 con un file video con dimensioni 320 x 240 (dimensioni predefinite di FLVPlayback). L'effetto di distorsione dello skin è più pronunciato nei file FLV con dimensioni ridotte o su un monitor più grande. Al contrario, risulta meno pronunciato nei file FLV di dimensioni maggiori o sui monitor più piccoli. Ad esempio, se modificate la dimensione da 640 x 480 a 1600 x 1200, la dimensione dello skin aumenta comunque ma appare meno distorta.

Potete impostare la proprietà `skinScaleMaximum` per limitare la modifica in scala dello skin di FLVPlayback. Il valore predefinito è 4.0, o 400 %. La limitazione della modifica in scala dello skin richiede tuttavia una combinazione di hardware e software per modificare il file FLV e ciò può influenzare negativamente le prestazioni con i file FLV di grandi dimensioni e codificati con una velocità di trasferimento elevata. Se il video è grande (ad esempio, almeno 640 pixel di larghezza, almeno 480 pixel di altezza), non dovete impostare la proprietà `skinScaleMaximum` su un valore basso perché su monitor grandi potrebbero verificarsi seri problemi di prestazione. La proprietà `skinScaleMaximum` consente di gestire il compromesso tra prestazioni e qualità e l'aspetto di uno skin di grandi dimensioni.

Uscire dalla modalità a schermo intero

Per uscire dalla modalità a schermo intero, fate nuovamente clic sul pulsante della modalità a schermo intero o premete il tasto Esc.

Se impostate le proprietà seguenti e chiamate i metodi seguenti, vengono apportate delle modifiche al layout che forzano il componente FLVPlayback a uscire dalla modalità a schermo intero: `height`, `registrationHeight`, `registrationWidth`, `registrationX`, `registrationY`, `scaleX`, `scaleY`, `width`, `x`, `y`, `setScale()` `setSize()`.

Se impostate la proprietà `align` o `scaleMode`, FLVPlayback le imposta su `center` e `maintainAspectRatio` fino a quando non si esce dalla modalità a schermo intero.

Se modificate il valore della proprietà `fullScreenTakeOver` da `true` a `false` mentre usate la modalità a schermo intero, anche la modalità di accelerazione hardware forza Flash a uscire dalla modalità a schermo intero.

Allineamento del layout per riprodurre più file video

Il componente ActionScript 3.0 FLVPlayback include una proprietà `align` che consente di specificare se il file video dovrà essere centrato quando viene ridimensionato o posizionato nella parte superiore, inferiore, sinistra o destra del componente. Oltre alle proprietà `x`, `y`, `width` e `height` del componente, il componente ActionScript 3.0 dispone anche delle proprietà `registrationX`, `registrationY`, `registrationWidth` e `registrationHeight` che inizialmente corrispondono alle proprietà `x`, `y`, `width` e `height`. Quando caricate i successivi file video, queste proprietà non vengono modificate mediante la modifica automatica del layout, quindi è possibile che i nuovi file video siano centrati nella stessa posizione. Se `scaleMode = VideoScaleMode.MAINTAIN_ASPECT_RATIO`, i successivi file FLV possono adattarsi alle dimensioni originali del componente, anziché causare la modifica della relativa larghezza e altezza.

Riproduzione automatica dei file video a scaricamento progressivo

Quando caricate un file video a scaricamento progressivo, FLVPlayback ne avvia la riproduzione solo quando è stata scaricata una quantità sufficiente del file per consentirne la riproduzione dall'inizio alla fine.

Se desiderate riprodurre il file video prima che ne sia stata scaricata una quantità sufficiente, chiamate il metodo `play()` senza parametri.

Se desiderate tornare allo stato di attesa (lo stato in cui la riproduzione ha inizio solo dopo che è stata scaricata una quantità sufficiente del file video), chiamate il metodo `pause()`, quindi il metodo `playWhenEnoughDownloaded()`.

Usare i cue point

Il cue point è un punto in corrispondenza del quale il lettore video invia un evento `cuePoint` durante la riproduzione di un file video. Potete aggiungere cue point a un file FLV quando desiderate eseguire un'azione su un altro elemento della pagina Web. Inoltre, potete visualizzare un testo o un'immagine grafica oppure sincronizzare un'animazione Flash o ancora influire sulla riproduzione del file FLV mettendolo in pausa, cercando un altro punto del video o passando a un altro file FLV. I cue point consentono di controllare il codice ActionScript per sincronizzare i punti del file FLV con altre azioni sulla pagina Web.

Ci sono tre tipi diversi di cue point: Navigazione, Evento e ActionScript. I cue point di tipo Navigazione ed Evento sono conosciuti anche come cue point *incorporati* perché sono incorporati nel FLV in streaming e nel pacchetto di metadati del file FLV.

Un *cue point di navigazione* consente di cercare un fotogramma particolare del file FLV perché crea un fotogramma chiave all'interno del file FLV, che sarà il più vicino possibile alla posizione temporale specificata. Un *fotogramma chiave* è un segmento di dati che si trova tra i fotogrammi immagine del file FLV in streaming. Quando eseguite la ricerca di un cue point di navigazione, il componente cerca il fotogramma chiave e avvia l'evento `cuePoint`.

Un *cue point di evento* consente di sincronizzare un punto temporale all'interno del file FLV con un evento esterno sulla pagina Web. L'evento `cuePoint` si verifica esattamente nel momento specificato. Potete incorporare i cue point di navigazione e di evento in un file FLV usando la procedura guidata Importa video oppure Flash Video Encoder. Per ulteriori informazioni sulla procedura guidata Importa video e su Flash Video Encoder, vedete il Capitolo 16 "Operazioni con i file video" nella guida *Uso di Flash*.

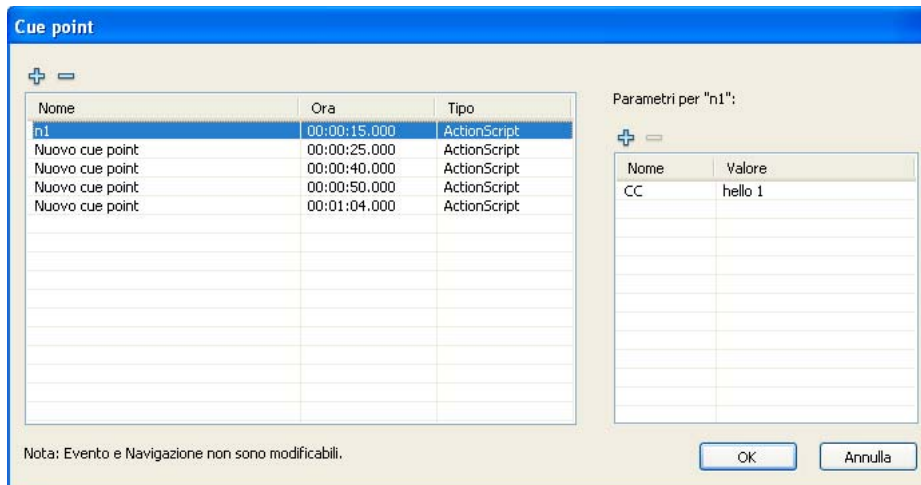
Un *cue point ActionScript* è un cue point esterno che potete aggiungere mediante la finestra di dialogo Cue point Flash Video oppure tramite il metodo `FLVPlayback.addASCuePoint()`. Il componente memorizza e tiene traccia dei cue point ActionScript in maniera indipendente dal file FLV, pertanto questi risultano meno precisi dei cue point incorporati. La valutazione della precisione dei cue point ActionScript è di un decimo di secondo. Potete migliorare la precisione dei cue point ActionScript abbassando il valore della proprietà `playheadUpdateInterval`, perché quando l'indicatore di riproduzione si aggiorna, il componente genera l'evento `cuePoint` per i cue point ActionScript. Per ulteriori informazioni, vedete la proprietà `FLVPlayback.playheadUpdateInterval` in [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

In ActionScript e all'interno dei metadati del file FLV, un cue point è rappresentato sotto forma di un oggetto con le proprietà seguenti: `name`, `time`, `type` e `parameters`. La proprietà `name` è una stringa che contiene il nome assegnato al cue point. La proprietà `time` è un numero che rappresenta l'ora in cui si verifica il cue point, espressa in ore, minuti, secondi e millisecondi (HH:MM:SS.mmm). La proprietà `type` è una stringa il cui valore può essere "navigation", "event" o "actionscript", a seconda del tipo di cue point che è stato creato. La proprietà `parameters` è un array di coppie nome/valore specificate.

Quando si verifica un evento `cuePoint`, l'oggetto cue point è disponibile nell'oggetto evento mediante la proprietà `info`.

Usare la finestra di dialogo Cue point Flash Video

Aprirete la finestra di dialogo Cue point Flash Video facendo doppio clic sulla cella Valore per attivare il parametro `cuePoints` nella finestra di ispezione dei componenti. La finestra di dialogo sarà simile alla seguente figura:



Finestra di dialogo Cue point

Nella finestra di dialogo vengono visualizzati i cue point ActionScript e incorporati. Potete utilizzare questa finestra di dialogo per aggiungere ed eliminare i cue point ActionScript e i parametri cue point. Inoltre, potete attivare o disattivare i cue point incorporati. Tuttavia, non potete aggiungere, modificare o eliminare i cue point incorporati.

Aggiungere un cue point ActionScript:

- 1 Fate doppio clic sulla cella Valore del parametro `cuePoints` nella finestra di ispezione dei componenti per aprire la finestra di dialogo Cue point Flash.
- 2 Fate clic sul segno più (+) nell'angolo in alto a sinistra, sopra l'elenco dei cue point, per aggiungere una voce cue point ActionScript predefinita.
- 3 Fate clic sul campo di testo Nuovo Cue Point nella colonna Nome e modificate il nome del cue point.
- 4 Fate clic sul valore Tempo 00:00:00:000 per modificarlo e assegnare la posizione temporale in cui desiderate che si verifichi il cue point. Potete specificare il tempo in ore, minuti, secondi e millisecondi (HH:MM:SS.mmm).

In caso di cue point multipli, il nuovo cue point viene spostato dalla finestra di dialogo nella giusta posizione cronologica all'interno dell'elenco.

- 5 Per aggiungere un parametro al cue point selezionato, fate clic sul segno più (+) sopra la sezione Parametri e immettete i valori nelle colonne Nome e Valore. Ripetete questa operazione per ciascun parametro.
- 6 Per aggiungere altri cue point ActionScript ripetete le operazioni descritte nei punti da 2 a 5 per ciascun cue point.
- 7 Fate clic su OK per salvare le modifiche.

Eliminare un cue point ActionScript:

- 1 Fate doppio clic sulla cella Valore del parametro `cuePoints` nella finestra di ispezione dei componenti per aprire la finestra di dialogo Cue point Flash.
- 2 Selezionate il cue point da eliminare.
- 3 Per eliminare il cue point, fate clic sul segno meno (-) nell'angolo in alto a sinistra, sopra l'elenco dei cue point.
- 4 Ripetete le operazioni descritte nei punti da 2 a 3 per ciascun cue point che desiderate eliminare.
- 5 Fate clic su OK per salvare le modifiche.

Per attivare o disattivare un cue point incorporato del file FLV:

- 1 Fate doppio clic sulla cella Valore del parametro `cuePoints` nella finestra di ispezione dei componenti per aprire la finestra di dialogo Cue point Flash.
- 2 Selezionate il cue point da attivare o disattivare.
- 3 Fate clic sul valore nella colonna Tipo per attivare il menu a comparsa, oppure fate clic sul pulsante freccia giù.
- 4 Fate clic sul nome corrispondente al tipo di cue point (ad esempio, Evento o Navigazione) per attivarlo. Fate clic su Disattivato per disattivarlo.
- 5 Fate clic su OK per salvare le modifiche.

Usare i cue point con ActionScript

Potete usare ActionScript per aggiungere cue point ActionScript, intercettare eventi `cuePoint`, trovare cue point di qualunque tipo o di un tipo specifico, cercare un cue point di navigazione, attivare o disattivare i cue point, controllare se un cue point è attivato e infine eliminare un cue point.

Negli esempi forniti in questa sezione viene usato un file FLV chiamato `cuepoints.flv`, che contiene i tre cue point seguenti:

Nome	Tempo	Tipo
point1	00:00:00.418	Navigazione
point2	00:00:07.748	Navigazione
point3	00:00:16.020	Navigazione

Aggiungere cue point ActionScript

Potete aggiungere cue point ActionScript a un file FLV mediante il metodo `addASCuePoint()`. Nell'esempio seguente vengono aggiunti due cue point ActionScript al file FLV quando questo è pronto per iniziare la riproduzione. Il primo cue point viene aggiunto utilizzando un oggetto cue point, che specifica il tempo, il nome e il tipo di cue point. La seconda chiamata specifica il tempo e il nome usando i parametri `time` e `name` del metodo.

```
// Requires an FLVPlayback instance called my_FLVPlayback on Stage
import fl.video.*;
import fl.video.MetadataEvent;
my_FLVPlayback.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
var cuePt:Object = new Object(); //create cue point object
cuePt.time = 2.02;
cuePt.name = "ASpt1";
cuePt.type = "actionscript";
my_FLVPlayback.addASCuePoint(cuePt); //add AS cue point
// add 2nd AS cue point using time and name parameters
my_FLVPlayback.addASCuePoint(5, "ASpt2");
```

Per ulteriori informazioni, vedete il metodo `FLVPlayback.addASCuePoint()` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Intercettare eventi cuePoint

L'evento `cuePoint` consente di controllare il codice ActionScript quando si verifica un tale evento. Quando si verifica un cue point nel seguente esempio, il listener `cuePoint` chiama una funzione del gestore di eventi che visualizza il valore della proprietà `playheadTime` e il nome e il tipo di cue point. Usare questo esempio in combinazione con l'esempio della precedente sezione Aggiungere cue point ActionScript per visualizzare i risultati.

Uso del componente FLVPlayback

```
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);
}
```

Per ulteriori informazioni sull'evento `cuePoint`, vedete l'evento `FLVPlayback.cuePoint` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Rilevare cue point

Mediante ActionScript potete cercare un cue point di qualunque tipo, il cue point in prossimità del tempo specificato o il cue point successivo con un nome specifico.

Il gestore di eventi `ready_listener()` nell'esempio seguente chiama il metodo `findCuePoint()` per cercare il cue point `ASpt1`, quindi il metodo `findNearestCuePoint()` per cercare il cue point di navigazione più vicino al tempo del cue point `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);
}
```

Nell'esempio seguente il gestore di eventi `ready_listener()` rileva il cue point `ASpt` e chiama il metodo `findNextCuePointWithName()` per rilevare il cue point successivo con lo stesso nome:

```
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);
}
```

Per ulteriori informazioni su come trovare i cue point, vedete i metodi `FLVPlayback.findCuePoint()`, `FLVPlayback.findNearestCuePoint()` e `FLVPlayback.findNextCuePointWithName()` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Rilevare cue point di navigazione

Potete cercare un cue point di navigazione oppure il cue point di navigazione successivo o precedente alla posizione temporale specificata. Nell'esempio seguente viene riprodotto il file `FLV cuepoints.flv` e viene cercato il cue point a 7.748 quando si verifica l'evento `ready`. Quando si verifica l'evento `cuePoint`, nell'esempio viene chiamato il metodo `seekToPrevNavCuePoint()` per cercare il primo cue point. Quando si verifica l'evento `cuePoint`, nell'esempio viene chiamato il metodo `seekToNextNavCuePoint()` per cercare l'ultimo cue point aggiungendo 10 secondi a `eventObject.info.time`, che rappresenta il tempo del cue point corrente.

```
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";
```

Per ulteriori informazioni, vedete i metodi `FLVPlayback.seekToNavCuePoint()`, `FLVPlayback.seekToNextNavCuePoint()` e `FLVPlayback.seekToPrevNavCuePoint()` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Attivare e disattivare dei cue point incorporati del file FLV

Potete attivare e disattivare i cue point del file FLV mediante il metodo `setFLVCuePointEnabled()`. I cue point disattivati non attivano l'evento `cuePoint` e non funzionano con i metodi `seekToCuePoint()`, `seekToNextNavCuePoint()` o `seekToPrevNavCuePoint()`. Tuttavia, potete trovare i cue point disattivati con i metodi `findCuePoint()`, `findNearestCuePoint()` e `findNextCuePointWithName()`.

Potete verificare se un cue point incorporato del file FLV viene attivato mediante il metodo `isFLVCuePointEnabled()`. Nel seguente esempio vengono disattivati i cue point incorporati `point2` e `point3` quando il video è pronto per la riproduzione. Tuttavia, quando si verifica il primo evento `cuePoint`, il gestore di eventi controlla se il cue point `point3` è disattivato e, in caso affermativo, lo attiva.

```
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");
    }
}
}
```

Per ulteriori informazioni, vedete i metodi `FLVPlayback.isFLVCuePointEnabled()` e `FLVPlayback.setFLVCuePointEnabled()` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Rimuovere un cue point ActionScript

Potete rimuovere un cue point ActionScript mediante il metodo `removeASCuePoint()`. Nel seguente esempio viene rimosso il cue point `ASpt2` quando si verifica il cue point `ASpt1`:

```
import fl.video.*;
my_FLVPlybk.source = "http://www.helpexamples.com/flash/video/cuepoints.flv"
my_FLVPlybk.addASCuePoint(2.02, "ASpt1");//add AS cue point
my_FLVPlybk.addASCuePoint(3.4, "ASpt2");//add 2nd Aspt
my_FLVPlybk.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
function cp_listener(eventObject:MetadataEvent):void {
    trace("Cue point name is: " + eventObject.info.name);
    if (eventObject.info.name == "ASpt1") {
        my_FLVPlybk.removeASCuePoint("ASpt2");
        trace("Removed cue point ASpt2");
    }
}
}
```

Per ulteriori informazioni, vedete il metodo `FLVPlayback.removeASCuePoint()` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Riprodurre file video multipli

Per riprodurre i file video in sequenza in un'istanza `FLVPlayback`, dovete semplicemente caricare un nuovo URL nella proprietà `source` al termine della riproduzione del file video precedente. Ad esempio, il seguente codice ActionScript intercetta l'evento `complete`, che si verifica al termine della riproduzione di un file video. Quando si verifica questo evento, il codice imposta il nome e la posizione del nuovo file video nella proprietà `source` e chiama il metodo `play()` per iniziare la riproduzione del nuovo video.

```
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");
    }
};
```

Usare lettori video multipli

Potete inoltre aprire più lettori video all'interno di una singola istanza del componente FLVPlayback, per riprodurre più video e passare da un video all'altro nel corso della riproduzione.

Il lettore video iniziale viene creato quando trascinate il componente FLVPlayback nello stage. Il componente assegna automaticamente il valore 0 al lettore video iniziale e lo imposta come lettore predefinito. Per creare un lettore video aggiuntivo è sufficiente impostare un nuovo numero per la proprietà `activeVideoPlayerIndex`. Con l'impostazione della proprietà `activeVideoPlayerIndex`, il lettore video specificato diventa attivo (*active*) ed è quindi influenzato dalle proprietà e dai metodi della classe FLVPlayback. Tuttavia, l'impostazione della proprietà `activeVideoPlayerIndex` non rende automaticamente visibile il lettore video. Per rendere visibile il lettore video, impostate la proprietà `visibleVideoPlayerIndex` sul numero del lettore video. Per ulteriori informazioni su come queste proprietà interagiscono con i metodi e le proprietà della classe FLVPlayback, vedete le proprietà `FLVPlayback.activeVideoPlayerIndex` e `FLVPlayback.visibleVideoPlayerIndex` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Il seguente codice ActionScript carica la proprietà `source` per riprodurre un file video nel lettore video predefinito e aggiunge un cue point. Quando si verifica l'evento `ready`, il gestore di eventi apre un secondo lettore video impostando la proprietà `activeVideoPlayerIndex` su 1. Specifica un file FLV e un cue point per il secondo lettore video, quindi rende nuovamente attivo il lettore predefinito (0).

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

Per passare a un file FLV mentre è in corso la riproduzione di un altro file, è necessario effettuare questo passaggio nel codice ActionScript. I cue point consentono di intervenire in punti specifici del file FLV mediante l'evento `cuePoint`. Il seguente codice crea un listener per l'evento `cuePoint` e chiama la funzione del gestore che mette in pausa il lettore video attivo (0), passa al secondo lettore (1) e riproduce il file FLV:


```
import fl.video.*;
// add listener for a cuePoint event
my_FLVPlayback.addEventListener(MetadataEvent.CUE_POINT, cp_listener);
// add the handler function for the cuePoint event
function cp_listener(eventObject:MetadataEvent):void {
    // display the no. of the video player causing the event
    trace("Hit cuePoint event for player: " + eventObject.vp);
    // test for the video player and switch FLV files accordingly
    if (eventObject.vp == 0) {
        my_FLVPlayback.pause(); //pause the first FLV file
        my_FLVPlayback.activeVideoPlayerIndex = 1; // make the 2nd player active
        my_FLVPlayback.visibleVideoPlayerIndex = 1; // make the 2nd player visible
        my_FLVPlayback.play(); // begin playing the new player/FLV
    } else if (eventObject.vp == 1) {
        my_FLVPlayback.pause(); // pause the 2nd FLV
        my_FLVPlayback.activeVideoPlayerIndex = 0; // make the 1st player active
        my_FLVPlayback.visibleVideoPlayerIndex = 0; // make the 1st player visible
        my_FLVPlayback.play(); // begin playing the 1st player
    }
}
my_FLVPlayback.addEventListener(VideoEvent.COMPLETE, complete_listener);
function complete_listener(eventObject:VideoEvent):void {
    trace("Hit complete event for player: " + eventObject.vp);
    if (eventObject.vp == 0) {
        my_FLVPlayback.activeVideoPlayerIndex = 1;
        my_FLVPlayback.visibleVideoPlayerIndex = 1;
        my_FLVPlayback.play();
    } else {
        my_FLVPlayback.closeVideoPlayer(1);
    }
};
```

Quando create un nuovo lettore video, l'istanza FLVPlayback imposta le proprietà con i valori del lettore predefinito, a eccezione delle proprietà `source`, `totalTime` e `isLive` che vengono impostate sempre sui valori predefiniti dall'istanza di FLVPlayback: rispettivamente, stringa vuota, 0 e `false`. La proprietà `autoplay`, che utilizza `true` come impostazione predefinita per il lettore video predefinito, viene impostata su `false`. La proprietà `cuePoints` non ha alcun effetto, così come non ne ha su un eventuale caricamento successivo nel lettore video predefinito.

I metodi e le proprietà che controllano il volume, il posizionamento, le dimensioni, la visibilità e i controlli UI sono sempre globali e il loro comportamento non viene alterato dall'impostazione della proprietà `activeVideoPlayerIndex`. Per ulteriori informazioni su questi metodi e proprietà e l'effetto dell'impostazione della proprietà `activeVideoPlayerIndex`, vedete la proprietà `FLVPlayback.activeVideoPlayerIndex` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#). I restanti metodi e proprietà definiscono come destinazione il lettore video identificato dal valore della proprietà `activeVideoPlayerIndex`.

Tuttavia, le proprietà e i metodi che controllano le dimensioni *interagiscono effettivamente* con la proprietà `visibleVideoPlayerIndex`. Per ulteriori informazioni, vedete la proprietà `FLVPlayback.visibleVideoPlayerIndex` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Streaming di file FLV da Flash Media Server

I requisiti per lo streaming di file FLV da Flash Media Server sono diversi a seconda che sia disponibile o meno il rilevamento nativo della larghezza di banda da parte del fornitore del servizio FVSS (Flash Video Streaming Service). Il rilevamento della larghezza di banda si definisce nativo quando questa funzione è incorporata nel server di streaming e assicura quindi prestazioni migliori. Verificate pertanto con il vostro provider se è disponibile il rilevamento nativo della larghezza di banda.

Per accedere ai file FLV sul server FMS, utilizzate un URL, ad esempio
`rtmp://my_servername/my_application/stream.flv`.

Quando riproducete un flusso dal vivo con il server FMS, dovete impostare la proprietà `isLive` di `FLVPlayback` su `true`. Per ulteriori informazioni, vedete la proprietà `FLVPlayback.isLive` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Per ulteriori informazioni sull'amministrazione di Flash Media Server, compresa l'impostazione di un flusso dal vivo, vedete la documentazione di Flash Media Server all'indirizzo www.adobe.com/support/documentation/en/flashmediaserver/.

Per il rilevamento nativo della larghezza di banda o nessun rilevamento

La classe `NCManagerNative` è una sottoclasse di `NCManager` che supporta il rilevamento nativo della larghezza di banda, che può essere supportato da alcuni fornitori del servizio FVSS (Flash Video Streaming Service). Quando usate `NCManagerNative`, non sono necessari file speciali sul server FMS. `NCManagerNative` consente inoltre la connessione a qualsiasi versione di Flash Media Server, senza un file `main.asc`, se non è richiesto il rilevamento della larghezza di banda.

Per usare `NCManagerNative` al posto della classe `NCManager`, aggiungete le seguenti righe di codice nel primo fotogramma del file FLA:

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

Per il rilevamento non nativo della larghezza di banda

Se il fornitore del servizio FVSS non offre il rilevamento nativo della larghezza di banda, ma questa funzione è necessaria, dovete aggiungere il file `main.asc` all'applicazione FLV di Flash Media Server. Potete trovare il file `main.asc` in linea all'indirizzo www.adobe.com/go/learn_fl_samples_it. Si trova nel file `Samples.zip` all'interno della directory `Samples\ComponentsAS2\FLVPlayback`.

Per impostare lo streaming di file FLV su Flash Media Server:

- 1 Create una cartella nella cartella dell'applicazione Flash Media Server e assegnatele un nome, ad esempio **my_application**.
- 2 Copiate il file `main.asc` nella cartella `my_application`.
- 3 Create una cartella denominata **streams** nella cartella `my_application`.
- 4 Create una cartella denominata **_definst_** nella cartella `streams`.
- 5 Inserite i file FLV nella cartella **_definst_**.

Personalizzare il componente FLVPlayback

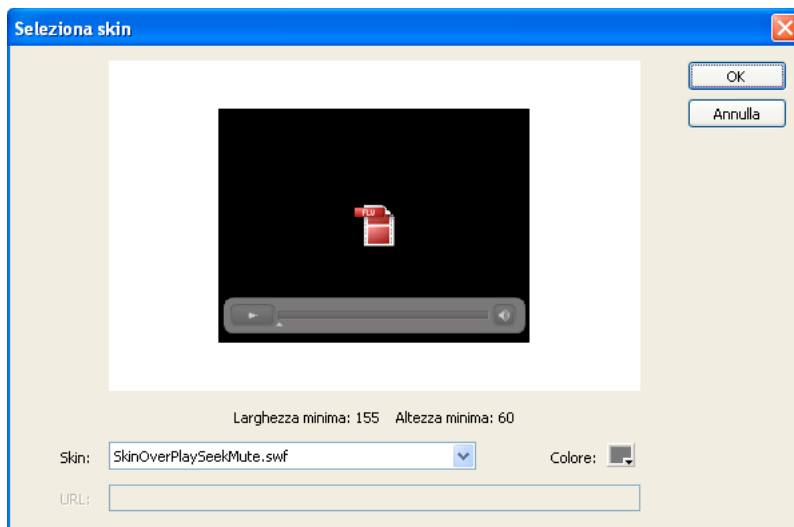
In questa sezione viene illustrato come personalizzare il componente FLVPlayback. La maggior parte dei metodi utilizzati per personalizzare altri componenti non funziona con il componente FLVPlayback. Per personalizzare il componente FLVPlayback, utilizzate solo le tecniche descritte in questa sezione.

Per personalizzare il componente FLVPlayback potete scegliere tra selezionare uno skin predefinito, selezionare i singoli componenti UI FLV di riproduzione personalizzati, oppure creare un nuovo skin. Per modificare il comportamento di uno skin potete inoltre utilizzare le proprietà FLVPlayback.

Nota: perché lo skin funzioni con il componente FLVPlayback, dovete caricare il file SWF dello skin e il file di programma SWF sul server Web.

Selezionare uno skin predefinito

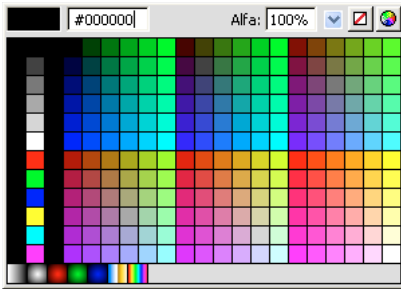
Per selezionare uno skin per il componente FLVPlayback, fate clic sulla cella `valore` del parametro `skin` nella finestra di ispezione dei componenti, quindi fate clic sull'icona della lente di ingrandimento per aprire la seguente finestra di dialogo Seleziona skin, che consente di selezionare uno skin o di immettere un URL che specifica la posizione del file SWF dello skin.



Finestra di dialogo Seleziona skin di FLVPlayback

Gli skin elencati nel menu a comparsa Skin si trovano nella cartella dell'applicazione Flash /Flash Configuration/FLVPlayback Skins/ActionScript 3.0. Potete rendere disponibili nuovi skin per questa finestra di dialogo creandoli e collocando il file SWF nella cartella. Il nome viene visualizzato nel menu a comparsa con l'estensione `.swf`. Per ulteriori informazioni sulla creazione di skin, vedete “[Creare un nuovo skin](#)” a pagina 166.

Per gli skin assegnati mediante l'impostazione della proprietà `skin`, sia tramite il parametro `skin` in fase di creazione sia tramite ActionScript in fase di runtime, potete assegnare i valori di colore e alfa (trasparenza) indipendentemente dalla scelta dello skin. Per assegnare i valori di colore e alfa in fase di creazione, aprite il Selettore colore nella finestra di dialogo Seleziona skin, come illustrato di seguito.



Selettore del colore nella finestra di dialogo Selezione skin

Per scegliere il colore, fate clic su un campione nel pannello o inserite il relativo valore numerico nella casella di testo. Per scegliere il valore alfa, usate il cursore o digitate una percentuale nella casella di testo Alfa.

Per assegnare i valori di colore e alfa in fase di runtime, impostate le proprietà `skinBackgroundColor` e `skinBackgroundAlpha`. Impostate la proprietà `skinBackgroundColor` sul valore `0xRRGGBB` (rosso, verde, blu). Impostate la proprietà `skinBackgroundAlpha` su un numero compreso tra 0,0 e 1,0. L'esempio seguente imposta `skinBackgroundColor` su `0xFF0000` (rosso) e `skinBackgroundAlpha` su 0,5.

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

I valori predefiniti sono gli ultimi valori scelti dall'utente.

Se desiderate assegnare uno skin al componente FLVPlayback usando i componenti UI FLV di riproduzione personalizzati, selezionate Nessuno dal menu a comparsa.

Associare skin ai singoli componenti UI FLV di riproduzione personalizzati

I componenti UI FLV di riproduzione personalizzati consentono di personalizzare l'aspetto dei controlli FLVPlayback all'interno del file FLV e permettono di visualizzare i risultati nell'anteprima della pagina Web. Tuttavia, questi componenti non sono concepiti per essere modificati in scala. Dovete modificare il clip filmato e il suo contenuto secondo una dimensione specifica. Per questo motivo, in genere è preferibile che il componente FLVPlayback sullo stage sia delle dimensioni desiderate, con la proprietà `scaleMode` impostata su `exactFit`.

Per iniziare, trascinate i componenti UI FLV di riproduzione personalizzati dal pannello Componenti, posizionateli dove preferite sullo stage e assegnate a ciascuno un nome di istanza.

Questi componenti possono funzionare senza ActionScript. Se sono posizionati sulla stessa linea temporale e sullo stesso fotogramma del componente FLVPlayback e per il componente non è impostato uno skin, il componente FLVPlayback si connette automaticamente a tali componenti. Se sullo stage sono presenti più componenti FLVPlayback o se il controllo personalizzato e l'istanza FLVPlayback non sono sulla stessa linea temporale, sono necessarie ulteriori operazioni.

Una volta che i componenti si trovano sullo stage, possono essere modificati come qualunque altro simbolo. Quando aprite i componenti, noterete che hanno impostazioni leggermente diverse l'uno dall'altro.

Componenti Button

I componenti Button hanno una struttura simile. Questi pulsanti includono i componenti BackButton, ForwardButton, MuteButton, PauseButton, PlayButton, PlayPauseButton e StopButton. La maggior parte di essi presenta un unico clip filmato nel fotogramma 1 con il nome di istanza placeholder_mc. Generalmente si tratta di un'istanza dello stato normale del pulsante, ma non è necessariamente così. Nel fotogramma 2 ci sono quattro clip filmato sullo stage per ogni visualizzazione: normale, sopra, giù e disabilitato. (In fase di runtime, il componente non passa mai al fotogramma 2; questi clip filmato si trovano in questa posizione per rendere più comode le operazioni di modifica e per forzarne il caricamento nel file SWF senza selezionare la casella di controllo Esporta nel primo fotogramma della finestra di dialogo Proprietà simbolo. È comunque necessario selezionare l'opzione Esporta per ActionScript.)

Per assegnare uno skin al pulsante, è sufficiente modificare ciascun clip filmato. Potete cambiare la dimensione e l'aspetto di ciascun clip.

Di solito appare un'istruzione ActionScript sul fotogramma 1. Non dovrebbe essere necessario modificare questo script. Questo script semplicemente arresta l'indicatore di riproduzione sul fotogramma 1 e specifica quale clip filmato utilizzare per un determinato stato.

Pulsanti PlayPauseButton, MuteButton, FullScreenButton e CaptionButton

I pulsanti PlayPauseButton, MuteButton, FullScreenButton e CaptionButton hanno delle impostazioni diverse dagli altri pulsanti; infatti dispongono soltanto di un fotogramma con due livelli e nessuno script. Su questo fotogramma si trovano due pulsanti, uno sopra l'altro: nel caso di PlayPauseButton, si tratta del pulsante per riprodurre e sospendere la riproduzione; nel caso di MuteButton del pulsante per attivare e disattivare il volume; nel caso di FullScreenButton del pulsante per attivare e disattivare la modalità schermo intero; nel caso di CaptionButton del pulsante per attivare e disattivare le didascalie. Potete associare degli skin a ognuno di questi due pulsanti interni come descritto in [“Associare skin ai singoli componenti UI FLV di riproduzione personalizzati”](#) a pagina 160; non sono richieste altre operazioni.

CaptionButton è destinato al componente FLVPlaybackCaptioning e deve essere associato a quel componente, non al componente FLVPlayback.

Pulsanti BackButton e ForwardButton

Anche i pulsanti BackButton e ForwardButton hanno delle impostazioni diverse dagli altri pulsanti. Sul fotogramma 2 sono presenti dei clip filmato aggiuntivi da utilizzare come un fotogramma intorno a uno o a entrambi i pulsanti. Questi clip filmato non sono obbligatori e non dispongono di alcuna funzione specifica; vengono forniti solo per comodità. Per utilizzarli è sufficiente trascinarli nello stage dal pannello Libreria e posizionarli in un punto qualsiasi dello stage. Se non li ritenete importanti, non utilizzateli o cancellateli dal pannello Libreria.

La maggior parte dei pulsanti, così come vengono forniti, si basa su un insieme comune di clip filmato: in tal modo potete modificare contemporaneamente l'aspetto di tutti i pulsanti. Potete decidere di utilizzare questa funzione oppure potete sostituire i clip comuni e scegliere un aspetto diverso per ciascun pulsante.

Componente BufferingBar

BufferingBar è un componente piuttosto semplice: è costituito da un'animazione resa visibile quando il componente attiva lo stato di bufferizzazione e non necessita di alcuna speciale azione ActionScript per essere configurato. Per impostazione predefinita, è una barra a strisce che si sposta da sinistra a destra con una maschera rettangolare, ma senza alcuna altra configurazione particolare.

Sebbene le barre di bufferizzazione nei file SWF dello skin utilizzino la modifica in scala a 9 porzioni, poiché devono essere modificate in scala in fase di runtime, il componente UI personalizzato FLV BufferingBar non utilizza e *non* può utilizzare la modifica in scala a 9 porzioni perché include dei clip filmato nidificati. Se desiderate che il componente BufferingBar sia più largo o più alto, può essere utile cambiare il contenuto piuttosto che modificarlo in scala.

Componenti SeekBar e VolumeBar

I componenti SeekBar e VolumeBar sono simili, ma hanno funzioni diverse. Entrambi dispongono di determinate maniglie, adottano lo stesso meccanismo di gestione di traccia e hanno un supporto per i clip nidificati per tenere traccia dello stato di avanzamento (progress) e di completamento (fullness).

Spesso il codice ActionScript del componente FLVPlayback presuppone che il punto di registrazione (detto anche *punto di origine* o *punto zero*) del componente SeekBar o VolumeBar si trovi nell'angolo in alto a sinistra del contenuto, pertanto è importante mantenere questa convenzione. In caso contrario, potrebbero verificarsi problemi con le maniglie e con i clip filmato progress e fullness.

Sebbene le barre di ricerca nei file SWF dello skin utilizzino la modifica in scala a 9 porzioni, poiché devono essere modificate in scala in fase di runtime, il componente UI personalizzato FLV SeekBar non utilizza e *non può* utilizzare la modifica in scala a 9 porzioni perché include dei clip filmato nidificati. Se desiderate che il componente SeekBar sia più largo o più alto, può essere utile cambiare il contenuto piuttosto che modificarlo in scala.

Maniglia

Un'istanza della maniglia clip filmato si trova sul fotogramma 2. Così come accade per i componenti BackButton e ForwardButton, il componente non passa mai al fotogramma 2; questi clip filmato si trovano in questa posizione per rendere più comode le operazioni di modifica e per forzarne il caricamento nel file SWF senza dover selezionare la casella di controllo Esporta nel primo fotogramma nella finestra di dialogo Proprietà simbolo. È comunque necessario selezionare l'opzione Esporta per ActionScript.

Come potete notare, la maniglia clip filmato presenta un rettangolo di sfondo con il valore alfa impostato su 0. Questo rettangolo aumenta le dimensioni dell'area attiva della maniglia, rendendo più semplice l'acquisizione senza dover modificare l'aspetto, come accade per lo stato area attiva di un pulsante. Poiché la maniglia viene creata in modo dinamico in fase di runtime, è necessario che essa sia un clip filmato e non un pulsante. Il rettangolo con il valore alfa impostato su 0 non è strettamente indispensabile e, in generale, si può sostituire l'interno della maniglia con qualsiasi altra immagine. Tuttavia, è consigliabile lasciare il punto di registrazione in posizione centrale orizzontale, al centro della maniglia clip filmato.

Il seguente codice ActionScript si trova nel fotogramma 1 del componente SeekBar per gestire le operazioni della maniglia:

```
stop();  
handleLinkageID = "SeekBarHandle";  
handleLeftMargin = 2;  
handleRightMargin = 2;  
handleY = 11;
```

La chiamata alla funzione `stop()` è necessaria per il contenuto del fotogramma 2.

La seconda riga specifica quale simbolo usare come maniglia; non occorre cambiarlo se modificate solo l'istanza della maniglia clip filmato sul fotogramma 2. In fase di runtime, il componente FLVPlayback crea sullo stage un'istanza del clip filmato specificato di pari livello dell'istanza del componente Bar, il che significa che i due componenti hanno lo stesso clip filmato principale. Così se ad esempio la barra si trova al livello principale, anche la maniglia dovrà essere collocata allo stesso livello.

La variabile `handleLeftMargin` determina la posizione originale della maniglia (0%), mentre la variabile `handleRightMargin` determina la posizione finale (100%). I numeri forniscono l'offset dall'estremità sinistra a quella destra del controllo della barra, dove i numeri positivi indicano i limiti all'interno della barra, mentre quelli negativi indicano i limiti al di fuori di essa. L'offset specifica dove può spostarsi la maniglia, in base al suo punto di registrazione. Se posizionate il punto di registrazione al centro della maniglia, le estremità destra e sinistra della maniglia supereranno i margini. Per funzionare correttamente, il punto di registrazione di un clip filmato della barra di ricerca deve trovarsi nell'angolo in alto a sinistra del contenuto.

La variabile `handleY` determina la posizione *y* della maniglia, relativa all'istanza della barra. Questa si basa sui punti di registrazione di ciascun clip filmato. Il punto di registrazione della maniglia *sample* si trova sul vertice del triangolo, nella posizione relativa nella parte visibile, senza considerare lo stato area attiva invisibile del rettangolo. Anche il punto di registrazione di un clip filmato della barra deve trovarsi nell'angolo in alto a sinistra del contenuto per funzionare correttamente.

Così considerando questi limiti, se ad esempio il controllo della barra è impostato su (100, 100) e ha una larghezza di 100 pixel, i valori possibili della maniglia saranno compresi tra 102 e 198 in direzione orizzontale e 111 in direzione verticale. Se cambiate i valori delle variabili `handleLeftMargin` e `handleRightMargin` su -2 e quello della variabile `handleY` su -11, i valori possibili della maniglia saranno compresi tra 98 e 202 in direzione orizzontale e 89 in direzione verticale.

Clip filmato progress e fullness

I componenti `SeekBar` e `VolumeBar` dispongono, rispettivamente, di un clip filmato *progress* e *fullness*, anche se nella pratica possono avere entrambi, nessuno o uno solo di questi clip filmato. Da un punto di vista strutturale e comportamentale sono uguali, ma tengono traccia di valori diversi. Un clip filmato di tipo *progress* si riempie durante la fase di scaricamento del file FLV (utile solo in caso di scaricamento via HTTP, perché risulta già pieno in caso di streaming da FMS), mentre un clip filmato di tipo *fullness* si riempie mentre la maniglia si sposta da sinistra a destra.

Il componente `FLVPlayback` rileva le istanze dei clip filmato cercando un nome di istanza specifico, pertanto è necessario che l'istanza del clip filmato di tipo *progress* abbia come istanza principale il clip filmato della barra e come nome di istanza `progress_mc`. Il nome del clip filmato di tipo *fullness* deve essere `fullness_mc`.

Potete impostare i clip filmato di tipo *progress* e *fullness* con o senza l'istanza di clip filmato `fill_mc` nidificata all'interno. Il clip filmato `fullness_mc` di `VolumeBar` mostra il metodo *con* il clip filmato `fill_mc`, mentre il clip filmato `progress_mc` di `SeekBar` mostra il metodo *senza* il clip filmato `fill_mc`.

Il metodo con il clip filmato `fill_mc` nidificato all'interno è utile quando desiderate un riempimento che non possa essere adattato in scala senza distorcerne l'aspetto.

Nel clip filmato `fullness_mc` di `VolumeBar` l'istanza del clip filmato nidificato `fill_mc` è mascherata. Potete mascherarla quando create il clip filmato oppure creando una maschera in modo dinamico in fase di runtime. Se scegliete di creare un clip filmato da usare come maschera, denominate l'istanza **mask_mc** e impostatela in modo che `fill_mc` venga visualizzato come quando la percentuale è pari al 100%. Se non mascherate `fill_mc`, la maschera creata in maniera dinamica sarà di forma rettangolare e con le stesse dimensioni di `fill_mc` pari a una percentuale del 100%.

Il clip filmato `fill_mc` viene rivelato tramite la maschera in uno dei due modi possibili, a seconda che `fill_mc.slideReveal` sia `true` o `false`.

Se `fill_mc.slideReveal` è `true`, `fill_mc` viene spostato da sinistra a destra per mostrarlo attraverso la maschera. Con la percentuale del 0%, il clip filmato si trova completamente a sinistra, in modo che nessuna parte venga mostrata attraverso la maschera. Con l'aumentare della percentuale il clip filmato si sposta verso destra, finché con il 100% torna nella stessa posizione in cui si trovava quando è stato creato sullo stage.

Se `fill_mc.slideReveal` è `false` o `undefined` (comportamento predefinito), la maschera viene ridimensionata da sinistra a destra per rivelare una parte maggiore del clip filmato `fill_mc`. Con il valore 0%, la maschera viene modificata in scala su 05 in direzione orizzontale e, con l'aumentare della percentuale, la proprietà `scaleX` aumenta finché al 100% il clip filmato `fill_mc` viene rivelato del tutto. Questo non significa necessariamente che `scaleX = 100`, perché `mask_mc` potrebbe essere stato modificato in scala nel momento in cui è stato creato.

Il metodo senza `fill_mc` è più semplice del metodo con `fill_mc`, ma il file viene distorto in senso orizzontale. Se non desiderate tale distorsione, dovete utilizzare `fill_mc`. Il clip filmato `progress_mc` di `SeekBar` illustra come usare questo metodo.

Il clip filmato di tipo `progress` o `fullness` viene modificato in scala in direzione orizzontale in base alla percentuale. Allo 0%, la proprietà `scaleX` dell'istanza è impostata su 0, perciò l'istanza risulta invisibile. Con l'aumentare della percentuale viene regolata anche la proprietà `scaleX` finché, con una percentuale del 100%, il clip presenta le stesse dimensioni di quando è stato creato sullo stage. Dunque, anche in questo caso non significa necessariamente che `scaleX = 100`, perché l'istanza del clip potrebbe essere stata modificata in scala nel momento in cui è stata creata.

Connettere i componenti UI FLV di riproduzione personalizzati

Se posizionate i componenti UI personalizzati sulla stessa linea temporale e sullo stesso fotogramma del componente FLVPlayback e per il componente non è impostata la proprietà `skin`, il componente FLVPlayback si connette automaticamente a tali componenti senza richiedere ActionScript.

Se sullo stage sono presenti più componenti FLVPlayback o se il controllo personalizzato e l'istanza FLVPlayback non sono sulla stessa linea temporale, dovete scrivere del codice ActionScript per connettere i componenti UI personalizzati all'istanza del componente FLVPlayback. Innanzitutto, dovete assegnare un nome all'istanza FLVPlayback e, mediante ActionScript, assegnare le istanze dei componenti UI FLV di riproduzione personalizzati alle proprietà FLVPlayback corrispondenti. Nell'esempio seguente l'istanza FLVPlayback è `my_FLVPlayback`, i nomi di proprietà FLVPlayback seguono il punto (.) e le istanze dei controlli UI FLV di riproduzione personalizzati si trovano a destra del segno uguale (=):

```
//FLVPlayback instance = my_FLVPlayback
my_FLVPlayback.playButton = playbtn; // set playButton prop. to playbtn, etc.
my_FLVPlayback.pauseButton = pausebtn;
my_FLVPlayback.playPauseButton = playpausebtn;
my_FLVPlayback.stopButton = stopbtn;
my_FLVPlayback.muteButton = mutebtn;
my_FLVPlayback.backButton = backbtn;
my_FLVPlayback.forwardButton = forbtn;
my_FLVPlayback.volumeBar = volbar;
my_FLVPlayback.seekBar = seekbar;
my_FLVPlayback.bufferingBar = bufbar;
```

Nei seguenti passaggi vengono creati i controlli personalizzati `StopButton`, `PlayPauseButton`, `MuteButton` e `SeekBar`:

- 1 Trascinate il componente FLVPlayback nello stage e assegnate il nome di istanza **my_FLVPlayback**.
- 2 Impostate il parametro `source` nella finestra di ispezione dei componenti su <http://www.helpexamples.com/flash/video/cuepoints.flv>.
- 3 Impostate il parametro `skin` su `None`.
- 4 Trascinate i componenti `StopButton`, `PlayPauseButton` e `MuteButton` nello stage e posizionatevi sopra l'istanza FLVPlayback, impilandoli verticalmente sulla sinistra. Assegnate a ciascun pulsante un nome di istanza, inserendolo nella finestra di ispezione Proprietà (ad esempio **my_stopbtn**, **my_plypausbtn** e **my_mutebtn**).
- 5 Nel pannello Libreria, aprite la cartella FLVPlayback Skins, quindi aprite la sottocartella `SquareButton`.
- 6 Selezionate il clip filmato `SquareBgDown` e fate doppio clic per aprirlo nello stage.

- 7 Fate clic con il pulsante destro del mouse (Windows) o fate clic tenendo premuto il tasto Ctrl (Macintosh), quindi scegliete Seleziona tutto dal menu ed eliminate il simbolo.
- 8 Selezionate lo strumento Ovale, disegnate un ovale nella stessa posizione e impostate il riempimento sul colore blu (#0033FF).
- 9 Nella finestra di ispezione Proprietà, impostate la larghezza (W:) su 40 e l'altezza (H:) su 20. Impostate la coordinata x (X:) su 0.0 e la coordinata y (Y:) su 0.0.
- 10 Ripetete le operazioni elencate ai punti da 6 a 8 per SquareBgNormal, ma impostate il riempimento sul colore giallo (#FFFF00).
- 11 Ripetete le operazioni elencate ai punti da 6 a 8 per SquareBgOver, ma impostate il riempimento sul colore verde (#006600).
- 12 Modificate i clip filmato per le varie icone del simbolo all'interno dei pulsanti (PauseIcon, PlayIcon, MuteOnIcon, MuteOffIcon e StopIcon). Questi clip filmato si possono trovare nel pannello Libreria sotto FLV Playback Skins/Label Button/Assets, dove Label è il nome del pulsante (Play, Pause e così via). Per ogni clip filmato, eseguite queste operazioni:
 - a Selezionate l'opzione Seleziona tutto.
 - b Cambiate il colore in rosso (#FF0000).
 - c Ridimensionate a 300%.
 - d Modificate la posizione X: del contenuto su 7,0 per alterare la posizione orizzontale dell'icona in tutti gli stati del pulsante.

Nota: questo metodo di modifica della posizione evita di aprire tutti gli stati del pulsante e di spostare l'istanza dell'icona clip filmato.
- 13 Fate clic sulla freccia Indietro sopra la linea temporale per tornare alla scena 1, fotogramma 1.
- 14 Trascinate il componente SeekBar nello stage e posizionatelo nell'angolo in basso a destra dell'istanza FLVPlayback.
- 15 Nel pannello Libreria, fate doppio clic su SeekBar per aprirlo nello stage.
- 16 Modificatelo in scala del 400%.
- 17 Selezionate il contorno e impostate il colore su rosso (#FF0000).
- 18 Fate doppio clic su SeekBarProgress nella cartella Skins/Seek Bar di FLVPlayback e impostate il colore su giallo (#FFFF00).
- 19 Fate doppio clic su SeekBarHandle nella cartella Skins/Seek Bar di FLVPlayback e impostate il colore su rosso (#FF0000).
- 20 Fate clic sulla freccia Indietro sopra la linea temporale per tornare alla scena 1, fotogramma 1.
- 21 Selezionate l'istanza SeekBar sullo stage e assegnate il nome di istanza **my_seekbar**.
- 22 Nel pannello Azioni del fotogramma 1 della linea temporale aggiungete un'istruzione `import` per le classi video e assegnate i nomi del pulsante e della barra di ricerca alle proprietà FLVPlayback corrispondenti, come mostrato nel seguente esempio:

```
import fl.video.*;
my_FLVPlybk.stopButton = my_stopbbtn;
my_FLVPlybk.playPauseButton = my_plypausbbtn;
my_FLVPlybk.muteButton = my_mutebbtn;
my_FLVPlybk.seekBar = my_seekbar;
```
- 23 Per verificare il filmato, premete Ctrl+Invio.

Creare un nuovo skin

Il modo migliore per creare un file SWF dello skin è di copiare uno dei file di skin di Flash e usarlo come punto di partenza. I file FLA per questi skin si trovano nella cartella dell'applicazione Flash in Configuration/FLVPlayback Skins/FLA/ActionScript 3.0/. Per rendere disponibile il file finale SWF dello skin come opzione della finestra di dialogo Seleziona skin, inseritelo nella cartella Configuration/FLVPlayback Skins/ActionScript 3.0, nella cartella dell'applicazione Flash o in una cartella locale degli utenti Configuration/FLVPlayback Skins/ActionScript 3.0.

Poiché potete impostare il colore di uno skin anche senza sceglierlo, non è necessario modificare il file FLA per cambiare il colore. Se create uno skin con un colore specifico e non desiderate che sia modificabile nella finestra di dialogo Seleziona skin, impostate `this.border_mc.colorMe = false;` nel codice ActionScript del file FLA dello skin. Per informazioni sull'impostazione del colore di uno skin, vedete “[Selezionare uno skin predefinito](#)” a pagina 159.

Esaminando i file FLA dello skin installati su Flash, alcuni elementi sullo stage potrebbero sembrare non necessari, ma molti di essi sono inseriti nei livelli guida. Con l'anteprima dal vivo in scala 9, potete verificare velocemente gli elementi effettivamente visualizzati nel file SWF.

Le sezioni seguenti riguardano le personalizzazioni più complesse e le modifiche ai clip filmato SeekBar, BufferingBar e VolumeBar.

Usare il layout dello skin

Quando aprite un file FLA dello skin in Flash, i clip filmato dello skin sono disposti sulla linea temporale principale. Questi clip e il codice ActionScript che si trova sullo stesso fotogramma definiscono la disposizione dei controlli in fase di runtime.

Sebbene il livello Layout appaia molto simile a come verrà visualizzato lo skin in fase di runtime, in questa fase i contenuti del livello non sono visibili. Serve solo a calcolare dove posizionare i controlli. Gli altri controlli presenti sullo stage vengono utilizzati in fase di runtime.

All'interno del livello Layout si trova un segnaposto per il componente FLVPlayback denominato `video_mc`. Tutti gli altri controlli sono disposti in base a `video_mc`. Se iniziate con uno dei file FLA di Flash e modificate la dimensione dei controlli, potrebbe essere possibile correggere il layout spostando i clip segnaposto.

Ogni clip segnaposto ha uno specifico nome di istanza. I nomi dei clip segnaposto sono: `playpause_mc`, `play_mc`, `pause_mc`, `stop_mc`, `captionToggle_mc`, `fullScreenToggle_mc`, `back_mc`, `bufferingBar_mc`, `bufferingBarFill_mc`, `seekBar_mc`, `seekBarHandle_mc`, `seekBarProgress_mc`, `volumeMute_mc`, `volumeBar_mc` e `volumeBarHandle_mc`. La parte che viene ricolorata quando scegliete un colore dello skin è detta `border_mc`.

Non importa quale clip venga utilizzato come controllo. Generalmente, per i pulsanti viene utilizzato il clip di stato normale. Per gli altri controlli vengono utilizzati i clip corrispondenti a quel controllo, ma solo per comodità. Ciò che è veramente importante sono le posizioni *x* (orizzontale) e *y* (verticale), l'altezza e la larghezza del segnaposto.

Oltre ai controlli standard, potete disporre di molti tipi di clip aggiuntivi. Il solo requisito per questi clip è accertarvi che per i relativi simboli nella libreria sia selezionato Esporta per ActionScript nella finestra di dialogo Concatenamento. Ai clip personalizzati nel livello Layout potete assegnare un nome di istanza qualsiasi, eccetto i nomi di istanza riservati elencati precedentemente. Il nome di istanza è necessario solo per l'impostazione di ActionScript sui clip al fine di determinare il layout.

Il clip `border_mc` è particolare. Se impostate la proprietà `FlvPlayback.skinAutoHide` su `true`, lo skin mostra quando il mouse si trova sul clip `border_mc`. Ciò è importante per gli skin che appaiono al di fuori dei limiti del lettore video. Per informazioni sulla proprietà `skinAutoHide`, vedete “[Modificare il comportamento di uno skin](#)” a pagina 170.

Tra i file FLA di Flash, `border_mc` viene usato per il colore di sfondo e per il bordo intorno ai pulsanti Avanti e Indietro.

Il clip `border_mc` fa parte anche dello skin i cui valori alfa e di colore vengono modificati dalle proprietà `skinBackgroundAlpha` e `skinBackgroundColor`. Per consentire l'uso di valori alfa e di colore personalizzabili, il codice ActionScript nel file FLA dello skin deve includere il codice seguente:

```
border_mc.colorMe = true;
```

ActionScript e il layout dello skin

In genere, il seguente codice ActionScript viene applicato a tutti i controlli. Alcuni controlli hanno un codice specifico ActionScript che definisce un comportamento aggiuntivo; questo viene spiegato nella sezione che riguarda quel determinato controllo.

Il codice ActionScript iniziale è una sezione estesa che specifica i nomi di classe per ogni stato di ogni componente. Potete visualizzare tutti questi nomi di classe nel file `SkinOverAll fla`. Ad esempio, per i pulsanti Riproduci e Pausa il codice è simile al seguente:

```
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";
```

I nomi di classe non dispongono in effetti di file di classe esterni; vengono specificati solo nella finestra di dialogo Concatenamento per tutti i clip filmato presenti nella libreria.

Nel componente ActionScript 2.0, sullo stage erano presenti clip filmato usati effettivamente in fase di runtime. Nel componente ActionScript 3.0, questi clip filmato sono ancora contenuti nel file FLA ma solo per rendere più comode le operazioni di modifica. Ora sono tutti nei livelli guida e non vengono esportati. Tutte le risorse degli skin nella libreria sono impostate su Esporta nel primo fotogramma e vengono create in modo dinamico mediante codice simile al seguente:

```
new fl.video.skin.PauseButtonDisabled();
```

Dopo la sezione iniziale, è presente il codice ActionScript che definisce il valore minimo della larghezza e dell'altezza dello skin. Questi valori si trovano nella finestra di dialogo Seleziona skin e vengono usati in fase di runtime per evitare che lo skin venga modificato in scala con dimensioni al di sotto dei valori minimi. Se non desiderate specificare la dimensione minima, lasciate il valore `undefined` oppure minore o uguale a zero.

```
// minimum width and height of video recommended to use this skin,  
// leave as undefined or <= 0 if there is no minimum  
this.minWidth = 270;  
this.minHeight = 60;
```

Per ogni segnaposto potete applicare le seguenti proprietà:

Proprietà	Descrizione
<code>anchorLeft</code>	Boolean. Posiziona il controllo corrispondente sul lato sinistro dell'istanza FLVPlayback. Il valore predefinito è <code>true</code> a meno che <code>anchorRight</code> non venga impostato esplicitamente su <code>true</code> , nel qual caso <code>false</code> diventa il valore predefinito.

Proprietà	Descrizione
anchorRight	Boolean. Posiziona il controllo corrispondente al lato destro dell'istanza FLVPlayback. Il valore predefinito è false.
anchorBottom	Boolean. Posiziona il controllo corrispondente alla parte inferiore dell'istanza FLVPlayback. Il valore predefinito è true, a meno che anchorTop non venga impostato esplicitamente su true, nel qual caso false diventa il valore predefinito.
anchorTop	Boolean. Posiziona il controllo corrispondente alla parte superiore dell'istanza FLVPlayback. Il valore predefinito è false.

Se le proprietà anchorLeft e anchorRight sono entrambe true, il controllo viene modificato in scala orizzontalmente in fase di runtime. Se le proprietà anchorTop e anchorBottom sono entrambe true, il controllo viene modificato in scala verticalmente in fase di runtime.

Per visualizzare gli effetti di queste proprietà, fate riferimento a come vengono usate dagli skin in Flash. I controlli BufferingBar e SeekBar sono gli unici che vengono modificati in scala, sono disposti l'uno sull'altro e hanno entrambi le proprietà anchorLeft e anchorRight impostate su true. La proprietà anchorLeft di tutti i controlli a sinistra dei componenti BufferingBar e SeekBar è impostata su true e la proprietà anchorRight di tutti i controlli a destra di tali componenti è impostata su true. La proprietà anchorBottom di tutti i controlli è impostata su true.

Potete provare a modificare i clip filmato sul livello Layout per creare uno skin in cui i controlli siano collocati nella parte superiore anziché inferiore. È sufficiente spostare i controlli verso l'alto, rispetto a video_mc, e impostare anchorTop su true per tutti i controlli.

Barra di bufferizzazione

La barra di bufferizzazione ha due clip filmato: bufferingBar_mc e bufferingBarFill_mc. È importante stabilire la posizione di ciascun clip sullo stage rispetto agli altri clip, perché le posizioni relative vengono mantenute. La barra di bufferizzazione usa due clip separati, perché il componente modifica in scala il clip bufferingBar_mc ma non il clip bufferingBarFill_mc.

Poiché al clip bufferingBar_mc viene applicata la modifica in scala a 9 porzioni, i margini non si distorcono quando viene modificato in scala. Poiché il clip bufferingBarFill_mc è molto ampio, non è necessario modificare la larghezza. In fase di runtime viene automaticamente mascherato, in modo che venga visualizzata solo la porzione al di sopra del clip allungato bufferingBar_mc. Per impostazione predefinita, le dimensioni esatte della maschera mantengono lo stesso margine verso destra e verso sinistra all'interno del clip bufferingBar_mc, in base alla differenza tra la posizione x (orizzontale) dei clip bufferingBar_mc e bufferingBarFill_mc. Potete personalizzare il posizionamento mediante il codice ActionScript.

Se la barra di bufferizzazione non deve essere modificata in scala o se non utilizza la modifica in scala a 9 porzioni, potete impostarla come il componente personalizzato BufferingBar UI FLV di riproduzione. Per ulteriori informazioni, vedete “[Componente BufferingBar](#)” a pagina 161.

La barra di bufferizzazione ha la seguente proprietà aggiuntiva:

Proprietà	Descrizione
fill_mc:MovieClip	Specifica il nome di istanza del riempimento della barra di bufferizzazione. Il valore predefinito è bufferingBarFill_mc.

Barra di ricerca e barra del volume

Anche la barra di ricerca ha due clip filmato: seekBar_mc e seekBarProgress_mc. È importante stabilire la posizione di ciascuno sul livello Layout rispetto agli altri clip, perché le posizioni relative vengono mantenute. Sebbene entrambi i clip vengano modificati in scala, il clip seekBarProgress_mc non può essere nidificato all'interno di seekBar_mc, perché seekBar_mc utilizza la modifica in scala a 9 porzioni, che non funziona bene con i clip filmato nidificati.

Poiché al clip seekBar_mc viene applicata la modifica in scala a 9 porzioni, i margini non subiscono alcuna distorsione quando viene modificato in scala. Anche il clip seekBarProgress_mc può essere modificato in scala, ma subisce delle distorsioni. Questo clip non utilizza la modifica in scala a 9 porzioni perché è un effetto di riempimento, che non crea problemi di visualizzazione in caso di distorsione.

Il clip seekBarProgress_mc funziona senza fill_mc, analogamente al modo in cui funziona il clip progress_mc nei componenti UI FLV di riproduzione personalizzati. In altre parole, non è mascherato e viene modificato in scala orizzontalmente. Le dimensioni esatte del clip seekBarProgress_mc al 100% sono definite dai margini destro e sinistro all'interno del clip seekBarProgress_mc. Per impostazione predefinita, queste dimensioni sono uguali e si basano sulla differenza tra le posizioni x (orizzontale) di seekBar_mc e seekBarProgress_mc. Potete personalizzare le dimensioni con ActionScript nel clip filmato della barra di ricerca, come mostrato nell'esempio seguente:

```
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;
```

Potete inserire questo codice sulla linea temporale del clip filmato SeekBar o insieme al resto del codice ActionScript sulla linea temporale principale. Se personalizzate il codice anziché modificare il layout, non è necessario che il riempimento si trovi nello stage, bensì solo nella libreria, impostata su Esporta per ActionScript sul fotogramma 1 e con il nome di classe corretto.

Così come accade per il componente UI FLV di riproduzione personalizzato, potete creare un clip filmato di tipo fullness per la barra di ricerca. Se non dovete modificare in scala la barra di ricerca o se è possibile modificarla in scala ma senza utilizzare la modifica in scala a 9 porzioni, potete impostare progress_mc o fullness_mc adottando uno qualsiasi dei metodi utilizzati per i componenti UI FLV di riproduzione personalizzati. Per ulteriori informazioni, vedete .

Poiché la barra del volume nello skin di Flash non può essere modificata in scala, è stata costruita allo stesso modo del componente VolumeBar UI FLV di riproduzione personalizzato. Per ulteriori informazioni, vedete “[Componenti SeekBar e VolumeBar](#)” a pagina 162. La differenza consiste nella diversa implementazione della maniglia.

Maniglie Seekbar e VolumeBar

Le maniglie SeekBar e VolumeBar sono collocate sul livello Layout vicino alla barra. Per impostazione predefinita, i valori del margine sinistro, del margine destro e dell'asse y della maniglia sono impostati secondo la sua posizione relativa al clip filmato della barra. Il margine sinistro viene impostato in base alla differenza tra la posizione x (orizzontale) della maniglia e la posizione x (orizzontale) della barra, mentre il margine destro è uguale al margine sinistro. Potete personalizzare questi valori mediante ActionScript nel clip filmato SeekBar o VolumeBar. Nell'esempio seguente viene usato lo stesso codice ActionScript dei componenti UI FLV di riproduzione personalizzati:

```
this.seekBar_mc.handleLeftMargin = 2;  
this.seekBar_mc.handleRightMargin = 2;  
this.seekBar_mc.handleY = 11;
```

Potete inserire questo codice sulla linea temporale del clip filmato SeekBar o insieme al resto del codice ActionScript sulla linea temporale principale. Se personalizzate il codice anziché modificare il layout, non è necessario che la maniglia si trovi nello stage, bensì solo nella libreria, impostata su Esporta per ActionScript sul fotogramma 1 e con il nome di classe corretto.

A parte queste proprietà, le maniglie sono dei semplici clip filmato che si impostano analogamente ai componenti UI FLV di riproduzione personalizzati. Entrambe hanno lo sfondo rettangolare con la proprietà `alpha` impostata su 0. Servono solo per aumentare l'area attiva e non sono indispensabili.

Clip di sfondo e di primo piano

I clip filmato `chrome_mc` e `forwardBackBorder_mc` vengono implementati come clip di sfondo.

Tra i clip filmato `ForwardBackBorder`, `ForwardBorder` e `BackBorder` sullo stage e i segnaposto dei pulsanti Avanti e Indietro, l'unico clip che *non* si trova su un livello guida è `ForwardBackBorder`. I pulsanti Avanti e Indietro vengono usati di fatto soltanto negli skin.

Il solo requisito per questi clip è accertarvi che siano impostati su Esporta per ActionScript sul fotogramma 1 nella libreria.

Modificare il comportamento di uno skin

Le proprietà `bufferingBarHidesAndDisablesOthers` e `skinAutoHide` consentono di personalizzare il comportamento dello skin FLVPlayback.

Se impostate la proprietà `bufferingBarHidesAndDisablesOthers` su `true`, il componente FLVPlayback nasconde SeekBar e la maniglia; inoltre quando il componente attiva lo stato di bufferizzazione, i pulsanti Riproduci e Pausa vengono disabilitati. Questa proprietà può essere utile quando scaricate un file FLV in streaming da FMS su una connessione lenta e con l'impostazione della proprietà `bufferTime` su un valore alto (ad esempio, 10). A questo punto dovete evitare di fare subito clic sui pulsanti Riproduci e Pausa, perché si potrebbe ritardare ulteriormente la riproduzione del file. Per evitare che ciò accada, mentre il componente è in stato di bufferizzazione, potete impostare `bufferingBarHidesAndDisablesOthers` su `true` e disattivare l'elemento SeekBar e i pulsanti Riproduci e Pausa.

La proprietà `skinAutoHide` interessa solo i file SWF dello skin predefinito e non i controlli creati dai componenti UI FLV di riproduzione personalizzati. Se è impostato su `true`, il componente FLVPlayback nasconde lo skin quando il mouse non si trova sull'area di visualizzazione. Il valore predefinito per questa proprietà è `true`.

Usare un file SMIL

Per gestire flussi multipli per ampiezze di banda differenti, la classe `VideoPlayer` usa una classe helper (`NCManager`) che supporta un sottoinsieme di SMIL. SMIL viene usato per identificare la posizione di un flusso video, il layout (larghezza e altezza) del file FLV e i file FLV di origine corrispondenti alle diverse ampiezze di banda. Può essere usato anche per specificare la velocità di trasferimento e la durata del file FLV.

Utilizzate il parametro `source` o la proprietà `FLVPlayback.source` (ActionScript) per specificare la posizione di un file SMIL. Per ulteriori informazioni, vedete il parametro e la proprietà `FLVPlayback.source` nella [Guida di riferimento di ActionScript 3.0 per Flash Professional](#).

Nell'esempio seguente viene mostrato un file SMIL che esegue lo streaming dei file FLV con ampiezze di banda multiple da un server FMS con connessione 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>
```

Il tag `<head>` può contenere i tag `<meta>` e `<layout>`. Il tag `<meta>` supporta solo l'attributo `base`, che viene utilizzato per specificare l'URL del video in streaming (RTMP da un server FMS).

Il tag `<layout>` supporta solo l'elemento `root-layout`, che viene utilizzato per impostare gli attributi `height` e `width` e, dunque, per determinare le dimensioni della finestra in cui avviene il rendering del file FLV. Questi attributi accettano solo i valori di pixel, non le percentuali.

Nel corpo del file SMIL potete includere un unico link al file FLV di origine oppure, se eseguite lo streaming di file multipli per ampiezze di banda differenti da un server FMS (come nell'esempio precedente), potete usare il tag `<switch>` per elencare i file di origine.

I tag `video` e `ref` all'interno del tag `<switch>` sono sinonimi, ossia possono entrambi utilizzare l'attributo `src` per specificare i file FLV. Inoltre, ciascuno di essi può utilizzare gli attributi `region`, `system-bitrate` e `dur` per specificare la regione, la larghezza di banda minima richiesta e la durata del file FLV.

All'interno del tag `<body>`, è permessa solo un'occorrenza del tag `<video>`, `<src>` o `<switch>`.

Nell'esempio seguente viene mostrato lo scaricamento progressivo di un singolo file FLV che non utilizza il rilevamento della larghezza di banda:

```
<smil>
    <head>
        <layout>
            <root-layout width="240" height="180" />
        </layout>
    </head>
    <body>
        <video src="myvideo.flv" />
    </body>
</smil>
```

<smil>

Disponibilità

Flash Professional 8.

Uso

```
<smil>  
...  
child tags  
...  
</smil>
```

Attributi

Nessuno.

Tag secondari

<head>, <body>

Tag principale

Nessuno.

Descrizione

Tag di livello più alto che identifica un file SMIL.

Esempio

Nell'esempio seguente viene mostrato un file SMIL che specifica tre file 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>

Disponibilità

Flash Professional 8.

Uso

```
<head>  
...  
child tags  
...  
</head>
```


Attributi

Nessuno.

Tag secondari

<meta>, <layout>

Tag principale

<smil>

Descrizione

Supporta i tag <meta> e <layout> e specifica la posizione e il layout predefinito (altezza e larghezza) dei file FLV di origine.

Esempio

Nell'esempio seguente viene impostato il layout root su 240 x 180 pixel:

```
<head>
  <meta base="rtmp://myserver/myapp/" />
  <layout>
    <root-layout width="240" height="180" />
  </layout>
</head>
```

<meta>

Disponibilità

Flash Professional 8.

Uso

```
<meta/>
```

Attributi

base

Tag secondari

<layout>

Tag principale

Nessuno.

Descrizione

Contiene l'attributo `base` che specifica la posizione (URL via RTMP) del file FLV di origine.

Esempio

Nell'esempio seguente viene mostrato un tag meta per la posizione base su `myserver`:

```
<meta base="rtmp://myserver/myapp/" />
```

<layout>

Disponibilità

Flash Professional 8.

Uso

```
<layout>  
...  
child tags  
...  
</layout>
```

Attributi

Nessuno.

Tag secondari

```
<root-layout>
```

Tag principale

```
<meta>
```

Descrizione

Specifica la larghezza e l'altezza del file FLV.

Esempio

Nell'esempio seguente viene specificato il layout di 240 x 180 pixel:

```
<layout>  
  <root-layout width="240" height="180" />  
</layout>
```

<root-layout>

Disponibilità

Flash Professional 8.

Uso

```
<root-layout...attributes.../>
```

Attributi

Width, height

Tag secondari

Nessuno.

Tag principale

```
<layout>
```

Descrizione

Specifica la larghezza e l'altezza del file FLV.

Esempio

Nell'esempio seguente viene specificato il layout di 240 x 180 pixel:

```
<root-layout width="240" height="180" />
```

<body>

Disponibilità

Flash Professional 8.

Uso

```
<body>  
...  
child tags  
...  
</body>
```

Attributi

Nessuno.

Tag secondari

<video>, <ref>, <switch>

Tag principale

<smil>

Descrizione

Contiene i tag <video>, <ref> e <switch>, che specificano il nome del file FLV di origine, la larghezza di banda minima e la durata del file FLV. L'attributo `system-bitrate` viene supportato solo quando utilizzate il tag <switch>. All'interno del tag <body> è permessa solo un'istanza del tag <switch><video> o <ref>.

Esempio

Nell'esempio seguente vengono specificati tre file FLV, due dei quali utilizzano il tag `video` e uno che utilizza il tag `ref`:

```
<body>  
  <switch>  
    <ref src="myvideo_cable.flv" dur="3:00.1"/>  
    <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>  
    <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1"/>  
  </switch>  
</body>
```

<video>

Disponibilità

Flash Professional 8.

Uso

```
<video...attributes.../>
```

Attributi

src, system-bitrate, dur

Tag secondari

Nessuno.

Tag principale

```
<body>
```

Descrizione

Sinonimo del tag `<ref>`. Supporta gli attributi `src` e `dur`, che specificano il nome e la durata del file FLV di origine. L'attributo `dur` supporta entrambi i formati utilizzati per indicare il tempo: il formato completo (00:03:00:01) e quello parziale (03:00:01).

Esempio

Nell'esempio seguente vengono impostate l'origine e la durata di un video:

```
<video src="myvideo_mdm.flv" dur="3:00.1"/>
```

<ref>

Disponibilità

Flash Professional 8.

Uso

```
<ref...attributes.../>
```

Attributi

src, system-bitrate, dur

Tag secondari

Nessuno.

Tag principale

```
<body>
```

Descrizione

Sinonimo del tag `<video>`. Supporta gli attributi `src` e `dur`, che specificano il nome e la durata del file FLV di origine. L'attributo `dur` supporta entrambi i formati utilizzati per indicare il tempo: il formato completo (00:03:00:01) e quello parziale (03:00:01).

Esempio

Nell'esempio seguente vengono impostate l'origine e la durata di un video:

```
<ref src="myvideo_cable.flv" dur="3:00.1"/>
```

<switch>

Disponibilità

Flash Professional 8.

Uso

```
<switch>
...
child tags
...
</switch>
```

Attributi

Nessuno.

Tag secondari

<video>, <ref>

Tag principale

<body>

Descrizione

Usato con il tag secondario <video> o <ref> per elencare i file FLV per lo streaming del video con larghezza di banda multipla. Il tag <switch> supporta l'attributo `system-bitrate`, che specifica la larghezza di banda minima e gli attributi `src` e `dur`.

Esempio

Nell'esempio seguente vengono specificati tre file FLV, due dei quali utilizzano il tag `video` e uno che utilizza il tag `ref`:

```
<switch>
  <ref src="myvideo_cable.flv" dur="3:00.1"/>
  <video src="myvideo_isdn.flv" system-bitrate="128000" dur="3:00.1"/>
  <video src="myvideo_mdm.flv" system-bitrate="56000" dur="3:00.1" />
</switch>
```

Capitolo 7: Uso del componente FLVPlaybackCaptioning

Il componente FLVPlayback consente di includere un lettore video in un'applicazione Adobe Flash CS5 Professional per riprodurre file Flash Video (FLV o F4V) scaricati e file FLV in streaming. Per ulteriori informazioni su FLVPlayback, vedete “[Uso del componente FLVPlayback](#)” a pagina 140.

Il componente FLVPlaybackCaptioning consente di includere il supporto per l'inserimento di didascalie in un video. Il componente per l'inserimento di didascalie supporta i tag Timed Text in formato XML standard di W3C e include le seguenti funzioni:

Inserimento di didascalie con cue point di evento incorporati Potete associare i cue point di evento incorporati in un file FLV con XML per supportare l'inserimento di didascalie anziché usare un file XML Timed Text.

Istanze di FLVPlaybackCaptioning multiple Potete creare più istanze di FLVPlaybackCaptioning per più istanze di FLVPlayback.

Controllo con pulsante di tipo premuto/non premuto Consente l'interazione dell'utente con la funzione di inserimento di didascalie tramite un pulsante di tipo premuto/non premuto.

Usare il componente FLVPlaybackCaptioning

Il componente FLVPlaybackCaptioning viene usato con uno o più componenti FLVPlayback. Nel caso più semplice, si trascina un componente FLVPlayback nello stage, quindi un componente FLVPlaybackCaptioning nello stesso stage, si identifica l'URL della didascalia e infine si imposta la visualizzazione delle didascalie. Inoltre, per personalizzare le didascalie del componente FLVPlayback, potete impostare diversi parametri.

Aggiungere didascalie al componente FLVPlayback

Potete aggiungere il componente FLVPlaybackCaptioning a qualsiasi componente FLVPlayback. Per informazioni sull'aggiunta dei componenti FLVPlayback all'applicazione, vedete “[Creare un'applicazione con il componente FLVPlayback](#)” a pagina 141.

Aggiungere il componente FLVPlaybackCaptioning dal pannello Componenti:

- 1 Nel pannello Componenti, aprite la cartella Video.
- 2 Trascinate il componente FLVPlaybackCaptioning (o fate doppio clic su di esso) e aggiungetelo nello stesso stage del componente FLVPlayback a cui desiderate aggiungere le didascalie.


***Nota:** Adobe offre due file di esempio per apprendere più velocemente l'uso del componente FLVPlaybackCaptioning: `caption_video.flv` (un esempio di FLVPlayback) e `caption_video.xml` (un esempio di captioning). I file sono reperibili agli indirizzi www.helpexamples.com/flash/video/caption_video.flv e www.helpexamples.com/flash/video/caption_video.xml.*

- 3 (Opzionale) Trascinate il componente CaptionButton nello stesso stage dei componenti FLVPlayback e FLVPlaybackCaptioning. Il componente CaptionButton consente all'utente di attivare e disattivare l'inserimento di didascalie.

Nota: per attivare il componente `CaptionButton`, dovete trascinarlo nello stesso stage dei componenti `FLVPlayback` e `FLVPlaybackCaptioning`.

- 4 Dopo aver selezionato il componente `FLVPlaybackCaptioning` sullo stage, specificate le seguenti informazioni obbligatorie nella scheda Parametri della finestra di ispezione Proprietà:

- Impostate `showCaptions` su `true`.
- Specificate il parametro `source` del file XML Timed Text da scaricare.

 *Mentre provate le didascalie in Flash, impostate la proprietà `showCaptions` su `true`. Tuttavia, se includete il componente `CaptionButton` per consentire agli utenti di attivare e disattivare le didascalie, impostate la proprietà `showCaptions` su `false`.*

Sono disponibili altri parametri che consentono di personalizzare il componente `FLVPlaybackCaptioning`. Per ulteriori informazioni, vedete “[Personalizzare il componente FLVPlaybackCaptioning](#)” a pagina 189 e la *Guida di riferimento di ActionScript 3.0 per Flash Professional*.

- 5 Selezionate `Controllo > Prova filmato` per avviare il video.

Creare un'istanza in modo dinamico mediante ActionScript:

- 1 Trascinate il componente `FLVPlayback` dal pannello Componenti alla libreria (Finestra > Libreria).
- 2 Trascinate il componente `FLVPlaybackCaptioning` dal pannello Componenti al pannello Libreria.
- 3 Aggiungere il codice seguente nel pannello Azioni al fotogramma 1 nella linea temporale.

```
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 Sostituite `install_drive` con l'unità in cui avete installato Flash e modificate il percorso in modo che rifletta la posizione della cartella `Skins` nell'installazione locale:

Nota: se create un'istanza `FLVPlayback` con ActionScript, dovete assegnarle uno skin mediante l'impostazione in modo dinamico della proprietà `skin` con ActionScript. Uno skin applicato con ActionScript non viene pubblicato automaticamente con il file SWF. Copiate il file SWF dello skin e il file SWF dell'applicazione sul server, altrimenti il file SWF dello skin non sarà disponibile quando viene eseguito dall'utente.

Impostare i parametri del componente FLVPlaybackCaptioning

Per ogni istanza del componente `FLVPlaybackCaptioning` potete impostare i seguenti parametri nella finestra di ispezione Proprietà o nella finestra di ispezione dei componenti per personalizzare ulteriormente il componente. Nell'elenco seguente sono identificate le proprietà e viene fornita una breve spiegazione:

autoLayout Determina se il componente `FLVPlaybackCaptioning` controlla le dimensioni dell'area delle didascalie. Il valore predefinito è `true`.

captionTargetName Identifica il nome dell'istanza TextField o di MovieClip che contiene le didascalie. Il valore predefinito è `auto`.

flvPlaybackName Identifica il nome dell'istanza FLVPlayback a cui desiderate aggiungere delle didascalie. Il valore predefinito è `auto`.

simpleFormatting Quando è impostata su `true`, limita le istruzioni di formattazione del file XML Timed Text. Il valore predefinito è `false`.

showCaptions Determina la visualizzazione delle didascalie. Il valore predefinito è `true`.

source Identifica il percorso del file XML Timed Text.

Per ulteriori informazioni su tutti i parametri di FLVPlaybackCaptioning, vedete la *Guida di riferimento di ActionScript 3.0 per Flash Professional*.

Specificare il parametro source di

Il parametro `source` consente di specificare il nome e il percorso del file XML Timed Text che contiene le didascalie per il filmato. Inserite il percorso URL direttamente nella cella `source` nella finestra di ispezione dei componenti.

Visualizzare le didascalie

Per visualizzare le didascalie, impostate il parametro `showCaptions` su `true`.

Per ulteriori informazioni su tutti i parametri del componente FLVPlaybackCaptioning, vedete la *Guida di riferimento di ActionScript 3.0 per Flash Professional*.

Negli esempi precedenti avete appreso come creare e attivare il componente FLVPlaybackCaptioning per visualizzare le didascalie. Potete utilizzare due origini per le didascalie: (1) un file XML Timed Text contenente le didascalie oppure (2) un XML con il testo delle didascalie da associare con i cue point di eventi incorporati.

Usare le didascalie Timed Text

Il componente FLVPlaybackCaptioning consente di inserire le didascalie per il componente FLVPlayback associato scaricando un file XML Timed Text (TT). Per ulteriori informazioni sul formato Timed Text, vedete le relative informazioni disponibili nel sito <http://www.w3.org>.

In questa sezione viene fornita una panoramica dei tag Timed Text supportati, dei tag dei file di didascalie obbligatori e un esempio di file XML Timed Text. Per informazioni dettagliate su tutti i tag Timed Text supportati, vedete “[Tag Timed Text](#)” a pagina 182.

Il componente FLVPlaybackCaptioning supporta i seguenti tag Timed Text:

Categoria	Attività
Supporto per la formattazione del paragrafo	Allineare un paragrafo a destra, a sinistra o al centro
Supporto per la formattazione del testo	<ul style="list-style-type: none">• Impostare le dimensioni del testo con dimensioni di pixel assolute o con lo stile delta (ad esempio, +2, -4)• Impostare il carattere e il colore del testo• Applicare il formato grassetto e corsivo al testo• Impostare la giustificazione del testo
Altro tipo di supporto per la formattazione	<ul style="list-style-type: none">• Impostare il colore di sfondo del componente TextField per le didascalie• Impostare il colore di sfondo del componente TextField per le didascalie su trasparente (alfa 0)• Impostare il ritorno a capo automatico del componente TextField per le didascalie (attivato/disattivato)

Il componente FLVPlaybackCaptioning corrisponde al codice temporale del file FLV. Ogni didascalia deve avere un attributo `begin` che determina il momento in cui deve essere visualizzata la didascalia. Se alla didascalia non è associato un attributo `dur` o `end`, la didascalia scompare quando viene visualizzata quella successiva o quando termina il file FLV.

Di seguito è riportato un esempio di file XML Timed Text. Questo file (`caption_video.xml`) include le didascalie per il file `caption_video.flv`. I file sono reperibili agli indirizzi www.helpexamples.com/flash/video/caption_video.flv e www.helpexamples.com/flash/video/caption_video.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
  <tt xml:lang="en"
xmlns="http://www.w3.org/2006/04/ttaf1"xmlns:tts="http://www.w3.org/2006/04/ttaf1#styling">
<head>
  <styling>
<style id="1" tts:textAlign="right"/>
<style id="2" tts:color="transparent"/>
<style id="3" style="2" tts:backgroundColor="white"/>
<style id="4" style="2 3" tts:fontSize="20"/>
  </styling>
</head>
<body>
  <div xml:lang="en">
<p begin="00:00:00.00" dur="00:00:03.07">I had just joined <span
tts:fontFamily="monospaceSansSerif,proportionalSerif,TheOther"tts:fontSize="+2">Macromedia</span> in 1996,</p>
<p begin="00:00:03.07" dur="00:00:03.35">and we were trying to figure out what to do about the internet.</p>
<p begin="00:00:06.42" dur="00:00:03.15">And the company was in dire straights at the time.</p>
<p begin="00:00:09.57" dur="00:00:01.45">We were a CD-ROM authoring company,</p>
<p begin="00:00:11.42" dur="00:00:02.00">and the CD-ROM business was going away.</p>
<p begin="00:00:13.57" dur="00:00:02.50">One of the technologies I remember seeing was Flash.</p>
<p begin="00:00:16.47" dur="00:00:02.00">At the time, it was called <span
tts:fontWeight="bold" tts:color="#ccc333">FutureSplash</span>.</p>
<p begin="00:00:18.50" dur="00:00:01.20">So this is where Flash got its start.</p>
<p begin="00:00:20.10" dur="00:00:03.00">This is smart sketch running on the <span
tts:fontStyle="italic">EU-pin computer</span>,</p>
<p begin="00:00:23.52" dur="00:00:02.00">which was the first product that FutureWave did.</p>
<p begin="00:00:25.52" dur="00:00:02.00">So our vision for this product was to</p>
<p begin="00:00:27.52" dur="00:00:01.10">make drawing on the computer</p>
<p begin="00:00:29.02" dur="00:00:01.30" style="1">as <span tts:color="#ccc333">easy</span>
as drawing on paper.</p>
</div>
</body>
</tt>
```

Tag Timed Text

Il componente FLVPlaybackCaptioning supporta i tag Timed Text per l'inserimento di didascalie nei file XML. Per ulteriori informazioni sui tag Timed Text audio e video, vedete le informazioni disponibili nel sito <http://www.w3.org>. Nella tabella seguente sono elencati i tag supportati e non supportati.

Function	Tag/Valore	Uso/Descrizione	Esempio
Tag ignorati	metadati	Ignorato/consentito a qualsiasi livello del documento	
	set	Ignorato/consentito a qualsiasi livello del documento	
	xml:lang	Ignorato	
	xml:space	Ignorato/il comportamento viene sostituito da: xml:space="default"	

Function	Tag/Valore	Uso/Descrizione	Esempio
	layout	Ignorato/compresi gli eventuali tag region in una sezione di tag layout	
	tag br	Tutti gli attributi e il contenuto vengono ignorati.	
Sincronizza il contenuto multimediale per le didascalie	attributi begin	Consentiti solo nei tag p. Obbligatori per la distribuzione temporizzata delle didascalie in base al contenuto multimediale.	<p begin="3s">
	attributi dur	Consentiti solo nei tag p. Consigliati. Se non sono inclusi, la didascalia termina con il file FLV o quando inizia un'altra didascalia.	
	attributi end	Consentiti solo nei tag p. Consigliati. Se non sono inclusi, la didascalia termina con il file FLV o quando inizia un'altra didascalia.	
Sincronizza l'orario per le didascalie	00:03:00.1	Formato dell'ora completo	
	03:00.1	Formato dell'ora parziale	
	10	Ore offset senza unità di misura. Offset rappresenta i secondi.	
	00:03:00:05 00:03:00:05.1 30f 30t	non supportato. I formati dell'ora che includono frame o tacche non sono supportati.	
Tag body	body	Obbligatorio/È supportato un solo tag body.	<body><div>...</div></body>
Tag content	tag div	Ne sono consentiti zero o più. Viene utilizzato il primo tag.	
	tag p	Ne sono consentiti zero o più.	
	tag span	Contenitore logico per una sequenza di unità di contenuto testuale. Non supportato per le estensioni nidificate. Supportato per i tag style di attributi.	
	tag br	Denota un'interruzione di riga esplicita.	
Tag styling (Tutti i tag style vengono usati all'interno del tag p)	style	Riferimento a uno o più elementi di stile. Può essere usato come tag e come attributo. Come tag, è necessario un attributo ID (style può essere riutilizzato nel documento). All'interno del tag style sono supportati uno o più tag style.	
	tts:background Color	Consente di specificare una proprietà di stile per definire il colore di sfondo di una regione. Il valore alfa viene ignorato, a meno che sia impostato su zero (alfa 0) per rendere lo sfondo trasparente. Il formato di colore è #RRGGBBAA.	

Function	Tag/Valore	Uso/Descrizione	Esempio
	tts:color	Consente di specificare una proprietà di stile per definire il colore di primo piano. Il valore alfa non è supportato per tutti i colori. Il valore <code>transparent</code> viene convertito in nero.	<pre><style id="3" style="2" tts:backgroundColor="white"/> "transparent" = #00000000 "black"=#000000FF "silver"=#C0C0C0FF "grey"=#808080FF "white"=#FFFFFFFF "maroon"=#800000FF "red"=#FF0000FF "purple"=#800080FF "fuchsia"("magenta")= #FF00FFFF "green"=#008000FF "lime"=#00FF00FF "olive"=#808000FF "yellow"=#FFFF00FF "navy"=#000080FF "blue"=#0000FFFF "teal"=#008080FF "aqua"("cyan")=#00FFFFFF</pre>
	tts:fontFamily	Consente di specificare una proprietà di stile per definire la famiglia di caratteri.	<pre>"default" = _serif "monospace" = _typewriter "sansSerif" = _sans "serif" = _serif "monospaceSansSerif" = _typewriter "monospaceSerif" = _typewriter "proportionalSansSerif" = _sans</pre>
	tts:fontSize	Consente di specificare una proprietà di stile per definire le dimensioni del carattere. Solo il primo valore (vertical) viene utilizzato se ne vengono forniti due. I valori percentuali e le unità di misura vengono ignorati. Sono supportati sia dimensioni di pixel assolute (ad esempio, 12) che dimensioni di stile relative (ad esempio, +2).	
	tts:fontStyle	Consente di specificare una proprietà di stile per definire lo stile del carattere.	<pre>"normal" "italic" "inherit"* * Comportamento predefinito; eredita lo stile dal tag che lo contiene.</pre>

Function	Tag/Valore	Uso/Descrizione	Esempio
	tts:fontWeight	Consente di specificare una proprietà di stile per definire lo spessore del carattere.	"normal" "bold" "inherit"* * Comportamento predefinito; eredita lo stile dal tag che lo contiene.
	tts:textAlign	Consente di specificare una proprietà di stile per definire l'allineamento delle aree incorporate all'interno di un'area del blocco che le contiene.	"left" "destra" "center" "start" ("left") "end" ("right") "inherit"* *Eredita lo stile dal tag che lo contiene. Se non è impostato un tag textAlign, il valore predefinito è "left".
	tts:wrapOption	Consente di specificare una proprietà di stile per definire se viene applicato o meno il ritorno a capo automatico (interruzione di riga) nel contesto dell'elemento interessato. Questa impostazione ha effetto su tutti i paragrafi contenuti nell'elemento didascalia.	"wrap" "noWrap" "inherit"* *Eredita lo stile dal tag che lo contiene. Se non è impostato un tag wrapOption, il valore predefinito è "wrap".
Attributi non supportati	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		

Usare i cue point con le didascalie

I cue point consentono di interagire con un video; ad esempio, potete influire sulla riproduzione di un file FLV oppure visualizzare il testo in punti specifici del video. Se non è disponibile un file XML Timed Text da usare con un file FLV, potete incorporare dei cue point di evento in un file FLV, quindi associare questi cue point al testo. In questa sezione vengono fornite informazioni sugli standard relativi ai cue point del componente FLVPlaybackCaptioning, oltre a una breve panoramica sull'associazione di tali cue al testo per l'inserimento di didascalie. Per ulteriori informazioni su come incorporare i cue point di evento con la procedura guidata Importa video o con Flash Video Encoder, vedete il Capitolo 16, "Operazioni con i file video", nella guida *Uso di Flash*.

Nozioni fondamentali sugli standard relativi ai cue point del componente FLVPlaybackCaptioning

All'interno dei metadati del file FLV, un cue point è rappresentato sotto forma di un oggetto con le proprietà seguenti: `name`, `time`, `type` e `parameters`. I cue point ActionScript di FLVPlaybackCaptioning presentano i seguenti attributi:

name La proprietà `name` è una stringa che contiene il nome assegnato al cue point. La proprietà `name` deve iniziare con il prefisso `fl.video.caption.2.0.` a cui deve seguire una stringa. La stringa è costituita da una serie di numeri interi positivi che viene incrementata ogni volta per assicurare che i nomi siano univoci. Il prefisso include il numero di versione, che corrisponde anche al numero di versione di FLVPlayback. Per Adobe Flash CS4 e versioni successive, dovete impostare il numero di versione su `2.0`.

time La proprietà `time` corrisponde alla posizione temporale in cui deve essere visualizzata la didascalia.

type La proprietà `type` è una stringa il cui valore è `"event"`.

parameters La proprietà `parameters` è un array che supporta le seguenti coppie nome/valore:

- **text:String** Il testo in formato HTML per la didascalia. Questo testo viene passato direttamente alla proprietà `TextField.htmlText`. Il componente FLVPlaybackCaptioning supporta una proprietà `text:n` opzionale, che consente l'uso di tracce in più lingue. Per ulteriori informazioni, vedete ["Supportare tracce in più lingue con cue point incorporati"](#) a pagina 188.
- **endTime:Number** La posizione temporale in cui la didascalia deve essere nascosta. Se non specificate questa proprietà, viene considerata dal componente FLVPlaybackCaptioning come NaN (not a number) e la didascalia continua a essere visualizzata fino al termine del file FLV (ovvero quando l'istanza FLVPlayback invia l'evento `VideoEvent.COMPLETE`). Specificate la proprietà `endTime:Number` in secondi.
- **backgroundColor:uint** Questo parametro imposta la proprietà `TextField.backgroundColor`. Questa proprietà è opzionale.
- **backgroundColorAlpha:Boolean** Se per `backgroundColor` è impostato un valore alfa 0%, il parametro imposta `TextField.background = !backgroundColor`. Questa proprietà è opzionale.
- **wrapOption:Boolean** Questo parametro imposta la proprietà `TextField.wordWrap`. Questa proprietà è opzionale.

Nozioni fondamentali sulle didascalie per i cue point di evento incorporati

Se non disponete di un file XML Timed Text contenente le didascalie per il file FLV, potete creare le didascalie associandole a un file XML contenente didascalie con i cue point di eventi incorporati. Nel file XML di esempio si presuppone che abbiate eseguito la procedura riportata di seguito per creare i cue point di evento incorporati nel video:

- Aggiungete i cue point di evento (seguendo gli standard di FLVPlaybackCaptioning) ed eseguite la codifica del video.

- In Flash, trascinate nello stage un componente FLVPlayback e un componente FLVPlaybackCaptioning.
- Impostate le proprietà source dei componenti FLVPlayback e FLVPlaybackCaptioning (il percorso del file FLV e il percorso del file XML).
- Procedete alla pubblicazione.

Nell'esempio riportato di seguito viene importato il file XML nell'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>
```

Il componente FLVPlaybackCaptioning supporta inoltre l'uso di tracce in più lingue con cue point incorporati. Per ulteriori informazioni, vedete [“Supportare tracce in più lingue con cue point incorporati”](#) a pagina 188.

Supportare tracce in più lingue con cue point incorporati

La proprietà `track` del componente FLVPlaybackCaptioning supporta l'uso di tracce in più lingue con cue point incorporati, a condizione che il file XML Timed Text segua gli standard relativi ai cue point di FLVPlaybackCaptioning. (Per ulteriori informazioni, vedete [“Nozioni fondamentali sugli standard relativi ai cue point del componente FLVPlaybackCaptioning”](#) a pagina 186.) Tuttavia, il componente FLVPlaybackCaptioning non supporta l'uso di tracce in più lingue in file XML separati. Per usare la proprietà `track`, impostatela su un valore non uguale a 0. Ad esempio, se impostate la proprietà `track` su 1 (`track == 1`), il componente FLVPlaybackCaptioning cercherà i parametri cue point. Se non viene trovata una corrispondenza, nei parametri cue point viene usata la proprietà `text`. Per ulteriori informazioni, vedete la proprietà `track` nella *Guida di riferimento di ActionScript 3.0 per Flash Professional*.

Riprodurre più file FLV con didascalie

Potete aprire più lettori video all'interno di una singola istanza del componente FLVPlayback, per riprodurre più video e passare da un video all'altro nel corso della riproduzione. Potete inoltre associare le didascalie con ciascun lettore video all'interno del componente FLVPlayback. Per ulteriori informazioni sull'apertura di lettori video multipli, vedete [“Usare lettori video multipli”](#) a pagina 156. Per usare le didascalie in più lettori video, create un'istanza del componente FLVPlaybackCaptioning per ciascun `VideoPlayer` e impostate `videoPlayerIndex` del componente FLVPlaybackCaptioning sull'indice corrispondente. Quando esiste un solo `VideoPlayer`, il relativo indice predefinito è 0.

Nel seguente esempio di codice vengono assegnate didascalie univoche a video univoci. Per provare questo esempio, gli URL fittizi devono essere sostituiti con URL funzionanti.


```
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");
```

Personalizzare il componente FLVPlaybackCaptioning

Per usare immediatamente il componente FLVPlaybackCaptioning, potete scegliere di mantenerne le impostazioni predefinite che inseriscono le didascalie direttamente sul componente FLVPlayback. Il componente FLVPlaybackCaptioning può essere personalizzato in modo che le didascalie siano all'esterno del video.

Il codice seguente illustra come creare in modo dinamico un oggetto FLVPlayback con un pulsante di tipo premuto/non premuto per l'inserimento delle didascalie:

- 1 Collocate il componente FLVPlayback sullo stage nella posizione 0,0 e assegnate il nome di istanza **player**.
- 2 Collocate il componente FLVPlaybackCaptioning sullo stage in posizione 0,0 e assegnate il nome di istanza **captioning**.
- 3 Posizionare il componente CaptionButton sullo stage.
- 4 Nell'esempio seguente, impostate la variabile `testVideoPath:String` su un file FLV (specificando un percorso assoluto o relativo).

***Nota:** l'esempio di codice imposta la variabile `testVideoPath` sul video Flash di esempio, `caption_video.flv`. Modificate questa variabile specificando il percorso del componente video delle didascalie al quale state aggiungendo un componente pulsante per le didascalie.*

- 5 Nell'esempio seguente, impostate la variabile `testCaptioningPath:String` su un file XML Timed Text appropriato (specificando un percorso assoluto o relativo).

***Nota:** l'esempio di codice imposta la variabile `testCaptioningPath` sul file XML Timed Text, `caption_video.xml`. Modificate questa variabile specificando il percorso del file XML Timed Text che contiene le didascalie per il video.*

- 6 Salvate il codice seguente come `FLVPlaybackCaptioningExample.as` nella stessa directory del file FLA.
- 7 Impostate `DocumentClass` nel file FLA su `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;
        }
    }
}
```

Per ulteriori informazioni su tutti i parametri di FLVPlaybackCaptioning, vedete la *Guida di riferimento di ActionScript 3.0 per Flash Professional*.