

APPRENDIMENTO DI ACTIONSCRIPT™ PER FLASH® LITE™ 1.x

© 2007 Adobe Systems Incorporated. Tutti i diritti riservati.

Apprendimento di ActionScript™ per Flash® Lite™ 1.x

Se la presente guida viene distribuita con software che include un accordo di licenza per l'utente finale, la guida e il software in essa descritto sono concessi in licenza e possono essere usati e copiati solo in conformità con i termini di tale licenza. Ad eccezione di quanto eventualmente concesso da tale licenza, nessuna parte di questa guida può essere riprodotta, memorizzata in un sistema per il recupero dati o trasmessa in qualsiasi forma o con qualsiasi mezzo, elettronico, meccanico, di registrazione o altro, senza il previo consenso scritto da parte di Adobe Systems Incorporated. Il contenuto di questa guida è protetto dalle leggi sui diritti d'autore, anche se non distribuito con software corredato di accordo di licenza per l'utente finale.

Il contenuto di questa guida viene fornito unicamente a scopo informativo, è soggetto a modifiche senza preavviso e non comporta alcun impegno per Adobe Systems Incorporated. Adobe Systems Incorporated declina ogni responsabilità per eventuali errori o imprecisioni presenti in questa guida.

Se si inseriscono in un progetto grafica e immagini esistenti, si tenga presente che tali materiali potrebbero essere protetti dalla legge sul copyright. L'inserimento non autorizzato di tali materiali nel proprio lavoro potrebbe rappresentare una violazione dei diritti del titolare del copyright. Assicurarsi sempre di ottenere le eventuali autorizzazioni necessarie dal titolare dei diritti d'autore.

Tutti i riferimenti a nomi di società negli esempi forniti hanno scopo puramente dimostrativo e non intendono fare riferimento ad alcuna organizzazione realmente esistente.

Adobe, il logo Adobe, Flash Lite e Flash sono marchi registrati o marchi commerciali di Adobe Systems Incorporated negli Stati Uniti e/o in altri Paesi.

Informazioni di terze parti

Questo manuale contiene collegamenti a siti Web di terze parti che non sono sotto il controllo di Adobe Systems Incorporated. Adobe Systems Incorporated non potrà quindi essere ritenuta responsabile per il contenuto di qualsiasi sito collegato. Qualora si decida di accedere a un sito Web di terze parti citato nel presente documento, lo si farà sotto la propria completa responsabilità e a proprio rischio. Adobe Systems Incorporated fornisce questi collegamenti solo per comodità dell'utente e l'inclusione del collegamento non implica che Adobe Systems Incorporated sottoscriva o accetti qualsiasi responsabilità per il contenuto di tali siti di terze parti.



Tecnologia per la compressione e la decompressione video Sorenson™ Spark™, concessa in licenza da Sorenson Media, Inc.

Fraunhofer-IIS/Thomson Multimedia: tecnologia di compressione audio MPEG Layer-3 concessa in licenza da Fraunhofer IIS e Thomson Multimedia (<http://www.iis.fhg.de/amm/>).

Independent JPEG Group: questo software è parzialmente basato sul lavoro di Independent JPEG Group.

Nellymoser, Inc.: tecnologia per la compressione e la decompressione vocale concessa in licenza da Nellymoser, Inc. (<http://www.nellymoser.com>).

Browser Opera ® Copyright © 1995-2002 di Opera Software ASA e dei suoi fornitori. Tutti i diritti riservati.

Macromedia Flash 8 video è alimentato mediante tecnologia video On2 TrueMotion. © 1992-2005 On2 Technologies, Inc. Tutti i diritti riservati. <http://www.on2.com>.

Visual SourceSafe è un marchio registrato o un marchio commerciale di Microsoft Corporation negli Stati Uniti e/o in altri paesi.

Informazioni aggiornate e informazioni aggiuntive sul codice di terze parti sono disponibili all'indirizzo http://www.adobe.com/go/thirdparty_it/.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Avviso agli utenti finali di enti governativi degli Stati Uniti d'America: il Software e la Documentazione sono "Commercial Items" (Prodotti commerciali) secondo la definizione contenuta nell'articolo 48 C.F.R. §2.101, costituiti da "Commercial

Computer Software” (Software commerciale per Computer) e “Commercial Computer Software Documentation” (Documentazione relativa a software commerciale per Computer) secondo la definizione contenuta nell’articolo 48 C.F.R. §12.212 o 48 C.F.R. §227.7202, secondo i casi. In conformità con l’articolo 48 C.F.R. §12.212 o con gli articoli da 48 C.F.R. §§227.7202-1 a 227.7202-4 incluso, secondo i casi, i “Commercial Computer Software” e “Commercial Computer Software Documentation” vengono concessi in licenza agli utenti appartenenti al Governo degli Stati Uniti d’America (a) esclusivamente come “Commercial Items” e (b) con i soli diritti concessi a tutti gli altri utenti finali ai termini e alle condizioni qui contenuti. Tutti i diritti non pubblicati riservati, ai sensi della legge sul diritto d’autore vigente negli Stati Uniti d’America. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. Nei confronti degli utenti finali del Governo degli Stati Uniti, Adobe accetta di rispettare tutte le leggi applicabili sul diritto alle pari opportunità, comprese, ove applicabili, le direttive dell’Executive Order 11246, secondo revisione, la sezione 402 del “Vietnam Era Veterans Readjustment Assistance Act” del 1974 (38 USC 4212) e la sezione 503 del “Rehabilitation Act” del 1973, secondo revisione, oltre ai regolamenti esposti in 41 CFR da 60-1 a 60-60, 60-250 e 60-741. La clausola di azione affermativa e i regolamenti sopra elencati saranno incorporati tramite riferimento.

Indice

Capitolo 1: ActionScript per Flash Lite 1.x	7
Panoramica di ActionScript per Flash Lite 1.x	7
Differenze tra ActionScript per Flash Lite 1.0 e ActionScript per Flash Lite 1.1	8
Funzioni di ActionScript per Flash 4 non supportate da ActionScript per Flash Lite 1.x	9
Funzioni non disponibili in ActionScript per Flash Lite 1.x	9
Capitolo 2: Introduzione a ActionScript per Flash 4	11
Come ottenere e impostare le proprietà dei clip filmato	11
Controllo di altre linee temporali	12
Uso delle variabili	13
Emulazione degli array	13
Operazioni con il testo e con le stringhe	14
Uso della funzione call() per la creazione di funzioni	15
Uso della funzione eval()	20
Capitolo 3: Operazioni comuni di creazione di script	21
Come determinare le funzionalità del dispositivo e della piattaforma	21
Apertura di una pagina Web	22
Avvio di una telefonata	23
Invio di un messaggio di testo o multimediale	23
Invio di un messaggio e-mail	24
Caricamento di file SWF esterni	24
Caricamento di dati esterni	25
Indice analitico	29

ActionScript per Flash Lite 1.x

Il linguaggio ActionScript permette di aggiungere logica di programmazione e interattività alle applicazioni Adobe® Flash® Lite™. La versione di ActionScript per il software Macromedia® Flash® Lite™ 1.0 e Macromedia® Flash® Lite™ 1.1 di Adobe (definita globalmente ActionScript per Flash Lite 1.x) è un ibrido tra il linguaggio ActionScript per Macromedia® Flash® 4 di Adobe e una serie di comandi aggiuntivi e di proprietà specifiche del lettore Flash Lite, quali la possibilità di avviare telefonate o di inviare messaggi di testo oppure di interrogare il dispositivo per ottenere l'ora e la data.

In questo capitolo sono presentati i seguenti argomenti:

Panoramica di ActionScript per Flash Lite 1.x	7
Differenze tra ActionScript per Flash Lite 1.0 e ActionScript per Flash Lite 1.1	8
Funzioni di ActionScript per Flash 4 non supportate da ActionScript per Flash Lite 1.x	9
Funzioni non disponibili in ActionScript per Flash Lite 1.x	9

Panoramica di ActionScript per Flash Lite 1.x

ActionScript per Flash Lite 1.x è costituito dai seguenti componenti:

ActionScript per Flash Player 4 Operatori (ad esempio, operatori di confronto e di assegnazione), proprietà dei clip filmato (ad esempio, `_height`, `_x` e `_y`), funzioni di controllo della linea temporale (ad esempio, `gotoAndPlay()` o `stop()`) e funzionalità di rete quali `loadVariables()` e `loadMovie()` (solo per Flash Lite 1.1). Per un elenco degli elementi di ActionScript per Flash 4 non supportati, vedere “[Funzioni di ActionScript per Flash 4 non supportate da ActionScript per Flash Lite 1.x](#)” a pagina 9.

Comandi e proprietà di integrazione con il telefono Flash Lite è dotato di comandi che consentono, ad esempio, di interrogare il dispositivo per ottenere la data e l'ora correnti, di avviare una telefonata o inviare un SMS o di lanciare applicazioni esterne installate nel dispositivo.

Variabili relative alle funzionalità della piattaforma (solo Flash Lite 1.1) Queste proprietà forniscono informazioni sulle funzionalità del dispositivo o dell'ambiente Flash Lite in fase di runtime. Ad esempio, la variabile `_capLoadData` indica se l'applicazione è in grado di caricare dati in rete.

Funzione `fscommand2()` Come la funzione `fscommand()`, `fscommand2()` si utilizza per comunicare con l'ambiente o con il sistema host (in questo caso, con il cellulare o il dispositivo mobile). La funzione `fscommand2()`, una versione migliorata di `fscommand()`, è in grado tra l'altro di passare un numero arbitrario di argomenti e di recuperare immediatamente i valori restituiti (senza dover attendere il fotogramma successivo, come accade con `fscommand()`).

Differenze tra ActionScript per Flash Lite 1.0 e ActionScript per Flash Lite 1.1

Le seguenti funzioni di ActionScript per Flash Lite 1.1 non sono disponibili in Flash Lite 1.0:

- Accesso alla rete o informazioni sullo stato della rete. Ad esempio, in Flash Lite 1.0 non è possibile utilizzare le funzioni `loadVariables()` o `loadMovie()` per caricare dati esterni o file SWF, né i vari comandi `fscommand2()` per determinare il livello di potenza del segnale di connessione o lo stato di una richiesta alla rete.
- Interrogazione del dispositivo per ottenere la data e l'ora.
- Variabili relative alle funzionalità della piattaforma, che forniscono informazioni sulle funzionalità della piattaforma Flash Lite e del dispositivo.
- La funzione `fscommand2()` e i relativi comandi quali `SetSoftKeys` e `FullScreen`.
- Le proprietà dei campi di testo `scroll` e `maxscroll`.

Funzioni di ActionScript per Flash 4 non supportate da ActionScript per Flash Lite 1.x

ActionScript per Flash Lite 1.x non supporta o supporta solo parzialmente le seguenti funzioni di ActionScript per Flash 4:

- Le funzioni `startDrag()` e `stopDrag()`.
- ActionScript per Flash Lite 1.x supporta un sottoinsieme degli eventi associati ai pulsanti supportati in Flash Player 4. Per ulteriori informazioni sulla gestione degli eventi associati ai pulsanti, vedere Capitolo 1, “Creazione di contenuto interattivo e navigazione”, in *Sviluppo di applicazioni Flash Lite*.
- ActionScript per Flash Lite 1.x supporta un sottoinsieme degli eventi di tastiera supportati in Flash Player 4. Per ulteriori informazioni sugli eventi di tastiera supportati in Flash Lite, vedere Capitolo 1, “Creazione di contenuto interattivo e navigazione”, in *Sviluppo di applicazioni Flash Lite*.
- La proprietà `_dropTarget`.
- La proprietà `_soundBufTime`.
- La proprietà `_url`.
- La funzione di conversione `String()`.

Funzioni non disponibili in ActionScript per Flash Lite 1.x

Poiché si basa su una versione precedente di Flash Player, il lettore Flash Lite non supporta tutte le funzioni di programmazione disponibili nelle versioni più recenti di Flash Player o negli altri linguaggi di programmazione eventualmente utilizzati. Questa sezione descrive le funzioni di programmazione non disponibili in ActionScript per Flash Lite 1.x e le soluzioni alternative disponibili.

Funzioni definite dall'utente Flash Lite 1.x non supporta la definizione e la chiamata di funzioni personalizzate. È tuttavia possibile utilizzare la funzione `call()` per eseguire il codice che si trova su un fotogramma specifico della linea temporale. Per ulteriori informazioni, vedere [“Uso della funzione call\(\) per la creazione di funzioni” a pagina 15](#).

Array, oggetti o altri tipi di dati complessi nativi Flash Lite 1.x non supporta le strutture dati di array native o altri tipi di dati completi. È possibile tuttavia emulare gli array mediante pseudo-array, una tecnica che prevede l'uso della funzione `eval()` per valutare stringhe concatenate in modo dinamico. Per ulteriori informazioni, vedere [“Emulazione degli array” a pagina 13](#).

Caricamento di file audio o immagine esterni in fase di runtime A differenza della versione desktop di Flash Player, ActionScript per Flash Lite 1.x non consente di caricare file JPEG o MP3 esterni. In Flash Lite 1.1 è possibile utilizzare la funzione `loadMovie()` per caricare i file SWF esterni. Per ulteriori informazioni, vedere [“Caricamento di file SWF esterni” a pagina 24](#).

Introduzione a ActionScript per Flash 4

Il linguaggio ActionScript per Macromedia Flash Lite 1.x di Adobe si basa sulla versione di ActionScript introdotta con Macromedia Flash Player 4 di Adobe. Di conseguenza, numerose funzioni di programmazione disponibili nelle versioni successive di Flash Player (per sistemi desktop) non sono invece disponibili per le applicazioni Flash Lite 1.x.

Se si ha scarsa familiarità con la sintassi e le funzioni di ActionScript per Flash 4 o se non si ricordano determinati particolari delle applicazioni Flash sviluppate in precedenza, questo capitolo contiene un'introduzione all'uso del linguaggio ActionScript per Flash 4 nelle applicazioni Flash Lite.

In questo capitolo sono presentati i seguenti argomenti:

Come ottenere e impostare le proprietà dei clip filmato	11
Controllo di altre linee temporali	12
Uso delle variabili	13
Emulazione degli array	13
Operazioni con il testo e con le stringhe	14
Uso della funzione call() per la creazione di funzioni	15
Uso della funzione eval()	20

Come ottenere e impostare le proprietà dei clip filmato

Per ottenere o impostare una proprietà di un clip filmato (se è possibile impostarla) si può utilizzare la sintassi del punto o la funzione `setProperty()` o `getProperty()`.

È possibile utilizzare anche la funzione `tellTarget()`.

Per utilizzare la sintassi del punto, specificare il nome di istanza del clip filmato seguito da un punto (.) e quindi il nome della proprietà. Ad esempio, il codice seguente ottiene la coordinata *x* dello schermo (rappresentata dalla proprietà `_x` del clip filmato) del clip filmato `cartoonArea` e assegna il risultato a una variabile denominata `x_pos`.

```
x_pos = cartoonArea._x;
```

L'esempio seguente è analogo al precedente ma utilizza la funzione `getProperty()` per recuperare la posizione *x* del clip filmato:

```
x_pos = getProperty(cartoonArea, _x);
```

La funzione `setProperty()` consente di impostare una proprietà di un'istanza di clip filmato, come mostra l'esempio seguente:

```
setProperty(cartoonArea, _x, 100);
```

L'esempio seguente è analogo al precedente, ma utilizza la sintassi del punto:

```
cartoonArea._x = 100;
```

Le proprietà dei clip filmato si possono ottenere o impostare anche all'interno di un'istruzione `tellTarget()`. Il codice seguente è analogo a quello dell'esempio della funzione `setProperty()` illustrato sopra.

```
tellTarget("/cartoonArea") {  
    _x = 100;  
}
```

Per ulteriori informazioni sulla funzione `tellTarget()`, vedere [“Controllo di altre linee temporali” a pagina 12](#).

Controllo di altre linee temporali

Per specificare il percorso di una linea temporale, creare il riferimento al percorso con la sintassi della barra (/) abbinata ai punti (.). È possibile anche utilizzare la notazione di Flash 5 `_levelN`, `_root` o `_parent` per fare riferimento rispettivamente a un livello specifico del filmato, alla linea temporale principale dell'applicazione o alla linea temporale di livello superiore.

Ad esempio, si supponga che nella linea temporale principale del file SWF sia presente un'istanza di clip filmato denominata `box`, che a sua volta contiene un'altra istanza di clip filmato denominata `cards`. Gli esempi seguenti fanno riferimento al clip filmato `cards` partendo dalla linea temporale principale:

```
tellTarget("/box/cards")  
tellTarget("_level0/box/cards")
```

L'esempio seguente fa riferimento alla linea temporale principale partendo dal clip filmato `cards`:

```
tellTarget("../../cards")
tellTarget("_root")
```

L'esempio seguente fa riferimento al clip filmato `cards` principale:

```
tellTarget("../cards")
tellTarget("_parent/cards")
```

Uso delle variabili

Per specificare una variabile su una linea temporale, utilizzare la sintassi della barra (/) abbinata ai punti (..) e ai due punti (:). È possibile utilizzare anche la notazione del punto.

Il codice seguente fa riferimento alla variabile `car` nella linea temporale principale:

```
/:car
_root.car
```

L'esempio seguente mostra la variabile `car` in un'istanza di clip filmato situata nella linea temporale principale:

```
/mc1/mc2/:car
_root.mc1.mc2.car
```

L'esempio seguente mostra la variabile `car` in un'istanza di clip filmato situata nella linea temporale corrente:

```
mc2/:car
mc2.car
```

Emulazione degli array

Gli array sono utili per creare e manipolare elenchi ordinati di informazioni quali variabili e valori. Tuttavia, Flash Lite 1.1 non supporta strutture di dati array native. Nella programmazione con Flash Lite (e Flash 4), una tecnica comune consiste nell'emulare gli array mediante l'elaborazione delle stringhe. Gli array emulati sono detti anche pseudo-array. L'elaborazione degli pseudo-array si basa sulla funzione `eval()` di ActionScript che consente di accedere a variabili, proprietà o clip filmato in base al nome. Per ulteriori informazioni, vedere [“Uso della funzione eval\(\)” a pagina 20](#).

In genere, uno pseudo-array è costituito da due o più variabili con lo stesso nome di base seguito da un suffisso numerico. Il suffisso è la posizione di indice dei singoli elementi dell'array.

Si supponga ad esempio di creare le seguenti variabili di ActionScript:

```
color_1 = "orange";  
color_2 = "green";  
color_3 = "blue";  
color_4 = "red";
```

A questo punto è possibile utilizzare il codice seguente per eseguire elaborazioni cicliche degli elementi dello pseudo-array:

```
for (i = 1; i <=4; i++) {  
    trace (eval ("color_" + i));  
}
```

Oltre ad avere la possibilità di fare riferimento a variabili esistenti, è possibile utilizzare la funzione `eval()` a sinistra dell'assegnazione di una variabile per creare variabili in fase di runtime. Si supponga ad esempio di voler gestire un elenco dei punteggi più elevati raggiunti da un utente durante un gioco. Ogni volta che l'utente termina una partita, il punteggio viene aggiunto all'elenco:

```
eval("highScore" + i + scoreIndex) = currentScore;  
scoreIndex++;
```

Tutte le volte che viene eseguito, questo codice aggiunge una nuova voce all'elenco dei punteggi massimi e incrementa la variabile `scoreIndex` che determina la posizione di indice delle singole voci all'interno dell'elenco. Ad esempio, si possono ottenere le seguenti variabili:

```
highScore1 = 2000  
highScore2 = 1500  
highScore3 = 3000
```

Operazioni con il testo e con le stringhe

Flash Lite è dotato di alcuni comandi e proprietà ActionScript di base per eseguire operazioni con il testo. È possibile ottenere e impostare i valori dei campi di testo, concatenare le stringhe, eseguire la codifica o la decodifica URL delle stringhe di testo e creare campi di testo scorrevoli.

In questa sezione sono presentati i seguenti argomenti:

- [“Concatenazione di stringhe” a pagina 15](#)
- [“Testo scorrevole” a pagina 15](#)

Concatenazione di stringhe

Per concatenare le stringhe in Flash Lite si utilizza l'operatore `add`, come nell'esempio seguente:

```
city = "Boston";
team = "Red Sox";
fullName = city add " " add team;
// Risultato:
// fullName = "Boston Red Sox"
```

Testo scorrevole

La proprietà `scroll` dei campi di testo dinamici e di input si può utilizzare per ottenere o impostare la posizione di scorrimento corrente del campo. È inoltre possibile utilizzare la proprietà `maxscroll` per determinare la posizione di scorrimento corrente di un campo di testo rispetto alla posizione di scorrimento massima. Per un esempio di creazione di un campo di testo a scorrimento, vedere “Creazione di testo scorrevole” in *Sviluppo di applicazioni Flash Lite*.

Uso della funzione `call()` per la creazione di funzioni

In Flash Lite non è possibile definire o chiamare funzioni personalizzate come in Flash Player 5 e nelle versioni successive. È tuttavia possibile utilizzare la funzione `call()` di ActionScript per eseguire il codice che si trova su un fotogramma specifico nella linea temporale. Questa tecnica consente di incorporare parti di codice di uso comune in un'unica posizione, facilitandone la gestione.

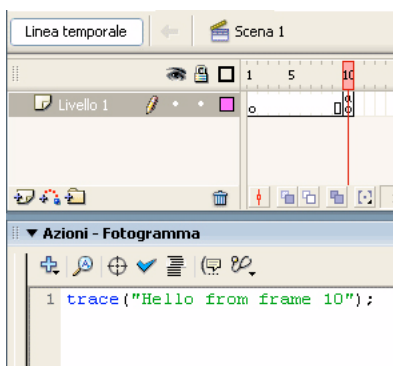
La funzione `call()` utilizza come parametro il numero o l'etichetta di un fotogramma. Ad esempio, il codice ActionScript seguente chiama il codice situato sul fotogramma `moveUp`:

```
call("moveUp");
```

La funzione `call()` opera in modo sincrono; qualsiasi codice ActionScript posto dopo una chiamata alla funzione `call()` viene eseguito solo al termine dell'esecuzione di tutto il codice ActionScript presente nel fotogramma specificato.

Per chiamare il codice ActionScript presente in un altro fotogramma:

1. In un nuovo documento Flash, inserire un fotogramma chiave nel fotogramma 10.



2. Con il nuovo fotogramma chiave selezionato, aprire il pannello Azioni (Finestra > Azioni) e digitare il seguente codice:

```
trace("Hello from frame 10");
```

3. Selezionare il fotogramma chiave nel fotogramma 1 e digitare il codice seguente nel pannello Azioni:

```
stop();  
call(10);
```

Questo codice arresta l'indicatore di riproduzione sul fotogramma 1 e quindi chiama il codice nel fotogramma 10.

4. Provare l'applicazione in Adobe® Device Central™.

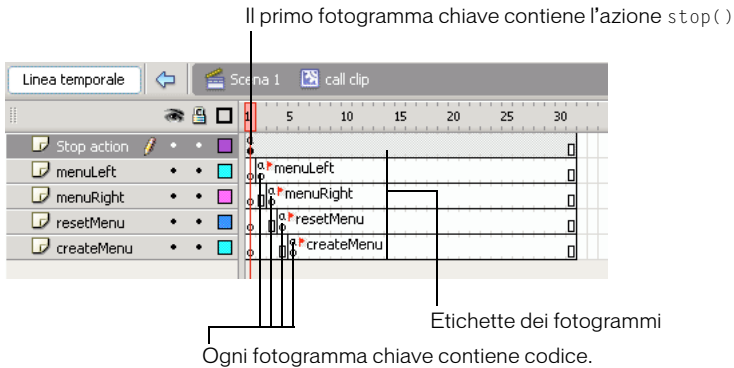
Nell'emulatore dovrebbe essere visualizzato "Hello from frame 10".

È possibile anche chiamare codice da un'altra linea temporale, ad esempio quella del clip filmato. Per eseguire il codice, indicare il nome di istanza del clip filmato seguito dai due punti e dal numero o dall'etichetta del fotogramma. Ad esempio, il codice ActionScript seguente chiama il codice che si trova sul fotogramma denominato `moveUp` nell'istanza di clip filmato `callClip`:

```
call("callClip:moveUp");
```

Questa tecnica viene spesso utilizzata per creare clip di chiamata (*call clips*) o clip di funzione (*function clips*), ovvero clip filmato il cui unico scopo è incorporare parti di codice utilizzate regolarmente. Un clip di chiamata contiene un fotogramma chiave per ogni funzione da creare. In genere, i nomi assegnati ai singoli fotogrammi chiave dipendono dalla funzione questi che svolgono. Adobe consiglia anche di creare un nuovo livello per ogni nuovo fotogramma chiave e di assegnare a ogni livello lo stesso nome dell'etichetta assegnata al fotogramma chiave.

La figura seguente mostra la linea temporale di un esempio di clip di chiamata. Il primo fotogramma chiave di un clip di chiamata contiene sempre un'azione `stop()`, che garantisce che l'indicatore di riproduzione non venga ripetuto in continuazione nei fotogrammi della sua linea temporale. I fotogrammi chiave successivi contengono codice per ogni "funzione". A ogni fotogramma chiave con una funzione viene assegnata un'etichetta descrittiva. Per facilitare la modifica e la visualizzazione del clip di chiamata, i fotogrammi chiave delle varie funzioni vengono generalmente inseriti in un livello separato.



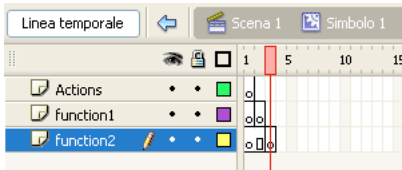
La procedura seguente illustra come creare e utilizzare un clip di chiamata.

Per creare e utilizzare un clip di chiamata:

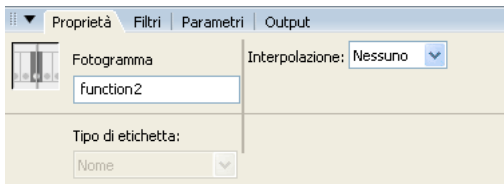
1. In Adobe® Flash® CS3, creare un nuovo documento utilizzando il modello Flash Lite 1.1 - Symbian Series 60.
2. Selezionare Inserisci > Nuovo simbolo.
3. Nella finestra di dialogo Crea un nuovo simbolo, digitare **Call Clip** nella casella di testo Nome e fare clic su OK.
Il clip filmato viene aperto in modalità di modifica.
4. Fare clic due volte sul pulsante Inserisci livello nella finestra Linea temporale per inserire due nuovi livelli.

Assegnare al primo livello il nome **Actions**, al secondo il nome **function1** e al terzo il nome **function2**.

- Inserire un fotogramma chiave nel fotogramma 2 del livello function1 e un altro fotogramma chiave nel fotogramma 3 del livello function2, come mostra la figura seguente:

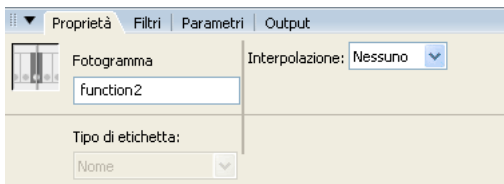


- Selezionare il fotogramma chiave del livello Actions e aprire il pannello Azioni.
- Aggiungere un'azione `stop()` al pannello Azioni.
- Selezionare il fotogramma chiave nel fotogramma 2 del livello function1 ed eseguire le seguenti operazioni:
 - Nella finestra di ispezione Proprietà, digitare **function1** nella casella di testo Etichetta fotogramma.



- Nel pannello Azioni (Finestra > Azioni), digitare il seguente codice:

```
trace("function1 was called.");
```
- Selezionare il fotogramma chiave nel fotogramma 3 del livello function2 ed eseguire le seguenti operazioni:
 - Nella finestra di ispezione Proprietà, digitare **function2** nella casella di testo Etichetta fotogramma.



- Nel pannello Azioni (Finestra > Azioni), digitare il seguente codice:

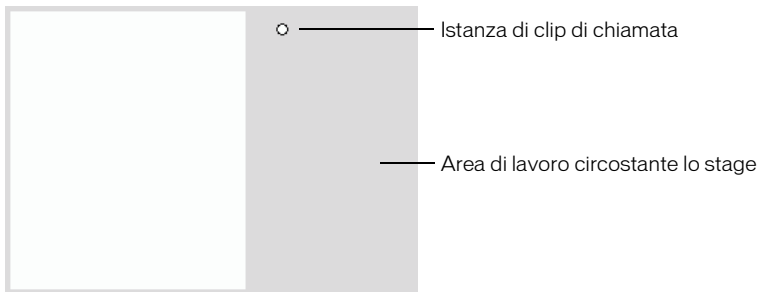
```
trace("function2 was called.");
```
- Premere **Ctrl+E** (Windows) o **Comando+E** (Macintosh) per tornare alla linea temporale principale.

11. Impostare la vista del documento per visualizzare l'area di lavoro circostante lo stage selezionando *Visualizza > Area di lavoro*.

Il clip di chiamata si può posizionare nell'area di lavoro, dal momento che non è necessario che sia visibile all'utente.

12. Aprire il pannello Libreria (*Finestra > Libreria*) e trascinare il simbolo Call Clip nell'area di lavoro attorno allo stage.

Poiché non contiene elementi visuali, il clip di chiamata è visualizzato sullo stage come un circoletto che rappresenta il punto di registrazione del clip filmato.



SUGGERIMENTO

Per rendere il clip di chiamata più facilmente individuabile sullo stage, aggiungere del testo o un altro elemento visuale al primo fotogramma chiave della linea temporale del clip di chiamata.

13. Nella finestra di ispezione Proprietà, digitare **callClip** nella casella di testo Nome istanza.
14. Nella linea temporale, selezionare il fotogramma 1 nel livello denominato **ActionScript**.
15. Nel pannello Azioni, immettere il codice seguente:

```
call("callClip:function1");  
call("callClip:function2");
```

16. Provare l'applicazione nell'emulatore (*Controllo > Prova filmato*).

Viene visualizzato il testo seguente:

```
function1 was called.  
function2 was called.
```

Uso della funzione eval()

La funzione `eval()` consente di fare riferimento in modo dinamico a variabili e a istanze di clip filmato in fase di runtime. La funzione `eval()` utilizza come parametro un'espressione stringa e restituisce il valore della variabile rappresentata dall'espressione o un riferimento a un clip filmato.

Ad esempio, il codice seguente valuta il valore della variabile ActionScript `name` e assegna il risultato a `nameValue`:

```
name = "Jack";
nameValue = eval("name");
// risultato: nameValue = "Jack"
```

La funzione `eval()` viene spesso utilizzata con cicli `for()` e con l'operatore `add` (concatenazione di stringhe) per creare array basati su stringhe, dal momento che Flash Lite non supporta le strutture di dati array native. Per ulteriori informazioni, vedere [“Emulazione degli array” a pagina 13](#).

La funzione `eval()` si può utilizzare anche per fare riferimento a istanze di clip filmato in base al nome. Si supponga ad esempio di disporre di tre clip filmato denominati rispettivamente `clip1`, `clip2` e `clip3`. Il ciclo `for()` seguente incrementa la posizione `x` di ogni clip di 10 pixel:

```
for(index = 1; index <= 3; index++) {
    eval("clip" + index)._x += 10
}
```

Operazioni comuni di creazione di script

In questo capitolo sono descritte le operazioni comuni di creazione di script in Flash Lite per il dispositivo dell'utente, quali ad esempio l'accesso alle informazioni sulle funzionalità del dispositivo, l'avvio di telefonate, l'invio di messaggi di testo e la determinazione dello stato della rete.

In questo capitolo sono presentati i seguenti argomenti:

Come determinare le funzionalità del dispositivo e della piattaforma	21
Apertura di una pagina Web	22
Avvio di una telefonata	23
Invio di un messaggio di testo o multimediale	23
Invio di un messaggio e-mail	24
Caricamento di file SWF esterni	24
Caricamento di dati esterni	25

Come determinare le funzionalità del dispositivo e della piattaforma

Macromedia Flash Lite 1.1 di Adobe comprende numerose variabili ActionScript che forniscono informazioni sulle caratteristiche e le funzionalità disponibili per le applicazioni Flash Lite eseguite su un determinato dispositivo. Ad esempio, la variabile `_capLoadData` indica se il dispositivo supporta il caricamento di dati esterni, mentre la variabile `_capSMS` indica se il dispositivo è predisposto per l'invio di SMS. Per un elenco completo delle variabili relative alle funzionalità, vedere "Funzionalità" in *Guida di riferimento di ActionScript per Flash Lite 1.x*.

Generalmente, le variabili relative alle funzionalità si utilizzano per determinare se un dispositivo supporta una funzione specifica prima di provare a utilizzarla. Ad esempio, si supponga di volere sviluppare un'applicazione che scarichi dati da un server Web mediante la funzione `loadVariables()`. Prima di provare a scaricare i dati, è possibile verificare il valore della variabile `_capLoadData` per determinare se il dispositivo supporta questa funzione, come nell'esempio seguente:

```
if(_capLoadData == 1) {  
    loadVariables("http://foo.com/data.txt");  
} else {  
    status_message = "Sorry, unable to load external data."  
}
```

Flash Lite definisce le variabili relative alle funzionalità nella linea temporale principale del file SWF principale. Pertanto, per accedere a tali variabili da un'altra linea temporale (ad esempio dalla linea temporale di un clip filmato) è necessario specificare il percorso completo della variabile. Nell'esempio seguente viene utilizzata la barra (/) per specificare il percorso completo della variabile `_capSMS`.

```
canSendSMS = /:_capSMS
```

Apertura di una pagina Web

Per aprire una pagina Web nel browser del dispositivo si utilizza il comando `getURL()`. Si tratta dello stesso sistema utilizzato per aprire le pagine Web da un'applicazione desktop di Flash. Il seguente codice, ad esempio, apre la home page del sito Web Adobe:

```
getURL("http://www.adobe.com");
```

Flash Lite elabora solo un'azione `getURL()` per ogni fotogramma o gestore di eventi. Alcuni tipi di microtelefoni limitano l'azione `getURL()` ai soli eventi di pressione di un tasto, nel qual caso la chiamata `getURL()` viene elaborata solo se viene attivata all'interno del gestore di eventi per la pressione di un tasto. Anche in questo caso, per ogni gestore di eventi relativo alla pressione di un tasto viene elaborata solo un'azione `getURL()`. Il codice seguente, associato a un'istanza di pulsante sullo stage, provoca l'apertura di una pagina Web quando si preme il pulsante di selezione sul dispositivo:

```
on (keyPress "<Enter>"){  
    getURL("http://www.adobe.com");  
}
```

Avvio di una telefonata

Per avviare una telefonata da un'applicazione Flash Lite si utilizza la funzione `getURL()`. In genere, questa funzione si usa per aprire una pagina Web, ma in questo caso si specifica `tel:` come protocollo (al posto di `http`) e si indica quindi il numero telefonico che si desidera comporre. Quando si chiama questa funzione, Flash Lite visualizza una finestra di dialogo che chiede all'utente di confermare l'esecuzione della chiamata al numero specificato.

Il codice seguente cerca di avviare una telefonata al numero 555-1212:

```
getURL("tel:555-1212");
```

Flash Lite elabora solo un'azione `getURL()` per ogni fotogramma o gestore di eventi. Alcuni tipi di microtelefoni limitano l'azione `getURL()` ai soli eventi di pressione di un tasto, nel qual caso la chiamata `getURL()` viene elaborata solo se viene attivata all'interno del gestore di eventi per la pressione di un tasto. Anche in questo caso, per ogni gestore di eventi relativo alla pressione di un tasto viene elaborata solo un'azione `getURL()`. L'esempio seguente avvia una telefonata quando l'utente preme il pulsante di selezione sul dispositivo:

```
on (keyPress "<Enter>"){  
    getURL("tel:555-1212");  
}
```

Invio di un messaggio di testo o multimediale

Flash Lite si può utilizzare per inviare SMS o MMS. Per inviare un SMS o un MMS in un'applicazione Flash Lite, utilizzare il comando `getURL()` con i protocolli `sms:` o `mms:` (anziché il protocollo standard `http`) seguiti dal numero telefonico a cui si desidera inviare il messaggio.

```
getURL("sms:555-1212");
```

Facoltativamente è possibile indicare il corpo del messaggio nella stringa di query dell'URL, come nell'esempio di codice seguente:

```
getURL("sms:555-1212?body=More info please");
```

Per inviare un MMS, utilizzare il protocollo `mms:` anziché il protocollo `sms:`, come nell'esempio seguente:

```
getURL("mms:555-1212");
```

NOTA

Da Flash Lite non è possibile specificare un allegato al messaggio MMS.

Flash Lite elabora solo un'azione `getURL()` per ogni fotogramma o gestore di eventi. Alcuni tipi di microtelefoni limitano l'azione `getURL()` ai soli eventi di pressione di un tasto, nel qual caso la chiamata `getURL()` viene elaborata solo se viene attivata all'interno del gestore di eventi per la pressione di un tasto. Anche in questo caso, per ogni gestore di eventi relativo alla pressione di un tasto viene elaborata solo un'azione `getURL()`. L'esempio seguente invia un SMS quando l'utente preme il pulsante di selezione sul dispositivo:

```
on (keyPress "<Enter>"){
    getURL("sms:555-1212");
}
```

Invio di un messaggio e-mail

È possibile utilizzare Flash Lite per inviare messaggi e-mail eseguendo il comando `getURL()` con il protocollo `mailto:` seguito dall'indirizzo e-mail del destinatario. Facoltativamente è possibile indicare il titolo e il corpo del messaggio nella stringa di query dell'URL, come nell'esempio seguente:

```
getURL("mailto:mobile-developer@adobe.com?subject=Flash Lite");
```

Per specificare solo il corpo del messaggio nella stringa di query, utilizzare il codice seguente:

```
getURL("mailto:mobile-developer@adobe.com?body=More+info+please");
```

Caricamento di file SWF esterni

La funzione `loadMovie()` consente di caricare file SWF da una rete o un sistema locale. Questa funzione è disponibile solo in Flash Lite 1.1 e versioni successive. Quando si caricano file SWF esterni, considerare quanto segue:

- Flash Lite consente di caricare altri file SWF di Flash Lite 1.0 o 1.1 oppure file SWF formattati per Flash 4 o versioni precedenti. Se si tenta di caricare un file SWF in un altro formato (ad esempio un file SWF di Flash Player 6), Flash Lite genera un errore di runtime.
- Flash Lite non permette di caricare direttamente file di immagine esterni, ad esempio in formato JPEG o GIF. Per caricare materiale multimediale di questi tipi, è necessario convertire i dati di immagine in formato SWF. L'operazione può essere eseguita "manualmente" nello strumento di creazione Flash importando il file di immagine in un nuovo documento e quindi esportando il file nel formato SWF per Flash Lite o Flash 4. Sono inoltre disponibili programmi di utilità di altri produttori che permettono di effettuare questo tipo di conversione in modo automatico.

Per ulteriori informazioni sul caricamento dei file SWF, vedere `loadMovie()` nella *Guida di riferimento di ActionScript per Flash Lite 1.x*.

Caricamento di dati esterni

Per caricare dati esterni in un'applicazione Flash Lite, utilizzare la funzione `loadVariables()`. È possibile caricare i dati attraverso una rete (da un indirizzo HTTP) oppure dal sistema locale. Questa funzione è disponibile solo in Flash Lite 1.1 e versioni successive.

Questa sezione spiega come utilizzare la funzione `loadVariables()` per caricare dati da un file esterno e visualizzarli in campi di testo dinamici. Innanzi tutto, occorre creare il file di dati, ovvero un file di testo che contiene cinque coppie nome-valore separate dal carattere della e commerciale (&). Quindi, si crea l'applicazione Flash Lite che carica e visualizza i dati contenuti nel file di testo.

L'esempio presuppone che il file di dati e il file SWF siano contenuti nella stessa cartella e si trovino nel computer locale (se si esegue la prova con l'emulatore) o nella scheda di memoria del dispositivo (se si effettua la prova su un dispositivo reale). Per provare l'applicazione sul dispositivo, procedere nel modo seguente:

- Trasferire il file di testo sul dispositivo e salvarlo nella stessa cartella del file SWF.
- Collocare il file di testo in corrispondenza di un URL su un server Web (ad esempio, `www.your-server.com/data.txt`). Quindi, modificare la chiamata `loadVariables()` nell'applicazione di esempio in modo che faccia riferimento all'URL utilizzato, come nell'esempio seguente:

```
loadVariables("http://www.your-server.com/data.txt", "data_clip");
```

Per un esempio di applicazione che carica i dati dalla rete, vedere “Notiziario Flash Lite” in *Esempi di Flash*.

Per creare il file di dati:

1. Utilizzando un editor di testo (ad esempio Blocco note o SimpleText), creare un file che contenga il testo seguente:

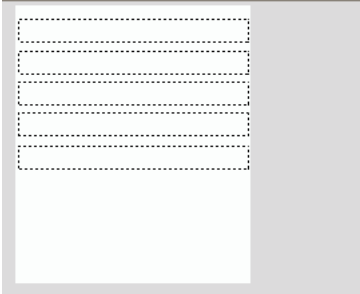
```
item_1=Hello&item_2=Bonjour&item_3=Hola&item_4=Buon+giorno&item_5=G'day
```

2. Salvare il file con nome `data.txt`.

Per creare l'applicazione Flash Lite che provvede a caricare i dati:

1. Creare un nuovo documento utilizzando il modello Flash Lite 1.1 - Symbian Series 60. Per ulteriori informazioni sull'uso dei modelli di documento Flash Lite, consultare la sezione “Uso dei modelli di documenti Flash Lite” nella *Guida introduttiva di Flash Lite 1.x*.
2. Salvare il file con il nome `dataloading fla` nella stessa cartella che contiene il file di testo (`data.txt`) creato in precedenza.

3. Nella linea temporale, selezionare il fotogramma 1 del livello denominato Content.
4. Utilizzando lo strumento Testo, creare cinque campi di testo dinamici sullo stage, come mostrato nella figura seguente:



5. Selezionare il primo campo di testo (quello più in alto) e, nella finestra di ispezione Proprietà, digitare `item_1` nella casella di testo Var.
Questo nome di variabile corrisponde al nome della prima variabile definita nel file `data.txt` (`item_1=Hello`).
6. Seguendo le istruzioni fornite nei due passaggi precedenti, assegnare ai quattro campi di testo rimanenti i nomi di variabile `item_2`, `item_3`, `item_4` e `item_5`.
7. Selezionare ogni campo di testo tenendo premuto il tasto Maiusc (in modo da selezionarli tutti), quindi selezionare **Elabora > Converti in simbolo**.
8. Nella finestra di dialogo **Converti in simbolo**, selezionare **Clip filmato** come tipo di simbolo, quindi fare clic su **OK**.
9. Selezionare il clip filmato appena creato, quindi, nella finestra di ispezione **Proprietà**, digitare `data_clip` nella casella di testo **Nome istanza**.
10. Nella Linea temporale, selezionare il fotogramma 1 del livello **Actions** e aprire il pannello **Azioni** (**Finestra > Azioni**).
11. Immettere il codice seguente nel pannello **Azioni**:

```
loadVariables("data.txt", "data_clip");
```

12. Salvare le modifiche (File > Salva) e provare l'applicazione nell'emulatore (Controllo > Prova filmato).

Ogni campo di testo dovrebbe apparire compilato con i dati del file di testo, come nella figura seguente:



Indice analitico

A

- ActionScript per Flash Lite 1.x
 - differenze tra 1.0 e 1.1 8
 - elementi di ActionScript per Flash 4 non supportati 9
 - funzioni non disponibili 9
 - panoramica 7
- apertura di una pagina Web 22
- array, emulazione mediante stringhe 13

C

- call(), uso della funzione 15
- caricamento di dati esterni 25
- caricamento di file SWF esterni 24
- chiamate telefoniche, esecuzione 23
- clip di funzione, creazione 15
- clip filmato
 - ottenere e impostare proprietà 11
 - riferimento dinamico 20
- concatenazione di stringhe 15

E

- eval(), uso della funzione 20

F

- funzioni, emulazione mediante call() 15

G

- getURL(), funzione
 - apertura di pagine Web 22
 - esecuzione di una chiamata telefonica 23
 - invio di un messaggio di testo 23
 - invio di un messaggio e-mail 24
 - invio di un messaggio multimediale 23

I

- invio di messaggi 23

L

- linee temporali, controllo con ActionScript 12
- loadMovie(), uso della funzione 24
- loadVariables(), uso della funzione 25

M

- messaggio di testo, invio 23
- messaggio e-mail, invio 24
- messaggio multimediale, invio 23

P

- pagina Web, apertura 22

S

- stringhe, concatenazione 15

T

- telefonata, esecuzione 23
- testo scorrevole, creazione 15

V

- variabili
 - riferimenti 13
 - riferimento dinamico 20
 - sintassi del punto e della barra 13
- variabili di funzionalità della piattaforma, informazioni 21

