



macromedia®

**FLASH**

Aprendizaje de ActionScript en Flash Lite 1.X

8

## Marcas comerciales

1 Step RoboPDF, ActiveEdit, ActiveTest, Authorware, Blue Sky Software, Blue Sky, Breeze, Breezo, Captivate, Central, ColdFusion, Contribute, Database Explorer, Director, Dreamweaver, Fireworks, Flash, FlashCast, FlashHelp, Flash Lite, FlashPaper, Flash Video Encoder, Flex, Flex Builder, Fontographer, FreeHand, Generator, HomeSite, JRun, MacRecorder, Macromedia, MXML, RoboEngine, RoboHelp, RoboInfo, RoboPDF, Roundtrip, Roundtrip HTML, Shockwave, SoundEdit, Studio MX, UltraDev y WebHelp son marcas registradas o marcas comerciales de Macromedia, Inc. y pueden estar registradas en Estados Unidos o en otras jurisdicciones, incluidas las internacionales. Otros nombres de productos, logotipos, diseños, títulos, palabras o frases mencionados en esta publicación pueden ser marcas comerciales, marcas de servicio o nombres registrados de Macromedia, Inc. o de otras entidades, y pueden estar registrados en ciertas jurisdicciones, incluidas las internacionales.

## Información de terceros

Esta guía contiene vínculos a sitios Web de terceros que no están bajo el control de Macromedia y, por consiguiente, Macromedia no se hace responsable del contenido de dichos sitios Web. El acceso a uno de los sitios Web de terceros mencionados en esta guía será a cuenta y riesgo del usuario. Macromedia proporciona estos vínculos únicamente como ayuda y su inclusión no implica que Macromedia se haga responsable del contenido de dichos sitios Web.

La tecnología de compresión y descompresión de voz tiene licencia de Nellymoser, Inc. ([www.nellymoser.com](http://www.nellymoser.com)).



La tecnología de compresión y descompresión de vídeo Sorenson™ Spark™ tiene licencia de Sorenson Media, Inc.

Navegador Opera ® Copyright © 1995-2002 Opera Software ASA y sus proveedores. Todos los derechos reservados.

Macromedia Flash 8 utiliza tecnología de vídeo de On2 TrueMotion. © 1992-2005 On2 Technologies, Inc. Todos los derechos reservados. <http://www.on2.com>.

Visual SourceSafe es una marca registrada o un marca comercial de Microsoft Corporation en Estados Unidos y otros países.

**Copyright © 2005 Macromedia, Inc. Todos los derechos reservados. No se permite la copia, fotocopia, reproducción, traducción ni la conversión en formato electrónico o legible por equipos, ya sea de forma total o parcial de este manual, sin la autorización previa por escrito de Macromedia, Inc. No obstante, el propietario o usuario autorizado de una copia válida del software con la que se proporcionó este manual puede imprimir una copia del manual a partir de una versión electrónica del mismo, con el solo fin de aprender a usar dicho software, siempre que no se imprima, reproduzca, distribuya, revenda o transmita ninguna parte de este manual para cualquier otro propósito, incluidos, sin limitación, fines comerciales, como la venta de copias de esta documentación o el suministro de servicios de soporte pagados.**

## Agradecimientos

Dirección del proyecto: Mary Leigh Burke

Redacción: Tim Statler

Directora de edición: Rosana Francescato

Edición: Linda Adler, Geta Carlson, Evelyn Eldridge, Mary Kraemer, Lisa Stanziano

Administración de la producción: Patrice O'Neill, Kristin Conradi, Yuko Yagi

Producción y diseño multimedia: Adam Barnett, Aaron Begley, Paul Benkman, John Francis, Geeta Karmarkar, Masayo Noda, Paul Rangel, Arena Reed, Mario Reynoso

Reconocimiento especial a Lisa Friendly, Bonnie Loo, Erick Vera, los probadores beta y todo el equipo de ingeniería de Flash Lite y los equipos de control de calidad.

Primera edición: Septiembre de 2005

Macromedia, Inc.

601 Townsend St.

San Francisco, CA 94103, EE.UU.

# Contenido

<b>Capítulo 1: ActionScript de Flash Lite 1.x</b> . . . . .	<b>5</b>
Información general sobre ActionScript de Flash Lite 1.x . . . . .	5
Diferencias entre ActionScript de Flash Lite 1.0 y Flash Lite 1.1 . . . . .	6
Elementos de ActionScript de Flash 4 no admitidos en ActionScript de Flash Lite 1.x . . . . .	6
Funciones no disponibles en ActionScript de Flash Lite 1.x . . . . .	7
<b>Capítulo 2: Flash 4 ActionScript Primer</b> . . . . .	<b>9</b>
Obtención y definición de las propiedades de clip de película . . . . .	9
Control de otras líneas de tiempo . . . . .	10
Utilización de variables . . . . .	11
Emulación de matrices . . . . .	11
Trabajo con texto y cadenas . . . . .	12
Utilización de la función call() para crear funciones . . . . .	13
Utilización de la función eval() . . . . .	18
<b>Capítulo 3: Tareas comunes de creación de scripts</b> . . . . .	<b>19</b>
Determinación de las características del dispositivo y la plataforma . . . . .	20
Apertura de una página Web . . . . .	20
Inicio de una llamada telefónica . . . . .	21
Inicio de un mensaje de texto o multimedia . . . . .	22
Inicio de un mensaje de correo electrónico . . . . .	23
Carga de archivos SWF externos . . . . .	23
Carga de datos externos . . . . .	24
<b>Índice alfabético</b> . . . . .	<b>27</b>



ActionScript se utiliza para añadir lógica de programación e interactividad a las aplicaciones Macromedia Flash Lite. La versión de ActionScript en Flash Lite 1.0 y 1.1, denominada ActionScript de Flash Lite 1.x de forma colectiva, es una combinación de elementos de ActionScript de Flash 4 con comandos adicionales y propiedades específicas del reproductor Flash Lite, como la posibilidad de iniciar llamadas telefónicas o mensajes de texto o de obtener información de fecha y hora del dispositivo.

Este capítulo contiene los siguientes temas:

Información general sobre ActionScript de Flash Lite 1.x.....	5
Diferencias entre ActionScript de Flash Lite 1.0 y Flash Lite 1.1.....	6
Elementos de ActionScript de Flash 4 no admitidos en ActionScript de Flash Lite 1.x.....	6
Funciones no disponibles en ActionScript de Flash Lite 1.x.....	7

## Información general sobre ActionScript de Flash Lite 1.x

ActionScript de Flash Lite 1.x consta de lo siguiente:

**ActionScript de Flash Player 4** Se incluyen operadores (por ejemplo, de comparación y asignación), propiedades de clip de película (por ejemplo, `_height`, `_x` e `_y`), funciones de control de la línea de tiempo (por ejemplo, `gotoAndPlay()` o `stop()`) y funciones de red, como las funciones `loadVariables()` y `loadMovie()` (Flash Lite 1.1 solamente). Para obtener una lista de elementos de ActionScript de Flash 4 no admitidos, consulte “Elementos de ActionScript de Flash 4 no admitidos en ActionScript de Flash Lite 1.x” en la página 6.

**Comandos y propiedades de integración telefónica** Flash Lite proporciona comandos que permiten, por ejemplo, consultar los datos de fecha y hora del dispositivo, iniciar una llamada telefónica o un mensaje de texto del servicio de mensajes cortos (SMS) o iniciar aplicaciones externas instaladas en el dispositivo.

**Variables de características de la plataforma (Flash Lite 1.1 solamente)** Estas propiedades ofrecen información sobre las características del dispositivo o el entorno de tiempo de ejecución de Flash Lite. Por ejemplo, la variable `_capLoadData` indica si la aplicación puede cargar datos a través de la red.

**Función `fscommand2()`** Al igual que la función `fscommand()`, `fscommand2()` se utiliza para comunicarse con el entorno o sistema host, en este caso el teléfono o dispositivo móvil. La función `fscommand2()` incorpora una serie de mejoras con respecto a `fscommand()`, incluida la posibilidad de transmitir un número arbitrario de argumentos y de recuperar de inmediato los valores devueltos (en lugar de tener que esperar al fotograma siguiente, como sucede con `fscommand()`).

## Diferencias entre ActionScript de Flash Lite 1.0 y Flash Lite 1.1

Las funciones de ActionScript de Flash Lite 1.1 siguientes no se encuentran disponibles en Flash Lite 1.0:

- Información de acceso de red o estado de red. Por ejemplo, en Flash Lite 1.0 no es posible utilizar las funciones `loadVariables()` o `loadMovie()` para cargar datos externos o archivos SWF, ni tampoco usar los distintos comandos `fscommand2()` para determinar la intensidad de la señal de conexión del dispositivo o el estado de una solicitud de red.
- Obtención de información de fecha y hora del dispositivo.
- Las variables de características de la plataforma, que proporcionan información acerca de las características de la plataforma Flash Lite y del dispositivo.
- La función `fscommand2()` y los comandos asociados con ella, como `SetSoftKeys` y `FullScreen`.
- Las propiedades de campo de texto `scroll` y `maxscroll`.

## Elementos de ActionScript de Flash 4 no admitidos en ActionScript de Flash Lite 1.x

Las siguientes funciones de ActionScript de Flash 4 no se pueden utilizar o sólo se admiten parcialmente en ActionScript de Flash Lite 1.x:

- Las funciones `startDrag()` y `stopDrag()`.

- ActionScript de Flash Lite 1.x admite un subconjunto de eventos de botón admitidos en Flash Player 4. Para obtener más información sobre el control de los eventos de botón, consulte Capítulo 1, “Creación de interactividad y navegación” en *Desarrollo de aplicaciones Flash Lite*.
- ActionScript de Flash Lite 1.x admite un subconjunto de eventos de teclas de Flash Player 4. Para obtener más información sobre los eventos de teclas admitidos en Flash Lite, consulte Capítulo 1, “Creación de interactividad y navegación” en *Desarrollo de aplicaciones Flash Lite*.
- La propiedad `_dropTarget`.
- La propiedad `_soundBufTime`.
- La propiedad `_url`.
- La función de conversión `String()`.

## Funciones no disponibles en ActionScript de Flash Lite 1.x

Como el reproductor Flash Lite se basa en una versión anterior de Flash Player, no admite todas las funciones de programación disponibles en las versiones más recientes de Flash Player ni otros lenguajes de programación con los que el usuario puede estar familiarizado. En esta sección se describen características de programación no disponibles en ActionScript de Flash Lite 1.x y las alternativas y soluciones disponibles.

**Funciones definidas por el usuario** Flash Lite 1.x no admite la posibilidad de definir y realizar llamadas a funciones personalizadas. Sin embargo, puede utilizar la función `call()` para ejecutar código que reside en un fotograma arbitrario de la línea de tiempo. Para más información, consulte [“Utilización de la función call\(\) para crear funciones” en la página 13](#).

**Matrices nativas, objetos y otros tipos de datos complejos** Flash Lite 1.x no admite las estructuras de datos de matrices nativas y otros tipos de datos complejos. Sin embargo, es posible emular matrices mediante el uso de pseudo-matrices, una técnica que requiere usar la función `eval()` para evaluar las cadenas concatenadas de forma dinámica. Para más información, consulte [“Emulación de matrices” en la página 11](#).

**Carga en tiempo de ejecución de archivos externos de imagen o de sonido** A diferencia de la versión de escritorio de Flash Player, ActionScript de Flash Lite 1.x no puede cargar archivos externos JPEG o MP3. En Flash Lite 1.1 es posible utilizar la función `loadMovie()` para cargar archivos SWF externos. Para más información, consulte [“Carga de archivos SWF externos” en la página 23](#).





ActionScript de Flash Lite 1.x se basa en la primera versión de ActionScript disponible en Flash Player 4. Por consiguiente, en las aplicaciones Flash Lite 1.x no se encuentran disponibles varias funciones de programación de las versiones más recientes de Flash Player (para sistemas de escritorio).

Si no está familiarizado con la sintaxis y las funciones de ActionScript de Flash 4 o ha olvidado algunos pormenores de las tareas anteriores de desarrollo de Flash, en este capítulo se ofrece una descripción básica del uso de ActionScript de Flash 4 en las aplicaciones Flash Lite.

Este capítulo contiene los siguientes temas:

Obtención y definición de las propiedades de clip de película . . . . .	9
Control de otras líneas de tiempo . . . . .	10
Utilización de variables . . . . .	11
Emulación de matrices . . . . .	11
Trabajo con texto y cadenas . . . . .	12
Utilización de la función <code>call()</code> para crear funciones . . . . .	13
Utilización de la función <code>eval()</code> . . . . .	18

## Obtención y definición de las propiedades de clip de película

Para obtener y definir una propiedad de clip de película (si es configurable), puede utilizar la sintaxis con punto o las funciones `setProperty()` o `getProperty()`. También puede usar la función `tellTarget()`.

Para utilizar la sintaxis con punto, especifique el nombre de la instancia del clip de película seguido de un punto (.) y, a continuación, el nombre de la propiedad. Por ejemplo, con el código siguiente se obtiene la coordenada de pantalla *x* (representada por la propiedad de clip de película `_x`) del clip de película denominado `cartoonArea` y el resultado se asigna a una variable con nombre `x_pos`.

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

El ejemplo siguiente equivale al anterior, pero se utiliza la función `getProperty()` para recuperar la posición *x* del clip de película:

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

La función `setProperty()` permite definir una propiedad de una instancia de clip de película, como se muestra en el ejemplo siguiente:

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

El ejemplo siguiente equivale al anterior, pero se utiliza la sintaxis con punto:

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

También puede obtener o definir propiedades de clip de película desde una sentencia `tellTarget()`. El código siguiente equivale al ejemplo de `setProperty()` anterior:

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

Para más información sobre la función `tellTarget()`, consulte [“Control de otras líneas de tiempo” en la página 10](#).

## Control de otras líneas de tiempo

Para especificar la ruta correspondiente a una línea de tiempo y crear la referencia de la ruta, utilice la sintaxis de barras (/) combinada con puntos (..). También puede usar `_levelM`, `_root` o `_parent`, respectivamente, de la notación de Flash 5 como referencia, un nivel de película específico, la línea de tiempo raíz de la aplicación o la línea de tiempo principal.

Por ejemplo, supongamos que existe una instancia de clip de película denominada `box` en la línea de tiempo principal del archivo SWF. A su vez, la instancia `box` contiene otra instancia de clip de película denominada `cards`. En el ejemplo siguiente se especifica el destino del clip de película `cards` desde la línea de tiempo principal:

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

En el ejemplo siguiente se especifica el destino de la línea de tiempo principal desde el clip de película `cards`:

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

En el ejemplo siguiente se especifica el destino del clip de película principal `cards`:

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

## Utilización de variables

Para especificar una variable en una línea de tiempo, utilice la sintaxis de barras (/) combinada con puntos (..) y dos puntos (:). También puede usar la notación con puntos.

El código siguiente se refiere a la variable `car` de la línea de tiempo principal:

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

En el ejemplo siguiente se muestra la variable `car` en una instancia de clip de película que reside en la línea de tiempo principal:

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

En el ejemplo siguiente se muestra la variable `car` en una instancia de clip de película que reside en la línea de tiempo actual:

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

## Emulación de matrices

Las matrices son útiles para crear y manipular listas ordenadas de información, como variables y valores. Sin embargo, Flash Lite 1.1 no admite estructuras de datos de matrices nativas. Una técnica común en la programación de Flash Lite (y Flash 4) consiste en emular matrices con procesamiento de cadenas. A las matrices emuladas también se las conoce como pseudo-matrices. La clave para procesar pseudo-matrices es la función `eval()` de ActionScript, que permite acceder a las variables, las propiedades o los clip de película por su nombre. Para más información, consulte [“Utilización de la función eval\(\)” en la página 18](#).

Una pseudo-matriz típica consta de dos o más variables que comparten el mismo nombre base, seguido de un sufijo numérico. El sufijo es el índice correspondiente a cada elemento de la matriz.

Por ejemplo, suponga que crea las siguientes variables de ActionScript:

```
color_1 = "orange";
color_2 = "green";
```

```
color_3 = "blue";  
color_4 = "red";
```

Luego puede utilizar el código siguiente para reproducir indefinidamente los elementos de la pseudo-matriz:

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

Además de establecer referencias con las variables existentes, puede utilizar la función `eval()` de la parte izquierda de una asignación de variables para crear variables en tiempo de ejecución. Por ejemplo, suponga que desea mantener una lista de los puntos que se anota un usuario mientras juega. Cada vez que el jugador complete una serie, añada los puntos a la lista:

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

Con cada ejecución de este código se añade un nuevo elemento a la lista de puntos y la variable `scoreIndex` aumenta, lo que determina el índice de cada elemento de la lista. Por ejemplo, puede que al final tenga las siguientes variables:

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

## Trabajo con texto y cadenas

Flash Lite proporciona algunos comandos y propiedades de ActionScript básicos para trabajar con texto. Puede obtener y definir valores de campos de texto, concatenar cadenas, codificar o decodificar cadenas URL de texto y crear campos de texto desplazable.

Esta sección contiene los siguientes temas:

- [“Concatenación de cadenas” en la página 12](#)
- [“Texto desplazable” en la página 13](#)

## Concatenación de cadenas

Para concatenar cadenas en Flash Lite, utilice el operador `add`, como se muestra en el ejemplo siguiente:

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

## Texto desplazable

Para obtener o definir la posición de desplazamiento actual del campo, puede utilizar la propiedad `scroll` de los campos de texto dinámico y de introducción de texto. También puede utilizar la posición `maxscroll` para determinar la posición de desplazamiento actual de un campo de texto en función de la posición de desplazamiento máxima. Para ver un ejemplo de cómo crear un campo de texto desplazable, consulte “Creación de texto desplazable (sólo Flash Professional)” en *Desarrollo de aplicaciones Flash Lite*.

## Utilización de la función `call()` para crear funciones

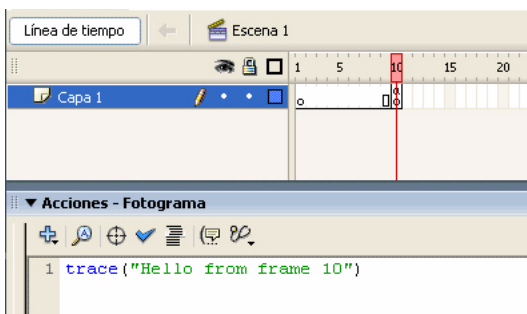
En Flash Lite no es posible definir o llamar funciones personalizadas de la misma forma que en Flash Player 5 y versiones posteriores. Sin embargo, puede utilizar la función `call()` de ActionScript para ejecutar código que reside en un fotograma arbitrario de la línea de tiempo. Esta técnica permite encapsular código de uso frecuente en un solo lugar, lo que facilita el mantenimiento.

La función `call()` toma como parámetro un número o una etiqueta de fotograma. Por ejemplo, ActionScript llama al código situado en el fotograma que lleva la etiqueta `moveUp`:  
`call("moveUp");`

La función `call()` funciona de forma sincronizada; el código de ActionScript que sigue a la función `call()` no se ejecuta hasta que termina de ejecutarse el código incluido en el fotograma especificado.

### Para llamar a ActionScript en otro fotograma:

1. En un documento de Flash nuevo, inserte un fotograma clave en el fotograma 10.



2. Con el nuevo fotograma clave seleccionado, abra el panel Acciones (Ventana > Acciones) e introduzca el código siguiente:

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

3. Seleccione el fotograma clave en el fotograma 1 e introduzca este código en el panel Acciones:

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

Este código detiene la cabeza lectora en el fotograma 1 y luego llama al código del fotograma 10.

4. Pruebe la aplicación en el emulador y abra el panel Salida (Ventana > Salida).  
En el panel Salida debería aparecer "Hello from frame 10".

También puede llamar al código de otra línea de tiempo, como la línea de tiempo de un clip de película. Para ejecutar el código, especifique el nombre de la instancia del clip de película seguido de dos puntos y, a continuación, del número o la etiqueta del fotograma. Por ejemplo, con el siguiente ActionScript se llama al código que reside en el fotograma etiquetado `moveUp` en la instancia del clip de película con nombre `callClip`:

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

Esta técnica se suele utilizar para crear *clips de llamada* o *clips de función*; es decir, clips de película cuya única finalidad es encapsular código usado periódicamente. El clip de llamada contiene un fotograma clave para cada función que desee crear. Por lo general, cada fotograma clave se etiqueta en función de su finalidad. Macromedia también recomienda crear una capa nueva para cada fotograma clave nuevo, y dar a cada capa el nombre incluido en la etiqueta del fotograma asignado al fotograma clave.

En la figura siguiente se muestra la línea de tiempo de un clip de llamada de ejemplo. El primer fotograma clave de un clip de llamada siempre contiene una acción `stop()`, que impide que la cabeza lectora se mueva continuamente sobre los fotogramas de su línea de tiempo. Los fotogramas siguientes contienen código para cada “función”. A cada fotograma clave de función se asigna una etiqueta que identifica lo que hace. Para facilitar la edición y visualización del clip de llamada, cada fotograma clave de función se inserta en una capa separada.



En el procedimiento siguiente se explica la forma de crear y utilizar un clip de llamada.

**Para crear y utilizar un clip de llamada:**

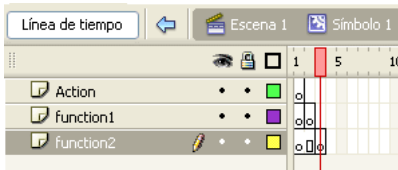
1. En Flash Professional 8, cree un nuevo documento a partir de la plantilla de documentos Flash Lite 1.1 Symbian Series 60.
2. Seleccione Insertar > Nuevo símbolo.
3. En el cuadro de diálogo Crear un nuevo símbolo, escriba **Call Clip** en el cuadro de texto Nombre y haga clic en Aceptar.

El clip de película se abre en el modo de edición.

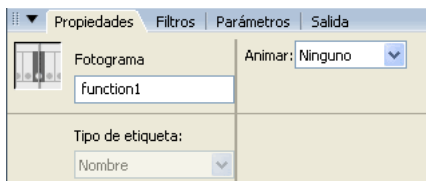
4. En la línea de tiempo, haga clic dos veces en el botón Insertar capa para añadir dos capas nuevas.

Denomine a la capa superior **Actions**, la segunda capa **function1** y la tercera capa **function2**.

5. Inserte un fotograma clave en el fotograma 2 de la capa function1 y otro fotograma clave en el fotograma 3 de la capa function2, como se muestra:

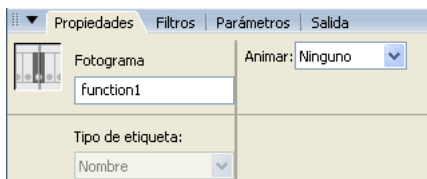


6. Seleccione el fotograma clave de la capa Actions y abra el panel Acciones.
7. Añada una acción `stop()` al panel Acciones.
8. Seleccione el fotograma clave situado en el fotograma 2 de la capa function1 y realice lo siguiente:
  - a. En el inspector de propiedades, escriba **function1** en el cuadro de texto Etiqueta de fotograma.



- b. En el panel Acciones (Ventana > Acciones), especifique el siguiente código:

```
trace("function1 was called.");
```
9. Seleccione el fotograma clave en el fotograma 3 de la capa function2 y realice lo siguiente:
    - a. En el inspector de propiedades, escriba **function2** en el cuadro de texto Etiqueta de fotograma.



- b. En el panel Acciones (Ventana > Acciones), especifique el siguiente código:

```
trace("function2 was called.");
```
10. Presione Ctrl+E (Windows) o Comando+E (Macintosh) para regresar a la línea de tiempo principal.

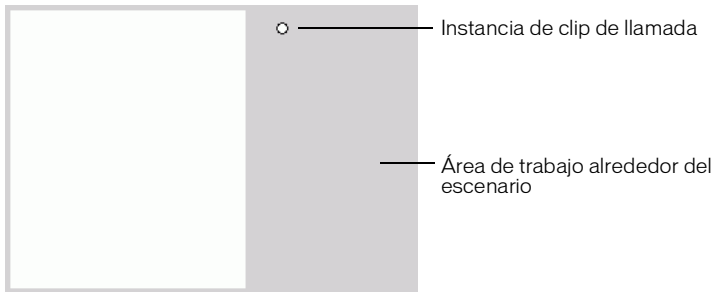


11. Defina la vista del documento para que incluya el área de trabajo en torno al escenario seleccionando Ver > Área de trabajo.

Como no es necesario que el clip de llamada esté visible, puede situarlo en el área de trabajo.

12. Abra el panel Biblioteca (Ventana > Biblioteca) y arrastre el símbolo del clip de llamada hasta el área de trabajo alrededor del escenario.

Como el clip de llamada no contiene elementos visuales, aparece en el escenario como un círculo pequeño que representa el punto de registro del clip de película.



#### SUGERENCIA

Para facilitar la identificación del clip de llamada en el escenario, añada texto u otro elemento visual al primer fotograma clave de la línea de tiempo del clip de llamada.

13. En el inspector de propiedades, escriba **callClip** en el cuadro de texto Nombre de instancia.
14. En la línea de tiempo, seleccione el fotograma 1 en la capa denominada ActionScript.
15. En el panel Acciones, introduzca el siguiente código:

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

16. Pruebe la aplicación en el emulador (Control > Probar película).

Debería ver el siguiente texto en el panel Salida:

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

## Utilización de la función eval()

La función `eval()` permite establecer referencias dinámicas con variables e instancias de clip de película en tiempo de ejecución. La función `eval()` toma una expresión de cadena como parámetro y devuelve el valor de la variable representada por esa expresión o una referencia al clip de película.

Por ejemplo, el código siguiente evalúa el valor de la variable `name` de `ActionScript` y asigna el resultado a `nameValue`:

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

La función `eval()` se suele utilizar con bucles `for()` y el operador `add` (concatenación de cadenas) para crear matrices basadas en cadenas, dado que `Flash Lite` no admite estructuras de datos de matrices nativas. Para más información, consulte [“Emulación de matrices” en la página 11](#).

La función `eval()` también se puede usar para hacer referencia a instancias de clip de película por su nombre. Por ejemplo, suponga que tiene tres clips de película con nombre `clip1`, `clip2` y `clip3`. El siguiente bucle `for()` incrementa la posición `x` de cada clip en 10 píxeles:

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

# Tareas comunes de creación de scripts

En este capítulo se tratan las tareas comunes de creación de scripts de Flash Lite para trabajar con el dispositivo del usuario. Se incluye, por ejemplo, la obtención de información sobre las características del dispositivo, el inicio de llamadas telefónicas y mensajes de texto y la determinación del estado de la red.

Este capítulo contiene los siguientes temas:

Determinación de las características del dispositivo y la plataforma .....	20
Apertura de una página Web .....	20
Inicio de una llamada telefónica .....	21
Inicio de un mensaje de texto o multimedia .....	22
Inicio de un mensaje de correo electrónico .....	23
Carga de archivos SWF externos .....	23
Carga de datos externos .....	24

# Determinación de las características del dispositivo y la plataforma

Flash Lite 1.1 incluye una serie de variables de ActionScript que proporcionan información acerca de las funciones y características disponibles en las aplicaciones Flash Lite instaladas en un dispositivo determinado. Por ejemplo, la variable `_capLoadData` indica si el dispositivo admite la carga de datos externos y la variable `_capSMS`, si el dispositivo permite enviar mensajes SMS (servicio de mensajes cortos). Para consultar la lista completa de las variables de función, consulte “Capacidades” en *Referencia del lenguaje ActionScript de Flash Lite 1.x*.

Normalmente, las variables de función se utilizan para determinar si un dispositivo admite una función específica antes de intentar usarla. Por ejemplo, suponga que quiere desarrollar una aplicación para descargar datos de un servidor Web mediante la función `loadVariables()`. Antes de intentar cargar los datos, puede comprobar el valor de la variable `_capLoadData` del siguiente modo para determinar si el dispositivo admite esa función:

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

Flash Lite define las variables de función en la línea de tiempo raíz del archivo SWF principal. Por consiguiente, para acceder a estas variables desde otra línea de tiempo (por ejemplo, la línea de tiempo de un clip de película) necesita introducir la ruta completa de la variable. Por ejemplo, en el ejemplo siguiente se emplea una barra (/) para especificar la ruta completa de la variable `_capSMS`.

```
canSendSMS = /:_capSMS
```

## Apertura de una página Web

Para abrir una página Web en el navegador del dispositivo se utiliza el comando `getURL()`. Se trata del mismo método utilizado para abrir una página Web desde una aplicación Flash de escritorio. Por ejemplo, el código siguiente abre la página Web de Macromedia:

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

Flash Lite sólo procesa una acción `getURL()` por fotograma o por controlador de eventos. En algunos teléfonos móviles, la acción `getURL()` se limita a eventos de pulsación de tecla exclusivamente, en cuyo caso la llamada a `getURL()` se procesa sólo si se activa en un controlador de eventos de pulsación de tecla. Incluso en estos casos se procesa una única acción `getURL()` por controlador de eventos de pulsación de tecla. Asociado a una instancia de botón en el escenario, el código siguiente abre una página Web cuando el usuario presiona el botón de selección del dispositivo:

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

## Inicio de una llamada telefónica

Para iniciar una llamada telefónica desde una aplicación Flash Lite se utiliza la función `getURL()`. Aunque esta función se suele utilizar para abrir una página Web, en este caso va a especificar `tel:` como protocolo (en lugar de `http`) y luego va a introducir el número de teléfono al que quiere llamar. Cuando se llama a esta función, Flash Lite muestra un cuadro de diálogo en el que se solicita confirmación para llamar al número especificado.

El código siguiente intenta iniciar la llamada al número 555-1212:

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

Flash Lite sólo procesa una acción `getURL()` por fotograma o por controlador de eventos. En algunos teléfonos móviles, la acción `getURL()` se limita a eventos de pulsación de tecla exclusivamente, en cuyo caso la llamada a `getURL()` se procesa sólo si se activa en un controlador de eventos de pulsación de tecla. Incluso en estos casos se procesa una única acción `getURL()` por controlador de eventos de pulsación de tecla. En el ejemplo siguiente, la llamada telefónica se inicia cuando el usuario presiona el botón de selección del dispositivo:

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

# Inicio de un mensaje de texto o multimedia

Puede utilizar Flash Lite para iniciar un mensaje del servicio de mensajes cortos (SMS) o del servicio de mensajes multimedia (MMS). Para iniciar un mensaje SMS o MMS en una aplicación Flash Lite, debe utilizar el comando `getURL()` y pasarlo al protocolo `sms:` o `mms:`, en lugar de usar el protocolo `http` estándar, seguido del número de teléfono al que quiere enviar el mensaje.

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

Si lo desea también puede especificar el cuerpo del mensaje en la cadena de consulta URL, como se muestra en el código siguiente:

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

Para iniciar un mensaje MMS, debe utilizar el protocolo `mms:` en lugar de `sms:`, como sigue:

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

NOTA

No es posible adjuntar archivos al mensaje MMS desde Flash Lite.

Flash Lite sólo procesa una acción `getURL()` por fotograma o por controlador de eventos. En algunos teléfonos móviles, la acción `getURL()` se limita a eventos de pulsación de tecla exclusivamente, en cuyo caso la llamada a `getURL()` se procesa sólo si se activa en un controlador de eventos de pulsación de tecla. Incluso en estos casos se procesa una única acción `getURL()` por controlador de eventos de pulsación de tecla. En el ejemplo siguiente, el mensaje SMS se inicia cuando el usuario presiona el botón de selección del dispositivo:

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

# Inicio de un mensaje de correo electrónico

Flash Lite permite iniciar un mensaje de correo electrónico. Para esto, debe utilizar el comando `getURL()` y pasarlo al protocolo `mailto:` seguido de la dirección de correo electrónico del destinatario. También puede especificar el asunto del mensaje en la cadena de consulta URL, como se muestra a continuación:

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

Para especificar el cuerpo del mensaje solamente en la cadena de consulta, utilice el código siguiente:

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

## Carga de archivos SWF externos

La función `loadMovie()` permite agregar archivos SWF desde un archivo local o de la red. Esta característica sólo está disponible en Flash Lite 1.1 y versiones posteriores. Es necesario tener en cuenta las advertencias siguientes cuando se cargan archivos SWF externos:

- Flash Lite puede cargar otros archivos SWF de Flash Lite 1.0 o Flash Lite 1.1 y archivos SWF con formato de Flash 4 y versiones anteriores. Si se trata de cargar un archivo SWF con otro formato (por ejemplo, un archivo SWF de Flash Player 6), Flash Lite generará un error en tiempo de ejecución.
- Flash Lite no puede cargar directamente archivos externos de imágenes, como imágenes JPEG o GIF. Para cargar estos tipos de medios, es necesario convertir los datos de imagen al formato SWF. Puede hacerlo “manualmente” con la herramienta de edición de Flash importando el archivo de imagen en un nuevo documento y exportando el archivo como un archivo SWF de Flash Lite o Flash 4. Existen herramientas de otros fabricantes que realizan este tipo de conversión automáticamente.

Para más información sobre la carga de archivos SWF, consulte `loadMovie()` en *Referencia del lenguaje ActionScript de Flash Lite 1.x*.

# Carga de datos externos

Para cargar datos externos en una aplicación Flash Lite se utiliza la función `loadVariables()`. Es posible cargar datos desde la red (desde una dirección HTTP) o desde el sistema de archivos local. Esta característica sólo está disponible en Flash Lite 1.1 y versiones posteriores.

En esta sección se muestra cómo utilizar la función `loadVariables()` para cargar datos desde un archivo externo y mostrar los datos en campos de texto dinámicos. Primero, se creará el archivo de datos, un archivo de texto que contiene cinco pares de nombre-valor separados por símbolos ampersand (&). Después se creará la aplicación Flash Lite que carga y muestra los datos del archivo de texto.

En este ejemplo se presupone que el archivo de datos y el archivo SWF se encuentran en la misma carpeta, bien sea en un ordenador (cuando se prueba con el emulador) o en la tarjeta de memoria del dispositivo (cuando se prueba en un dispositivo real). Para probar la aplicación en el dispositivo, siga uno de estos procedimientos:

- Transfiera el archivo de texto al dispositivo y colóquelo en la misma carpeta que el archivo SWF.
- Transfiera el archivo de texto a una dirección URL de un servidor web (por ejemplo: `www.su-servidor.com/data.txt`). Después modifique la llamada `loadVariables()` en la aplicación de muestra para utilizar dicha dirección URL, tal como se indica a continuación:

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

Si desea ver un ejemplo de una aplicación que cargue datos a través de la red, consulte “Lector de noticias de Flash Lite” en *Muestras de Flash*.

## Para crear el archivo de datos:

1. Utilice un editor de texto (por ejemplo, Bloc de notas o SimpleText) para crear un archivo que contenga el texto siguiente:

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

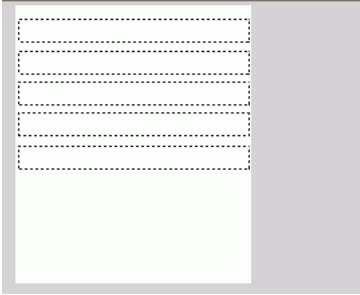
2. Guarde el archivo como `data.txt`.

## Para crear la aplicación Flash Lite que cargue los datos:

1. Cree un nuevo documento a partir de la plantilla Flash Lite 1.1 Symbian Series 60.  
Para más información sobre cómo utilizar plantillas de documentos de Flash Lite, consulte “Utilización de plantillas de documento de Flash Lite (sólo Flash Professional)” en *Primeros pasos con Flash Lite*.
2. Guarde el archivo como `dataloading fla` en la misma carpeta que contiene el archivo de texto (`data.txt`) creado anteriormente.



3. En la línea de tiempo, seleccione el fotograma 1 de la capa Content.
4. Con la herramienta Texto, cree cinco campos de texto dinámicos en el escenario, tal como aparece en la figura siguiente.



5. Seleccione el primer campo de texto (superior) y, en el inspector de propiedades, escriba `item_1` en el cuadro de texto Var.  
Este nombre de variable coincide con el nombre de la primera variable definida en el archivo `data.txt` que ha creado anteriormente (`item_1=Hello`).
6. Al igual que se ha descrito anteriormente en los dos pasos anteriores, asigne a los cuatro campos de texto restantes los nombres de variable `item_2`, `item_3`, `item_4` e `item_5`.
7. Seleccione todos los campos manteniendo presionada la tecla Mayús y elija Modificar > Convertir en símbolo.
8. En el cuadro de diálogo Convertir en símbolo, seleccione Clip de Película como tipo de símbolo y haga clic en Aceptar.
9. Seleccione el clip de película que acaba de crear y, en el inspector de propiedades, escriba **data\_clip** en el cuadro de texto Nombre de instancia.
10. En la línea de tiempo, seleccione el fotograma 1 de la capa Actions y abra el panel Acciones (Ventana > Acciones).
11. Introduzca el código siguiente en el panel Acciones:  

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

12. Guarde las modificaciones (Archivo > Guardar) y, a continuación, pruebe la aplicación en el emulador (Control > Probar película).

Debe ver cada campo de texto con los datos del archivo de texto tal como se aprecia en la figura siguiente:



# Índice alfabético

## A

- abrir una página Web 20
- ActionScript de Flash Lite 1.x
  - características no disponibles en 7
  - diferencias entre 1.0 y 1.1 6
  - elementos de ActionScript de Flash 4 no admitidos 6
  - información general 5

## C

- cadenas de concatenación 12
- cadenas, concatenar 12
- cargar archivos SWF externos 23
- cargar datos externos 24
- clips de función, crear 13
- clips de película
  - hacer referencia dinámicamente 18
  - obtención y definición de propiedades 9

## F

- función call(), utilizar 13
- función eval(), utilizar 18
- función loadMovie(), utilizar 23
- función loadVariables(), utilizar 24
- funciones, emular con call() 13

## G

- getUrl(), función
  - abrir páginas Web con 20
  - iniciar un mensaje de correo electrónico con 23
  - iniciar un mensaje de texto con 22
  - iniciar un mensaje multimedia con 22
  - iniciar una llamada telefónica con 21

## I

- iniciar llamadas de teléfono 21
- iniciar mensajes 22

## L

- líneas de tiempo, control con ActionScript 10
- llamada telefónica, iniciar 21

## M

- matrices, emular con cadenas 11
- mensaje de correo electrónico, iniciar 23
- mensaje de texto, iniciar 22
- mensaje multimedia, iniciar 22

## P

- página Web, abrir 20

## T

- texto desplazable, crear 13

## V

- variables
  - hacer referencia dinámicamente 18
  - referencias 11
  - sintaxis de puntos y sintaxis de barras 11
- variables de función de plataforma, información 20

