

Desarrollo de aplicaciones de FLASH[®] LITE[™] 4

Última modificación 29/4/2010

© 2010 Adobe Systems Incorporated and its licensors. All rights reserved.

Desarrollo de aplicaciones de Adobe® Flash® Lite® 4

This guide is protected under copyright law, furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

This guide is licensed for use under the terms of the Creative Commons Attribution Non-Commercial 3.0 License. This License allows users to copy, distribute, and transmit the user guide for noncommercial purposes only so long as (1) proper attribution to Adobe is given as the owner of the guide; and (2) any reuse or distribution of the user guide contains a notice that use of the guide is governed by these terms. The best way to provide notice is to include the following link. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>

Adobe, the Adobe logo, ActionScript, Flash, and Flash Lite are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

ActiveX and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Macintosh is a trademark of Apple Inc., registered in the United States and other countries. Symbian and all Symbian based marks and logos are trademarks of Symbian Limited. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries. Arial is a trademark of The Monotype Corporation registered in the U.S. Patent and Trademark Office and certain other jurisdictions. Helvetica is a trademark of Heidelberger Druckmaschinen AG exclusively licensed through Linotype Library GmbH, and may be registered in certain jurisdictions. All other trademarks are the property of their respective owners.

Updated Information/Additional Third Party Code Information available at <http://www.adobe.com/go/thirdparty>.

Portions include software under the following terms:

MPEG Layer-3 audio compression technology licensed by Fraunhofer IIS and Thomson Multimedia <http://www.iis.fhg.de/amm/>).

This software is based in part on the work of the Independent JPEG Group.

 Sorenson Spark™ video compression and decompression technology licensed from Sorenson Media, Inc.

Video compression and decompression in Flash is powered by On2 TrueMotion video technology. © 1992-2005 On2 Technologies, Inc. All Rights Reserved. <http://www.on2.com>.

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

Notice to U.S. Government End Users. The Software and Documentation are “Commercial Items,” as that term is defined at 48 C.F.R. §2.101, consisting of “Commercial Computer Software” and “Commercial Computer Software Documentation,” as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contenido

Capítulo 1: Información general sobre Flash Lite 4

Conceptos básicos de Flash Lite	1
Funciones de Flash Lite 4	2

Capítulo 2: Introducción a Flash Lite

Aplicación Flash Lite Hello World	5
El emulador de Adobe Device Central y la base de datos de dispositivos	6
Flujo de trabajo para la edición de aplicaciones de Flash Lite	6
Tipos de contenido de Flash Lite	7

Capítulo 3: Tutorial: Creación de una aplicación de Flash Lite

Visualización de la aplicación finalizada	9
Creación de la aplicación	9

Capítulo 4: Creación de interactividad y navegación

Interacción del usuario y teclas admitidas	15
Navegación predeterminada	16
Administración de eventos de tecla y de ratón	22
Compatibilidad con la tecnología multitáctil	28

Capítulo 5: Utilización de texto y fuentes

Texto	30
Introducción de texto	31
Motor de texto de Flash	38
Representación de fuentes	39
Texto desplazable	42

Capítulo 6: Trabajo con sonido

Sonido de dispositivo	44
Sonido nativo	45

Capítulo 7: Trabajo con vídeo

Trabajo con archivos FLV	47
RTMP	49
Mejoras	50

Capítulo 8: Trabajo con las API del dispositivo

Localización geográfica	54
Acelerómetro	55

Capítulo 9: Introducción a ActionScript en Flash Lite 4

Clases y elementos de idioma de ActionScript admitidos, parcialmente admitidos y no admitidos	58
Clases de ActionScript admitidas parcialmente: detalles	148
Elementos de Actionscript no admitidos: detalles	162

Capítulo 1: Información general sobre Flash Lite 4

Adobe® Flash Lite™ es una versión de Adobe® Flash® Player® diseñada para dispositivos.

Conceptos básicos de Flash Lite

Acerca de Flash Lite

Flash Lite 4 se basa en Flash Player 10. Ofrece funciones para mejorar la experiencia de navegación por Internet para usuarios de dispositivos móviles. Muchas de estas funciones estaban incluidas en Flash Player 9 pero no se admitían en Flash Lite; ahora se admiten en Flash Lite 4. Flash Lite 4 introduce la capacidad de reproducir contenido de Adobe® ActionScript™ 3.0 cuando se encuentra en sitios web de Flash 10. No obstante, Flash Lite 4 no admite el procesamiento y la reproducción de ActionScript 3.0, que se introdujo en Flash Player 9. Para obtener más información, consulte [“Funciones de Flash Lite 4”](#) en la página 2.

Acerca de los componentes de Flash Lite

Los componentes instalados con Adobe® Flash® Professional CS5 (por ejemplo, DataGrid) fueron designados para su uso en aplicaciones de escritorio de Flash. Debido a sus requisitos de memoria y de capacidad de procesamiento, no suele ser posible utilizarlos en las aplicaciones de Flash Lite. Adobe desaconseja el uso de componentes de interfaz de usuario estándar en las aplicaciones de Flash Lite.

Optimización del contenido para mejorar el rendimiento y la memoria

Para optimizar el contenido en dispositivos móviles, se deben tener en cuenta unos principios básicos de desarrollo de contenido. Por ejemplo, con frecuencia los desarrolladores de Flash han tenido que evitar ilustraciones complejas, interpolación excesiva y el uso abusivo de las transparencias.

En ocasiones, los desarrolladores de Flash Lite deben enfrentarse a algunos retos debido a las limitaciones de los dispositivos móviles. Algunos dispositivos funcionan mejor que otros y a veces la diferencia es espectacular. La creación de contenido móvil suele requerir la publicación en diversos dispositivos distintos. Como resultado, los desarrolladores crean dispositivos que cuentan con las funciones mínimas.

La optimización de contenido para dispositivos móviles suele implicar dificultades en el proceso de creación. Por ejemplo, una técnica puede presentar un mejor aspecto, mientras que otra puede dar como resultado un mejor rendimiento. Al cuantificar estos aspectos, se repiten continuamente las pruebas tanto en el emulador y como en el dispositivo de destino.

Adobe Flash Professional CS5 incluye un emulador de Adobe Flash Lite disponible en Adobe® Device Central CS5. Este emulador permite probar la aplicación en la herramienta de edición tal y como aparece y funciona en un dispositivo real. Cuando esté satisfecho con la aplicación en el emulador, puede probarla en el dispositivo real.

Funciones de Flash Lite 4

Interactividad y desplazamiento mejorados

Flash Lite 4 ofrece distintas funciones para mejorar la interactividad y el desplazamiento de los usuarios de dispositivos móviles. Algunas de estas funciones se encuentran en las versiones anteriores de Flash Player, pero no se admiten en Flash Lite. Estas funciones son ahora compatibles con Flash Lite 4.

Compatibilidad con la tecnología multitáctil

Flash Lite 4 introduce la compatibilidad con la tecnología multitáctil para administrar eventos multitáctiles y eventos de gestos nativos. Todos los dispositivos móviles que admiten esta función responden a varios puntos de toque en pantalla. Asimismo, una serie de datos entrantes del punto de toque se puede procesar en gestos de nivel más elevado.

Mejoras de texto y fuente

Motor de texto de Flash Player 10

Flash Lite 4 admite el nuevo motor de diseño de texto de Flash Player 10. Este motor admite un lenguaje complejo en dos direcciones, la fuente TrueType y funciones avanzadas como, por ejemplo, rotación, enmascaramiento, etc. El nuevo motor se ha desarrollado para optimización de memoria y alto rendimiento. Para obtener más información, consulte [“Motor de texto de Flash”](#) en la página 38.

Utilización de la introducción de texto en línea

En Flash Lite 4, los eventos de fondo pueden continuar durante la introducción de texto en línea, lo cual no era posible en las versiones anteriores. Asimismo, con esta versión, el modo restringido no se aplica y ActionScript se procesa durante la introducción de texto en línea. Para obtener más información, consulte [“Utilización de la introducción de texto en línea”](#) en la página 32.

Mejoras de vídeo

Canal de datos RTMP

ActionScript 3.0 utiliza el formato AMF 3 para serializar y deserializar los objetos de ActionScript. Este formato envía cadenas y características de objetos junto con instancias de objetos. Asimismo, AMF 3 admite algunos de los tipos de datos introducidos en ActionScript 3.0. Para obtener más información, consulte [“RTMP”](#) en la página 49.

RTMPE

Flash Lite 4 es compatible con los flujos RTMP cifrados. Esto significa que Flash Lite solicita una versión cifrada de un flujo de medios desde Adobe® Flash® Media Server 3. Flash Lite descifra posteriormente ese flujo en el dispositivo móvil. Para obtener más información, consulte [“RTMPE”](#) en la página 51.

RTMPT y RTMPTE

En Flash Lite 4, RTMPT y RTMPTE están habilitados mediante HTTP. Como resultado, las transmisiones de paquetes RTMP se activan para los usuarios que intenten conectarse tras un firewall que bloquee los intentos de transmisión. Para obtener más información, consulte [“RTMPT”](#) en la página 51 y [“RTMPTE”](#) en la página 52.

Transmisión con varias velocidades de bits

Flash Lite 4 admite el cambio de velocidad de bits dinámica cuando se transmite desde Flash Media Server. Como resultado, la aplicación puede adaptarse a los cambios dinámicos en las condiciones de la red mientras se transmite el contenido grabado. Para obtener más información, consulte “[Transmisión con varias velocidades de bits](#)” en la página 52.

Nuevas API de dispositivo

Flash Lite 4 permite obtener la localización geográfica y los valores de aceleración direccional de un dispositivo móvil con la introducción de dos API de dispositivo.

Localización geográfica

Esta función permite registrarse para las actualizaciones de localización desde el sensor de localización integrado de un dispositivo móvil. El dispositivo devuelve su localización geográfica en forma de coordenadas latitudinales y longitudinales. Para obtener más información, consulte “[Localización geográfica](#)” en la página 54.

Acelerómetro

El acelerómetro permite obtener los datos de aceleración de un dispositivo mediante un sensor del acelerómetro instalado en el dispositivo. Estos datos representan la localización del dispositivo a lo largo de un eje tridimensional o el movimiento del dispositivo a lo largo del eje tridimensional. Para obtener más información, consulte “[Acelerómetro](#)” en la página 55.

Cambios de comportamiento del objeto compartido

La implementación del objeto compartido en Flash Lite 4 es totalmente compatible con los archivos SWF de Flash Player 10.

Flash Lite 4 admite todos los métodos de la clase SharedObject de ActionScript 3.0. Un archivo SWF se ejecuta del mismo modo en Flash Lite 4 que en Flash Player 10. La única diferencia es que no existe administrador de configuración en el dispositivo. Como resultado, el tamaño de almacenamiento de un objeto compartido se limita a menos de 100 kb por dominio. Si el tamaño de almacenamiento aumenta a un valor superior a 100 kb, se producirá un error en las transacciones de objeto compartido.

Generalmente, los dispositivos móviles disponen de menos capacidades de almacenamiento y memoria. Al limitar el tamaño de almacenamiento, se reduce el almacenamiento en disco y la carga de acceso en estos dispositivos. Para obtener más información sobre los objetos remotos compartidos, consulte www.help.adobe.com.

Nota: cualquier contenido nuevo de ActionScript 3.0 se debe editar conforme a la documentación de Flash Player 10.

Objetos compartidos remotos

Con Flash Lite 4, es posible crear y utilizar objetos compartidos remotos. Estos objetos se comparten en tiempo real con todos los clientes conectados a la aplicación. Si un cliente cambia la propiedad de un objeto compartido remoto, dicho cambio se aplica para todos los demás clientes que estén conectados.

Para obtener más información sobre los objetos remotos compartidos, consulte www.help.adobe.com.

Extensiones de ActionScript

La función Extensions de Flash Lite 4 permite que otros desarrolladores añadan funciones y API de ActionScript a Flash Lite mediante una biblioteca vinculada de manera estática o dinámica. La DLL proporciona la implementación de las funciones, y Flash Lite permite que se use la DLL llamando a las API de ActionScript. Como resultado, es posible interactuar con las API que se expongan mediante los dispositivos de host. Para ello, escriba una DLL que conecte las capacidades del dispositivo y Flash Player. Estas extensiones se pueden vincular de manera estática o dinámica a la implementación del host. Por ejemplo, si usan esta capacidad, los desarrolladores pueden escribir aplicaciones que accedan al sistema de archivos del dispositivo, o bien usar los datos de GPS proporcionados por el dispositivo, y así consecutivamente.

Nota: la función *Extensions de ActionScript* está disponible para otros desarrolladores sólo si la ha activado el fabricante del dispositivo de destino.

Comprobación y depuración

Adobe Device Central CS5 permite probar y depurar los proyectos de Flash Lite en varios dispositivos. Es posible emular la pantalla del dispositivo, el uso de memoria y el rendimiento en dispositivos específicos.

Clases con compatibilidad parcial

Para obtener una lista detallada de las clases que se admiten parcialmente en esta versión, consulte [“Introducción a ActionScript en Flash Lite 4”](#) en la página 58.

Clases no admitidas

Para obtener una lista detallada de las clases que no se admiten en esta versión, consulte [“Introducción a ActionScript en Flash Lite 4”](#) en la página 58.

Capítulo 2: Introducción a Flash Lite

Aplicación Flash Lite Hello World

En esta sección, creará una sencilla aplicación de Flash Lite y la probará en el emulador de Adobe Device Central CS5. Para ver una aplicación de ejemplo más completa, consulte “[Tutorial: Creación de una aplicación de Flash Lite](#)” en la página 9.

En este tutorial, se presupone que desea desarrollar contenido para el reproductor autónomo de Flash Lite. El procedimiento utiliza un dispositivo genérico.

Primero, decida los dispositivos y el tipo de contenido de Flash Lite que va a utilizar.

Para configurar y crear una sencilla aplicación de Flash Lite

- 1 Inicie Flash.
- 2 En la pantalla principal de Flash, seleccione Crear nuevo > Flash Lite 4.
- 3 En el panel Herramientas, seleccione la herramienta Texto y arrastre para crear un campo de texto en el escenario. Escriba **Hello, world!** (o cualquier otro texto) en el cuadro de texto.
- 4 En la ficha Propiedades, seleccione Texto clásico en el menú emergente Motor de texto y Texto estático en el menú emergente Tipo de texto.
- 5 Seleccione Control > Probar película > en Device Central para exportar su aplicación a Adobe Device Central y ver su aplicación en el emulador de Adobe Device Central.

***Nota:** durante la evaluación en Device Central, puede cambiar el dispositivo y el tipo de contenido para ver su aplicación en una plataforma diferente. Para ello, haga doble clic en un dispositivo en el panel Biblioteca y seleccione un nuevo tipo de contenido en Tipo de contenido. Cuando regresa a Flash, éste recuerda las configuraciones que utilizó la última vez en el emulador.*

Configuración de ajustes importantes para aplicaciones móviles

Determinados ajustes requieren una consideración especial para las aplicaciones de Flash Lite 4 en dispositivos móviles. Los valores más recomendados se establecen automáticamente cuando se selecciona Crear nuevo > Flash Lite 4 en la pantalla de bienvenida, o bien, Archivo > Nuevo > Flash Lite 4 en el menú.

Definición de las propiedades del documento

- 1 En Flash Professional, vaya al escenario. Haga clic en una parte no utilizada del escenario de modo que no se seleccione ningún componente.
- 2 Abra la ventana Propiedades, presionando Ctrl+F3, o bien, seleccione Ventana > Propiedades en el menú. Si se ha anulado correctamente la selección de todos los componentes, el panel superior de la ventana Propiedades muestra el título “Documento” y el nombre del proyecto.
- 3 En la ventana Propiedades, expanda el área del mismo nombre.
- 4 En FPS, seleccione una velocidad de fotogramas adecuada para el dispositivo de destino. La velocidad del procesador, la disponibilidad de la aceleración de hardware y la cantidad de memoria puede afectar al rendimiento de la reproducción de vídeo. Los OEM del dispositivo pueden proporcionar la información necesaria para optimizar esta configuración para un dispositivo determinado.

- 5 Establezca el tamaño para ajustarse a las dimensiones de la pantalla del dispositivo predeterminado. Este valor define el tamaño predeterminado. Para otros dispositivos, se puede cambiar el tamaño dinámicamente en tiempo de ejecución utilizando la clase `flash.display.Stage`.

Cómo definir la configuración de publicación

- 1 Seleccione Archivo > Configuración de publicación para abrir el cuadro de diálogo homónimo.
- 2 En la ficha Formatos, seleccione HTML (.html) para crear un archivo SWF incorporado que se ejecute en un navegador.
- 3 En la ficha Flash, seleccione Flash Lite 4.0 de la lista de versiones y, después, seleccione ActionScript 3.0 de la lista de versiones de ActionScript.

El emulador de Adobe Device Central y la base de datos de dispositivos

El emulador de Adobe Device Central le permite probar el contenido tal como se ejecutará y se mostrará en un dispositivo real. Puede seleccionar un dispositivo o un tipo de contenido de evaluación diferente y ver información detallada acerca de su aplicación.

Device Central incluye una exhaustiva base de datos de dispositivos móviles de entre los que puede seleccionar aquéllos con los que desea crear su propio entorno de evaluación.

Flujo de trabajo para la edición de aplicaciones de Flash Lite

La creación de contenido de Flash Lite es un proceso iterativo y requiere llevar a cabo los pasos siguientes:

Identificar el dispositivo de destino y el tipo de contenido de Flash Lite Los distintos dispositivos disponen de pantallas de diferentes tamaños, admiten diferentes formatos de sonido y ofrecen varias profundidades de color, entre otras características. Estos factores pueden influir en el diseño o la implementación de la aplicación.

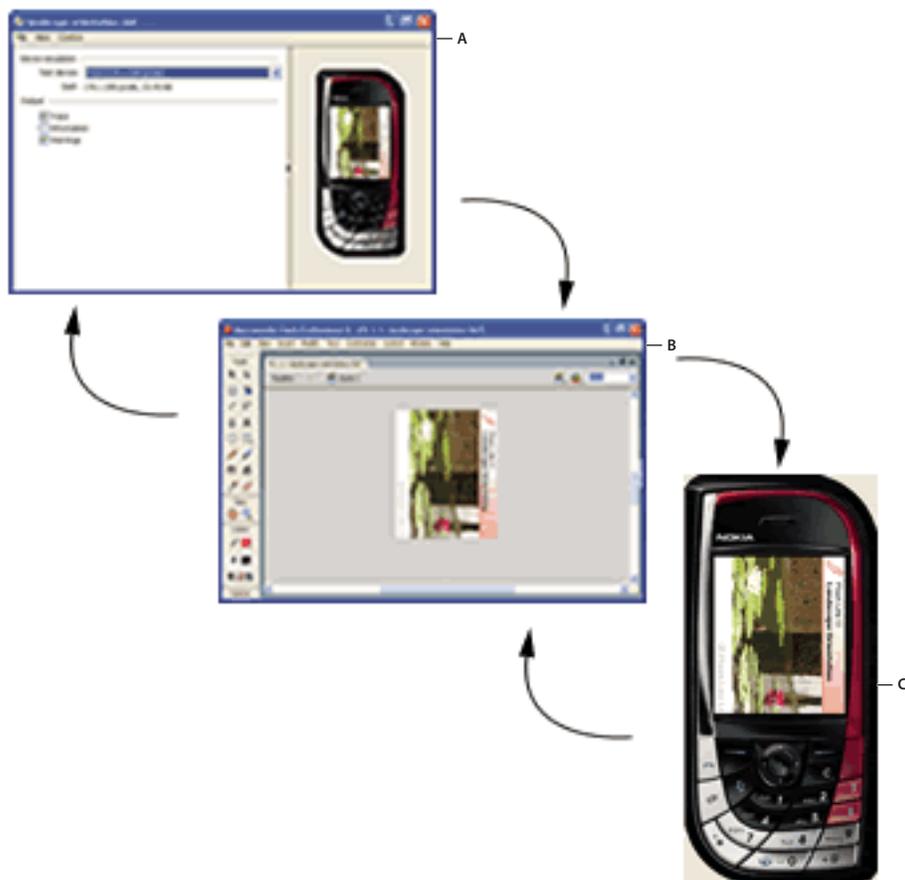
Además, cada dispositivo admite distintos tipos de contenido de Flash Lite, como protectores de pantalla, aplicaciones autónomas o tonos de llamada animados. El tipo de contenido que desarrolle también determinará las características disponibles en la aplicación.

Creación de la aplicación en Flash y prueba en Adobe Device Central Adobe Flash Professional CS5 incluye un emulador en Adobe Device Central CS que permite probar la aplicación sin tener que transferirla a un dispositivo. El emulador se utiliza para depurar el diseño de la aplicación y solucionar los posibles problemas antes de probarla en un dispositivo móvil.

Probar la aplicación en el dispositivo o los dispositivos de destino El emulador de Adobe Device Central no puede emular todos los aspectos de los dispositivos de destino, como la velocidad del procesador, la profundidad de color y la latencia de la red. Observe el ejemplo siguiente. Supongamos que una animación se ejecuta sin problemas en el emulador. No se puede esperar que se ejecute tan rápido en el dispositivo debido a su velocidad de procesador más lenta. Por ejemplo, es posible que un degradado de color que aparece correctamente en el emulador se vea con bandas

en el dispositivo real. Después de probar la aplicación en el dispositivo, es posible que deba depurar el diseño de la aplicación en la herramienta de edición de Flash.

En la figura siguiente se muestra el proceso iterativo de desarrollo y prueba:



A. Edición del archivo FLA en Adobe Flash B. Comprobación en el emulador de Adobe Device Central C. Comprobación en un dispositivo

Tipos de contenido de Flash Lite

Antes de comenzar a desarrollar una aplicación de Flash Lite, recuerde lo siguiente:

- El dispositivo o dispositivos en los que desea ejecutar el contenido (*dispositivos de destino*). El reproductor de Flash Lite se instala en diversos dispositivos. Para obtener una lista de los dispositivos con Flash Lite instalado, consulte la página Supported Devices (Dispositivos admitidos) del sitio web de Adobe, en www.adobe.com/go/mobile_supported_devices_es.
- Los tipos de contenido de Flash Lite que admiten los dispositivos de destino. Cada instalación de Flash Lite admite uno o varios modos de aplicación (*tipos de contenido*). Por ejemplo, algunos dispositivos utilizan Flash Lite para activar protectores de pantalla o tonos de llamada animados que están basados en Flash. Otros utilizan Lite para mostrar contenido de Flash que se encuentra incorporado en las páginas Web para móviles. Algunos tipos de contenido no admiten todas las características de Flash Lite.

Cada tipo de contenido de Flash Lite, junto con un dispositivo concreto, establece el conjunto específico de características de Flash Lite que estarán disponibles en la aplicación. Por ejemplo, una aplicación de escritorio que se ejecute como un protector de pantalla normalmente no podrá realizar conexiones de red ni descargar datos.

Las funciones de realización de pruebas con Flash Lite incluidas en Adobe Device Central permiten probar varios dispositivos y diferentes tipos de contenido de Flash Lite. Esto permite comprobar si la aplicación utiliza funciones que no están disponibles para el tipo de contenido que está desarrollando.

Capítulo 3: Tutorial: Creación de una aplicación de Flash Lite

En este tutorial, desarrollará una aplicación de Adobe Flash Lite. La aplicación adopta una fuente dinámica de un sitio web en línea, analiza el código XML y muestra la información en la pantalla de su dispositivo.

Visualización de la aplicación finalizada

Puede ver la aplicación en el emulador de Adobe Device Central. También puede transferir las aplicaciones finalizadas a la versión autónoma del reproductor Flash Lite instalada en un dispositivo móvil.

- 1 Descargue y descomprima los archivos de tutoriales. Vaya a www.adobe.com/go/learn_flt_samples_and_tutorials_es. Descargue el archivo ZIP de su versión de Flash Lite.
- 2 En Flash, seleccione Archivo > Abrir y elija el archivo denominado blogs-rss_complete fla en la carpeta Tutorial.
- 3 Seleccione Archivo > Configuración de publicación. En la ficha Flash del cuadro de diálogo Configuración de publicación, compruebe que Player se establece en Flash Lite 4.0.
- 4 Seleccione Control > Probar película > En Device Central para probar la aplicación con Adobe Device Central. Asegúrese de seleccionar el perfil de dispositivo de prueba Flash Lite 4.0 32 240x320.

Nota: esta sencilla aplicación tutorial no admite la funcionalidad multitáctil.

- 5 Para interactuar con la aplicación, realice lo siguiente:
 - En la pantalla principal, espere a que los titulares del blog se carguen en la pantalla del dispositivo. Haga clic en la tecla abajo para desplazarse por las entradas del blog. Haga clic en la tecla de selección para ver detalles sobre una entrada del blog.
 - Haga clic en la tecla abajo para desplazarse por las entradas completas. Presione 9 en el teclado del dispositivo para volver a la lista de titulares. Se puede presionar 7 para abrir la entrada del blog en un navegador web del dispositivo.

Creación de la aplicación

La aplicación tutorial lee una fuente RSS del sitio web de blogs de Adobe (blogs.adobe.com). Muestra un catálogo de resumen de las entradas del blog y permite al usuario ver el texto completo de cualquier entrada. Los usuarios también pueden optar por iniciar el navegador para ver una entrada en el sitio web.

Descarga de la fuente RSS

- 1 En Flash, elija Archivo > Abrir y seleccione el archivo denominado cafe_tutorial_start fla situado en la carpeta de tutoriales.
- 2 Elija Archivo > Guardar como y guarde el archivo como blogs-rss fla.
- 3 En la línea de tiempo (Ventana > Línea de tiempo), seleccione el fotograma 1 de la capa action y abra el panel Acciones (Ventana > Acciones).

- 4 Bajo la sentencia de importación en la línea 1, añade un componente preloader en el escenario. Este objeto en un marcador de posición gráfico que sólo aparece hasta que la aplicación analiza el contenido RSS XML:

```
var a1:preloader = new preloader();
```

- 5 Defina variables globales para su uso en toda la aplicación:

```
var myNews:Array;
var visi:Number=0;
var selectedNews=0;
var newsshowing:Sprite;
var isSet:Boolean=false;
```

Nota: los siguientes tres pasos no requieren ninguna acción por su parte. Simplemente describen lo que realiza el código del tutorial existente.

- 6 El siguiente código existente crea un objeto URLLoader y le añade detectores de eventos:

```
var loader:URLLoader = new URLLoader();
loader.addEventListener(Event.COMPLETE,loadxml);
loader.addEventListener(IOErrorEvent.IO_ERROR,ioerror);
loader.addEventListener(SecurityErrorEvent.SECURITY_ERROR,securityerror);
```

- 7 El siguiente código existente llama a la función URLLoader.load(), transmitiendo la URL de la fuente dinámica del blog mediante un objeto URLRequest:

```
loader.load(new URLRequest("http://blogs.adobe.com/index.xml"));
```

- 8 El siguiente código existente define funciones callback para eventos loader:

```
function loadxml(e):void{
}
function ioerror(e):void {
    trace(e.type);
}
function securityerror(e):void {
    trace(e.type);
}
```

Análisis de la fuente RSS descargada

En los siguientes pasos, añada el código indicado al cuerpo de la función loadxml():

- 1 Ahora que está listo para analizar el código XML y mostrar la fuente RSS, elimine el objeto preloader del escenario:

```
stage.removeChild(a1);
```

- 2 Cree un objeto XML y asigne la fuente RSS al objeto XML:

```
var xml:XML=new XML(loader.data);
```

- 3 Cree un conjunto y asígnelo a la variable global myNews:

```
myNews = new Array();
```

- 4 Analice el código XML para mostrar el título, la URL, descripción y fecha de publicación para la entrada del blog:

```
for each (var param:XML in xml.channel.item) {
    myNews.push({title:param.title,link:param.link,
        description:param.description,date:param.pubDate} );
}
```

La función loadxml() presenta ahora el siguiente aspecto:

```
function loadxml(e):void {
    stage.removeChild(a1);
    var xml=new XML(loader.data);
    myNews = new Array();
    for each (var param:XML in xml.channel.item) {
        myNews.push({title:param.title,link:param.link,
            description:param.description,date:param.pubDate});
    }
}
```

Creación de la pantalla principal

- 1 En la línea de tiempo, seleccione el fotograma 1 en la capa header. Haga clic en el panel Biblioteca y arrastre una instancia del objeto header a la parte superior del documento en el escenario.
- 2 Seleccione el fotograma 1 en la capa stage. Añada cuatro campos de texto dinámicos, alineados verticalmente, en los cuatro espacios bajo el objeto header. Para cada campo de texto, establezca la anchura en 230,00 y la altura en 45,25.
- 3 En el panel Propiedades, asigne los nombres tf1, tf2, tf3 y tf4 a las instancias de texto respectivamente, desde la parte superior a la inferior.
- 4 Seleccione el fotograma 1 en la capa action y abra el panel Acciones. Cree la función `showText()` para mostrar los títulos de entrada del blog en los campos de texto:

```
function showText():void {
    tf1.text=myNews[visi].title;
    tf2.text=myNews[visi+1].title;
    tf3.text=myNews[visi+2].title;
    tf4.text=myNews[visi+3].title;
}
```

Nota: la variable `visi` almacena el valor de índice de la entrada del blog más visible.

- 5 Llame a la función `showText()` en la línea final de la función `loadxml()`:

```
function loadxml(e):void {
    stage.removeChild(a1);
    var xml=new XML(loader.data);
    myNews = new Array();
    for each (var param:XML in xml.channel.item) {
        myNews.push({title:param.title,link:param.link,
            description:param.description,date:param.pubDate});
    }
    showText();
}
```

- 6 En el panel Biblioteca, arrastre una instancia del objeto `highlight` y sitúela en `tf1`. En la ficha Propiedades, indique `highlightBar` como nombre de instancia.

Visualización de la fecha y hora actuales en el encabezado

El componente de encabezado (header) contiene el título “Adobe blogs” (blogs de Adobe) en la parte izquierda y un campo de texto denominado `tf` a la derecha. Con los siguientes pasos, se añade una marca de fecha/hora a `tf`.

- 1 En el panel Biblioteca, haga doble clic en el componente de encabezado para abrirlo para su edición. El componente de encabezado se convierte en el único objeto del escenario.

- 2 Anule la selección de todos los elementos de encabezado, haciendo clic en un área del escenario sin ocupar. Abra el panel Acciones.
- 3 En el panel Acciones, añada un detector para el evento ENTER_FRAME. Para ello, pegue el siguiente código en la capa 1, fotograma 1 de la selección actual (componente header):

```
addEventListener(Event.ENTER_FRAME, changeDate);
```

- 4 En la siguiente línea, añada el código siguiente para implementar el controlador de eventos que muestra la fecha y la hora:

```
function changeDate(e):void {
    var date = new Date();
    tf.text = date.toString();
    tf.appendText("\n"+date.toLocaleTimeString());
}
```

Creación de una lista de desplazamiento de entradas del blog

- 1 Abra la escena 1 en el escenario.
- 2 En la línea de tiempo, seleccione el fotograma 1 en la capa scroller. En el panel Biblioteca, arrastre una instancia del objeto de desplazamiento hacia el borde derecho del documento del escenario. Coloque el objeto de desplazamiento de forma que su parte superior se sitúe en el borde derecho del encabezado. En la ficha Propiedades, indique scrollee como nombre de instancia.
- 3 Seleccione el fotograma 1 en la capa de acciones. En el panel Acciones, añada un detector de eventos al escenario para detectar eventos de teclado:

```
stage.addEventListener(KeyboardEvent.KEY_DOWN, onkeyDown);
```

- 4 Añada un controlador para el evento KEY_DOWN. En este controlador, se implementa la funcionalidad para el desplazamiento mediante los titulares del blog. También se añade la capacidad de seleccionar un titular del blog y mostrar sus detalles:

```
function onkeyDown(e):void {
    if (e.keyCode==Keyboard.ENTER) {
        trace(myNews[visi+selectedNews].title);
        if(isSet==false){
            showNews();
            isSet=true;
        }
        return;
    }
    if (e.keyCode==57) { // 9 key
        comeBack();
    }
    if (e.keyCode==55) { // 7 key
        goto();
    }
    if (e.keyCode==53) { // 5 key
        if(isSet==false){onrefresh();}
    }
    if (e.keyCode==Keyboard.DOWN) {
        if(isSet==true){return;}
        if (selectedNews<3) {
            highlightBar.y+=62;
            selectedNews+=1;
            scrollee.scroller.y+=(280/myNews.length);
        }
    }
}
```

```

        return;
    }
    if (((visi+4) >= (myNews.length))) {
        return;
    }
    scrollee.scroller.y+=(280/myNews.length);
    visi+=1;
    showText();
}
if (e.keyCode==Keyboard.UP) {
    if (isSet==true){return;}
    if (selectedNews>0) {
        highlightBar.y-=62;
        selectedNews-=1;
        scrollee.scroller.y-=(280/myNews.length);
        return;
    }
    if (visi==0) {
        return;
    }
    scrollee.scroller.y-=(280/myNews.length);
    visi-=1;
    showText();
}
}
}

```

Adición de la función showNews()

Cuando se presiona la tecla de selección para ver los detalles de una entrada del blog, la aplicación llamada a la clase `readNews`. Esta clase se define en el archivo `readNews.as` en la carpeta de tutoriales. La clase `readNews` proporciona funcionalidad que muestra todo el contenido de una entrada del blog. En el fotograma 1 de la capa de acción, pegue el siguiente código en el panel Acciones:

```

function showNews():void {
    newsshowing=new readNews(stage,myNews[visi+selectedNews].title,
        myNews[visi+selectedNews].description,myNews[visi+selectedNews].date);
    newsshowing.x=3;
    newsshowing.y=32;
    addChild(newsshowing);
}

```

Adición de la función comeback()

Para volver a la lista de entradas del blog, presione 9 en el teclado. En el fotograma 1 de la capa de acción, añada el siguiente código en el panel Acciones:

```

function comeBack():void {
    if (isSet==true) {
        removeChild(newsshowing);
        isSet=false;
    }
    return;
}

```

Actualización de la lista de entradas del blog

Para actualizar la lista de entradas del blog, se presiona 5 en el teclado. La aplicación actualiza la fuente RSS y muestra el código XML actual.

- 1 En Fotograma 1 de la capa de acciones, añada el siguiente código en el panel Acciones:

```
function onrefresh():void {
    visi=0;
    selectedNews=0;
    scrollee.scroller.y=0;
    highlightBar.y=73.95;
    loader.load(new URLRequest("http://blogs.adobe.com/index.xml"));
    stage.addChild(a1);
}
```

- 2 Junto a la parte superior del código del panel Acciones, llame a la función `onrefresh()` inmediatamente después de agregar los detectores de eventos `loader`:

```
var loader:URLLoader = new URLLoader();
loader.addEventListener(Event.COMPLETE, loadxml);
loader.addEventListener(IOErrorEvent.IO_ERROR, ioerror);
loader.addEventListener(SecurityErrorEvent.SECURITY_ERROR, securityerror);
onrefresh();
```

Desplazamiento en la página web de la entrada del blog

Para ver una entrada del blog en una ventana del navegador independiente, se presiona 7 en el teclado. En Fotograma 1 de la capa de acciones, añada el siguiente código en el panel Acciones:

```
function goto():void {
    if (isSet==true) {
        navigateToURL(new URLRequest(myNews[visi+selectedNews].link));
    }
    return;
}
```

Prueba y publicación de la aplicación

La aplicación de tutorial se ha completado. Se puede probar en Device Central, seleccionando Depurar > Depurar película > En Device Central, o bien, Control > Probar película > En Device Central.

Para publicar la aplicación, seleccione Archivo > Publicar. Flash Professional guarda el archivo SWF incorporado como `blogs-rss.html`. La seguridad de Flash Player requiere que registre su archivo SWF local como de confianza antes de que Flash Player pueda mostrarlo. Para obtener información sobre el registro de archivos, consulte www.adobe.com/go/learn_fl_securitypanel_es.

Capítulo 4: Creación de interactividad y navegación

Para interactuar con su aplicación de Adobe Flash Lite, un usuario debe poder llevar a cabo lo siguiente:

- Determinar qué objeto en pantalla está seleccionado actualmente.
- Desplazarse entre los objetos.
- Iniciar una operación seleccionando un objeto.

Interacción del usuario y teclas admitidas

Interacción del usuario en Flash Lite

Flash Lite admite la navegación y la interacción del usuario por medio del teclado del dispositivo o de la interfaz de pantalla táctil o con lápiz stylus. Las opciones disponibles en la aplicación varían en función del dispositivo de destino y del tipo de contenido. Para obtener más información sobre los tipos de contenido, consulte el apartado [“Tipos de contenido de Flash Lite”](#) en la página 7.

La forma más sencilla de incorporar interactividad basada en teclas en una aplicación de Flash Lite consiste en el desplazamiento predeterminado. En el desplazamiento predeterminado se utiliza el teclado de cuatro direcciones del dispositivo como las teclas de flecha o el tabulador y Mayús+tabulador en una aplicación de escritorio. El usuario puede desplazar la selección hasta el objeto deseado y, después, presionar la tecla de selección. La aplicación incluye código de controlador de eventos para responder ante estos eventos de botones.

El desplazamiento predeterminado en Flash Lite funciona con botones, campos de introducción de texto y, opcionalmente, clips de película. El desplazamiento predeterminado resulta más adecuado para las interacciones sencillas del usuario como, por ejemplo, menús.

Las aplicaciones también pueden responder a eventos arbitrarios de pulsación de tecla o de ratón que genera Flash Lite cuando un usuario hace clic en un botón del ratón o presiona una tecla concreta. El desplazamiento basado en eventos permite crear aplicaciones de Flash Lite, como juegos con modelos complejos de interacción con el usuario.

Teclas que pueden utilizarse en Flash Lite

Además de las teclas alfanuméricas disponibles en los teléfonos estándar, la mayoría de los dispositivos móviles cuenta con un teclado de desplazamiento. Este teclado permite al usuario desplazarse y seleccionar elementos en la pantalla del dispositivo.

Un teclado de navegación típico tiene cuatro teclas de navegación (arriba, abajo, izquierda y derecha) y una tecla de selección (que se suele encontrar en el centro del teclado). Las distintas aplicaciones pueden utilizar estas teclas de formas diferentes. Por ejemplo, un juego utiliza las teclas de desplazamiento para que el usuario pueda mover un personaje en pantalla. El juego también puede utilizar la tecla de selección para realizar otra acción como, por ejemplo, hacer que salte el personaje.

No todos los dispositivos y tipos de contenido de Flash Lite admiten estas teclas. Por ejemplo, los dispositivos que admiten navegación en dos direcciones no son compatibles con las teclas de desplazamiento izquierda y derecha.

Flash Lite admite las siguientes teclas en dispositivos móviles:

Descripción	Teclas	Disponibilidad
Numeric, *, #	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, *, #	Todos los dispositivos
Teclado de cinco teclas	Selección, arriba y abajo	Todos los dispositivos
	Izquierda y derecha	Dispositivos que solamente admiten el desplazamiento en cuatro direcciones (consulte "Modos de desplazamiento con teclado" en la página 16)
Teclas del teclado	!, ", #, \$, %, &, ', (,), *, +, ,, -, ., /, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, :, ;, <, +, >, ?, @, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, [\], ^, _ ' a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, { }, ~, Retroceso	Dispositivos con teclado QWERTY estándar

Para obtener más información sobre la clase `System.Capabilities`, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Navegación predeterminada

Utilización del desplazamiento con teclado predeterminado en Flash Lite

En las aplicaciones de escritorio, el tabulador y las teclas Mayús+tabulador permiten a los usuarios cambiar la selección de los distintos objetos de la pantalla. El funcionamiento de las teclas de flecha es similar al del resto de aplicaciones. En Flash Lite, las teclas de desplazamiento del teclado de navegación del dispositivo realizan la misma función que las teclas de flecha o el tabulador y la combinación Mayús+tabulador de una aplicación de escritorio. Una vez seleccionado el objeto deseado, el usuario puede presionar la tecla de selección para activar una acción de la aplicación. Se pueden definir controladores de eventos para responder a la selección de un botón o un clip de película; para obtener más información, consulte ["Administración de eventos de tecla y de ratón"](#) en la página 22.

El desplazamiento predeterminado mediante teclado en Flash Lite se realiza con botones y campos de introducción de texto. Los clips de película también se incluyen si cumplen con los siguientes criterios:

- Su propiedad `tabEnabled` se establece en `true`.
- Cuentan con controladores de eventos asociados a los mismos y su propiedad `tabEnabled` no se establece en `false`.

Cuando se selecciona un campo de introducción de texto y el usuario presiona la tecla de selección, Flash Lite activa el campo de texto para la edición en línea.

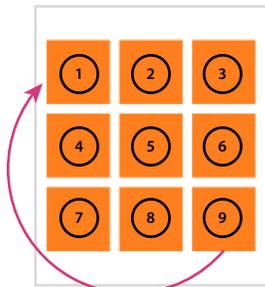
Para ver un ejemplo que ilustra el desplazamiento predeterminado, consulte ["Creación de un menú sencillo utilizando botones con desplazamiento de ratón"](#) en la página 24.

Modos de desplazamiento con teclado

Flash Lite admite tres modos de desplazamiento con teclado: dos direcciones, cuatro direcciones y cuatro direcciones con rotación. Cada dispositivo y tipo de contenido de Flash Lite admite un modo de navegación distinto.

El *desplazamiento en dos direcciones* en Flash Lite es similar al desplazamiento con tabulación de las aplicaciones de escritorio. En las aplicaciones de escritorio, mediante las teclas de tabulación y Mayús+Tabulador se realiza el desplazamiento entre los objetos en pantalla. La tecla de navegación hacia abajo del dispositivo equivale al tabulador y la tecla de navegación hacia arriba equivale a la combinación Mayús+tabulador.

El orden predeterminado de tabulación en la navegación en dos direcciones suele ser de izquierda a derecha y de arriba abajo. Por ejemplo, en la siguiente imagen se muestra una cuadrícula de tres por tres con objetos Button en una aplicación de Flash Lite. Los números que aparecen sobre cada botón indican el orden en el que se seleccionará cada botón con el teclado cuando el usuario pulse varias veces la tecla de desplazamiento hacia abajo del dispositivo. Una vez seleccionado el botón situado en la esquina inferior izquierda con el teclado, la selección “rotará” al botón situado arriba a la izquierda cuando el usuario presione la tecla de desplazamiento hacia abajo.



Ejemplo del orden de tabulación en la navegación en dos direcciones

Se puede personalizar el orden de tabulación en la navegación en dos direcciones mediante la propiedad `tabIndex` de los objetos Button, MovieClip y TextField. Para obtener más información, consulte el apartado “[Control del orden de tabulación en la navegación en dos direcciones](#)” en la página 21.

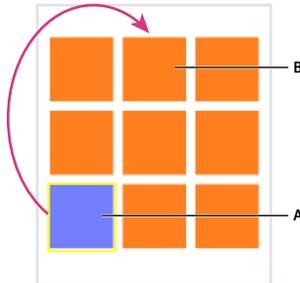
Para ver un ejemplo de la navegación en dos direcciones, consulte la página de tutoriales y ejemplos de Flash Lite en www.adobe.com/go/learn_flt_samples_and_tutorials_es. Localice el archivo ZIP para su versión de ActionScript y descargue y descomprima este archivo. Desplácese a la carpeta de muestras para acceder al archivo de ejemplo denominado 2-way fla.

La navegación en cuatro direcciones en Flash Lite es similar a la utilización de las teclas de flecha en el teclado de un equipo de escritorio para desplazarse por los objetos en la pantalla. Las teclas de navegación hacia arriba, abajo, izquierda y derecha del dispositivo equivalen a las cuatro teclas de flecha del teclado de un equipo. Si se presiona una tecla de navegación, la selección del teclado pasa al objeto que se encuentra en dicha dirección (si existe). Si no hay ningún objeto en dicha dirección, la selección del teclado permanece en el objeto actual.

Nota: la propiedad `tabIndex` no se admite en dispositivos compatibles con el desplazamiento en cuatro direcciones, pero las propiedades `tabEnabled` y `tabChildren` sí se admiten. Este comportamiento es diferente del modo en que funcionan las propiedades en las aplicaciones de escritorio.

Para ver un ejemplo de la navegación que utiliza dos direcciones, consulte la página de tutoriales y ejemplos de Flash Lite en www.adobe.com/go/learn_flt_samples_and_tutorials_es. Localice el archivo ZIP para su versión de ActionScript y descargue y descomprima este archivo. Posteriormente, desplácese a la carpeta de muestras para acceder al archivo de ejemplo denominado 4-way fla.

La navegación en cuatro direcciones con rotación se comporta como una combinación de la navegación de cuatro direcciones estándar y la navegación en dos direcciones. Al igual que ocurre con la navegación en cuatro direcciones descrita anteriormente, el usuario puede cambiar la selección del teclado con las teclas de navegación en cuatro direcciones del dispositivo. La diferencia estriba en que, al igual que ocurre con la navegación en dos direcciones, la selección del teclado “rota” al objeto del lado opuesto de la pantalla. Por ejemplo, en la siguiente imagen, el botón con la selección actual del teclado se encuentra en la esquina inferior izquierda de la pantalla. Si el usuario presiona la tecla de navegación hacia abajo, el siguiente botón seleccionado será el que se encuentra en el centro de la fila superior de botones.



A. Botón con la selección actual B. Siguiente botón seleccionado después de que el usuario haya presionado la tecla de navegación

Se puede probar el comportamiento de los modos de navegación en dos y cuatro direcciones en el emulador de Adobe Device Central utilizando los archivos de muestra llamados 2-way.fla y 4-way.fla, que se encuentran en www.adobe.com/go/learn_flt_samples_and_tutorials_es. En la página de tutoriales y ejemplos, busque, descargue y descomprima el archivo ZIP de su versión de Flash Lite y, a continuación, acceda a la carpeta Samples para poder ejecutar los ejemplos.

Cada archivo de muestra se compone de la misma cuadrícula de botones, tal como se ha descrito anteriormente. Sin embargo, existe una sola diferencia. Todos los archivos FLA se configuran para centrarse en una combinación del dispositivo y del tipo de contenido de Flash Lite que admite el modo de desplazamiento (dos direcciones o cuatro direcciones).

Para utilizar un archivo de muestra, ábralo en Flash y pruébelo con el emulador de Adobe Device Central (seleccione Control > Probar película > En Device Central). Haga clic en las teclas de flecha del teclado del emulador (o presione las teclas de flecha del teclado) para ver cómo cada modo de navegación afecta a la navegación del usuario.



Botones de un archivo de muestra

Directrices para el uso de la navegación predeterminada

A continuación se proporcionan directrices y consideraciones para utilizar la navegación predeterminada en las aplicaciones de Flash Lite.

- Si se deshabilita el rectángulo de selección estableciendo `focusRect` en `false`, asegúrese de proporcionar un indicador de selección alternativo. Este indicador de selección es para botones, campos de introducción de texto y clips de película con tabulación. Para los botones, se puede hacer lo mismo añadiendo un elemento visual al estado correspondiente al desplazamiento del puntero sobre el botón. Este estado es la parte de la línea de tiempo de un objeto de botón que se muestra cuando se selecciona el botón. Para ver un ejemplo de esta técnica, consulte la aplicación de ejemplo descrita en [“Creación de un menú sencillo utilizando botones con desplazamiento de ratón”](#) en la página 24. En el caso de los campos de introducción de texto, se puede utilizar el objeto `Stage` para determinar en qué momento se selecciona el campo de texto y mostrar el indicador de selección deseado.
- Es preciso contar con, al menos, dos objetos (campos de introducción de texto, clips de película con tabulación o una combinación de ambos) en la pantalla al mismo tiempo. Supongamos que la pantalla contiene un solo campo de introducción de texto, botón o clip de película con tabulación. En esta situación, el usuario no puede cambiar la selección y puede sentirse bloqueado en la interfaz.

Si una pantalla de la aplicación contiene solamente un botón para la interacción del usuario, puede ser conveniente detectar un evento de pulsación de tecla en lugar de utilizar eventos de botones.

- Si procede, plantéese la utilización de la propiedad `Stage.focus()` para establecer la selección inicial en un objeto específico de la pantalla. De este modo podrá guiar al usuario en la interfaz y reducir la cantidad de teclas de navegación que debe pulsar. Por ejemplo, suponga que una pantalla de la aplicación contiene un campo de introducción de texto. Para introducir un valor en el campo de texto, en primer lugar el usuario presiona una tecla de desplazamiento para seleccionar el campo. A continuación, el usuario presiona la tecla de selección para abrir el cuadro de diálogo de entrada de texto. Podría utilizar el siguiente código `ActionScript` para establecer automáticamente la selección del teclado en el campo de introducción de texto denominado `inputTxt`:

```
stage.focus = inputTxt;
```

Para ver un ejemplo, consulte la aplicación de muestra descrita en [“Control de la selección con `ActionScript`”](#) en la página 20.

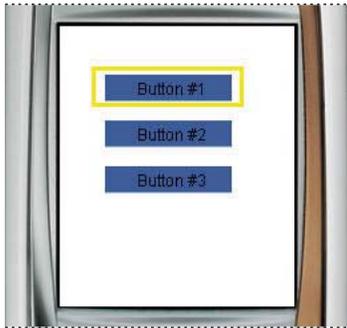
- La alineación de los objetos en la pantalla es importante en la navegación predeterminada. Si los objetos no están alineados en la pantalla, la secuencia en la que se seleccionan tal vez no sea intuitiva para los usuarios. Esto se puede evitar aplicando el mismo tamaño a todos los objetos y alineándolos vertical y horizontalmente tanto como sea posible. Con el desplazamiento en dos direcciones, también es posible controlar la secuencia utilizando la propiedad `tabIndex`.

El rectángulo de selección

De forma predeterminada, Flash Lite dibuja un rectángulo de selección alrededor del botón o campo de introducción de texto que está seleccionado. Los clips de película también se incluyen si cumplen las siguientes condiciones:

- Su propiedad `tabEnabled` se establece en `true`.
- Cuentan con controladores de eventos asociados a los mismos y su propiedad `tabEnabled` no se establece en `false`.

El rectángulo de selección permite al usuario saber qué objeto de la pantalla responderá cuando pulse la tecla de selección del dispositivo. Por ejemplo, la imagen siguiente muestra el rectángulo de selección alrededor de un botón seleccionado con el teclado:



Botón con la selección actual y rectángulo de selección predeterminado

Para los botones y clips de película, el recuadro de delimitación del rectángulo de selección se determina mediante el *área de toque* del objeto. El área de toque es la zona invisible que (en las aplicaciones de escritorio) define la parte del botón o del clip de película que responde a los clics de ratón. En los campos de introducción de texto, las dimensiones del campo de texto determinan el recuadro de delimitación del rectángulo de selección.

Se puede personalizar el color del rectángulo de selección o desactivarlo.

Personalización del rectángulo de selección

El rectángulo de selección es un resaltado amarillo predeterminado que indica qué botón o campo de introducción de texto está seleccionado. Los clips de película se incluyen si su propiedad `tabEnabled` se establece en `true`. Los clips de película también se seleccionan si tienen controladores de eventos asociados y su propiedad `tabEnabled` no se establece en `false`.

Puede desactivar el rectángulo de selección, definiendo la propiedad `Stage.stageFocusRect()` como `false`. El rectángulo de selección también se puede deshabilitar para objetos interactivos específicos como, por ejemplo, clips de película y botones, estableciendo la propiedad `focusRect` del objeto. Para obtener más información,

Control de la selección con ActionScript

El objeto `Stage` se puede utilizar para obtener y definir la selección del teclado, o bien para recibir una notificación cuando un objeto se selecciona con el teclado o cuando se anula su selección.

El uso del objeto `Stage` resulta útil, por ejemplo, si desea seleccionar automáticamente un botón específico cuando la aplicación se carga por primera vez. El objeto `Stage` también se puede usar si desea recibir una notificación cuando un objeto específico en pantalla se haya seleccionado o se haya anulado su selección mediante el teclado. En función de la notificación, puede actualizar la pantalla.

El código siguiente utiliza el método `Stage.Focus()` para seleccionar la instancia de botón denominada `login_btn`:

```
Stage.Focus = login_btn;
```

Los eventos `FocusEvent.FOCUS_IN` y `FocusEvent.FOCUS_OUT` permiten determinar cuándo ha cambiado la selección. Por ejemplo, estos eventos se pueden utilizar para crear un administrador de selección personalizado para todos los objetos interactivos. En el siguiente procedimiento se muestra cómo crear un gestor de selección personalizado que cambie el color del borde del objeto `TextField` cuando éste se seleccione. Para ver una aplicación de muestra finalizada (`custom_focus_manager fla`), consulte la página de tutoriales y ejemplos de Flash Lite en www.adobe.com/go/learn_flt_samples_and_tutorials_es. Localice el archivo ZIP para su versión de ActionScript y descargue y descomprima este archivo. A continuación, desplácese a la carpeta `Samples` para acceder al ejemplo.

Crear un gestor de selección de entrada de datos de texto personalizado

- 1 En Flash, cree un nuevo documento móvil y guárdelo como `custom_focus_manager fla`.
- 2 Con ayuda de la herramienta Texto, cree un campo de texto en el escenario.
- 3 Con el campo de texto aún seleccionado, en la ficha Propiedades, seleccione Texto de entrada en el menú emergente Tipo de texto. Escriba `inputTxt_1` en el cuadro de texto Nombre de instancia y seleccione la opción Mostrar borde alrededor del texto.
- 4 De forma similar, cree otro campo de introducción de texto debajo del primero con el nombre de instancia `inputTxt_2`. Después, seleccione la opción Mostrar borde alrededor del texto para el segundo campo de texto.
- 5 En la línea de tiempo, seleccione el fotograma 1 en la capa denominada `ActionScript`.
- 6 Abra el panel Acciones (Ventana > Acciones) e introduzca (o copie y pegue) el siguiente código:

```
//Set the display to full screen.
stage.displayState = StageDisplayState.FULL_SCREEN
//Add focus in and out event listeners for the current context.
stage.addEventListener(FocusEvent.FOCUS_IN, focushandler);
stage.addEventListener(FocusEvent.FOCUS_OUT, focusouthandler);
function focushandler(e:FocusEvent): void {
    if (e.target is TextField){
        //Set border color of text field with focus to red.
        e.target.borderColor = 0xFF0000;
    }
}
function focusouthandler(e:FocusEvent): void {
    if (e.target is TextField){
        //Set border color of text field without focus to black.
        e.target.borderColor = 0x000000;
    }
}
```

- 7 Guarde las modificaciones y pruebe la aplicación en el emulador (Control > Probar película > En Device Central).
- 8 Presione las teclas arriba y abajo del emulador para cambiar la selección del teclado entre los dos campos de texto. El campo de texto seleccionado debería tener un borde rojo y el que no está seleccionado debería tener un borde negro. Presione la tecla de selección cuando el campo de texto esté seleccionado para hacer que aparezca el cuadro de diálogo de introducción de texto.

Control del orden de tabulación en la navegación en dos direcciones

La navegación en dos direcciones de Flash Lite es similar a la navegación con tabulación de Flash y, por lo tanto, admite la propiedad `InteractiveObject.tabIndex`, que permite al usuario definir el orden de tabulación específico de los botones, clips de película y campos de introducción de texto. En los dispositivos compatibles con el desplazamiento en cuatro direcciones, no se admite la propiedad `tabIndex`. Por lo tanto, no es posible establecer el orden de tabulación utilizando la propiedad `tabIndex` para el desplazamiento en cuatro direcciones.

Para controlar el orden de tabulación en la navegación en dos direcciones, se puede asignar la propiedad `tabIndex` de cada objeto interactivo a un número que especifique el orden del objeto en la navegación predeterminada. Por ejemplo, suponga que una aplicación contiene un botón (`my_button`), un clip de película (`my_movieclip`) y un campo de introducción de texto (`my_inputTxt`). El siguiente código establece el orden de tabulación de tal forma que el botón se selecciona primero, después se selecciona el clip de película y, por último, se selecciona el campo de introducción de texto.

```
my_button.tabIndex = 1;
my_movieclip.tabEnabled = true;
my_movieclip.tabIndex = 2;
my_inputTxt.tabIndex = 3;
```

Nota: Los objetos más interactivos deben tener `tabEnabled` establecido en `true` menos para unas cuantas excepciones, incluyendo botones y campos de introducción de texto. Consulte la descripción de la propiedad `flash.display.InteractiveObject.tabEnabled()` para obtener más información.

Administración de eventos de tecla y de ratón

En ActionScript 3.0, la arquitectura del evento se ha reescrito desde la base.

Se pueden utilizar objetos interactivos (cualquier objeto derivado de `flash.display.InteractiveObject`) para añadir interactividad a las aplicaciones de Flash Lite.

Acerca de los eventos de ratón

Los controladores y los detectores de eventos especifican el modo en que responde la aplicación a las instancias generadas por el usuario y el sistema. Además de utilizar la navegación predeterminada y responder a eventos relacionados, una aplicación de Flash Lite puede detectar y responder a eventos de ratón.

Gestión de eventos de ratón

Flash Lite admite los siguientes eventos de ratón de ActionScript:

Evento de ratón	Descripción
CLICK	Define el valor de la propiedad <code>type</code> de un objeto de evento <code>click</code> .
DOUBLE_CLICK	Define el valor de la propiedad <code>type</code> de un objeto de evento <code>doubleClick</code> .
MOUSE_DOWN	Define el valor de la propiedad <code>type</code> de un objeto de evento <code>mouseDown</code> .
MOUSE_MOVE	Define el valor de la propiedad <code>type</code> de un objeto de evento <code>mouseMove</code> .
MOUSE_OUT	Define el valor de la propiedad <code>type</code> de un objeto de evento <code>mouseOut</code> .
MOUSE_OVER	Define el valor de la propiedad <code>type</code> de un objeto de evento <code>mouseOver</code> .
MOUSE_UP	Define el valor de la propiedad <code>type</code> de un objeto de evento <code>mouseUp</code> .
MOUSE_WHEEL	Define el valor de la propiedad <code>type</code> de un objeto de evento <code>mouseWheel</code> .
ROLL_OUT	Define el valor de la propiedad <code>type</code> de un objeto de evento <code>rollOut</code> .
ROLL_OVER	Define el valor de la propiedad <code>type</code> de un objeto de evento <code>rollOver</code> .

El siguiente procedimiento muestra cómo crear una aplicación sencilla que gestione eventos de ratón. Para obtener un ejemplo de cómo crear un menú sencillo utilizando botones con desplazamiento del ratón, consulte [“Creación de un menú sencillo utilizando botones con desplazamiento de ratón”](#) en la página 24.

Creación de un controlador de eventos de ratón para botones

1 En Flash, cree un nuevo documento móvil y guárdelo como `custom_mouse_handler fla`.

- 2 Seleccione Ventana > Bibliotecas comunes > Botones para abrir una biblioteca externa de símbolos de botón creados previamente.
- 3 En el panel Biblioteca, haga doble clic en la carpeta de botones clásicos para abrirla y, a continuación, abra la carpeta Circle Buttons.
- 4 Arrastre una instancia del símbolo de botón Menu al escenario.
- 5 En la ficha Propiedades, en el campo de texto Nombre de instancia, escriba **btn_1**.
- 6 Arrastre otra instancia del mismo botón al escenario y sitúelo justo debajo del primero.
- 7 En la ficha Propiedades, en el campo de texto Nombre de instancia, escriba **btn_2**.
- 8 En la línea de tiempo, seleccione el fotograma 1 en la capa denominada ActionScript.
- 9 Abra el panel Acciones (Ventana > Acciones) e introduzca el siguiente código:

```
// Disable the focus rectangle because buttons have an over state.
stage.stageFocusRect = false;

//Add event listeners for btn_1.
btn_1.addEventListener(MouseEvent.CLICK, downhandler1);
btn_1.addEventListener(MouseEvent.CLICK, uphandler1);
btn_1.addEventListener(MouseEvent.CLICK, overhandler1);
btn_1.addEventListener(MouseEvent.CLICK, outhandler1);

//Add event handlers for btn_1.
function downhandler1(e:MouseEvent):void {
    trace("you pressed Button 1");
}

function uphandler1(e:MouseEvent):void {
    trace("you released Button 1");
}

function overhandler1(e:MouseEvent):void {
    trace("Button 1 has focus");
}

function outhandler1(e:MouseEvent):void {
    trace("Button 1 lost focus");
}

//Add event listeners for btn_2.
btn_2.addEventListener(MouseEvent.CLICK, downhandler2);
```

```
btn_2.addEventListener(MouseEvent.CLICK, clickHandler2);
btn_2.addEventListener(MouseEvent.MOUSE_UP, uphandler2);
btn_2.addEventListener(MouseEvent.MOUSE_OVER, overhandler2);
btn_2.addEventListener(MouseEvent.MOUSE_OUT, outhandler2);

//Add event handlers for btn_2.
function downhandler2(e:MouseEvent):void {
    trace("you pressed Button 2");
}

function uphandler2(e:MouseEvent):void {
    trace("you released Button 2");
}

function overhandler2(e:MouseEvent):void {
    trace("Button 2 has focus");
}

function outhandler2(e:MouseEvent):void {
    trace("Button 2 lost focus");
}
```

10 Pruebe la aplicación en el emulador (Control > Probar película > En Device Central).

Observe los mensajes en el panel Salida conforme hace clic en los botones en el escenario.

Otros tipos de objetos admiten diferentes eventos. Por ejemplo, el objeto TextField incluye un evento `change` que se ejecuta cuando cambia el contenido de un campo de texto. Puede escribir el código del controlador de eventos para estos eventos utilizando el mismo formato que para los controladores de eventos de botón de este procedimiento. Para obtener más información, consulte las clases TextField y MovieClip en *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Creación de un menú sencillo utilizando botones con desplazamiento de ratón

Para crear un menú sencillo utilizando botones con desplazamiento de ratón, designe el menú y después escriba el código de gestión de eventos para gestionar los eventos de ratón. Para crear el menú, se utilizan tres símbolos de botón, uno para cada opción de menú. Posteriormente se escribe el código de control de eventos que muestra un mensaje cuando el usuario se desplaza sobre cada opción de menú. Es decir, cuando el usuario selecciona el botón correspondiente y hace clic en la opción de menú concreta. Para obtener más información sobre la gestión de eventos de ratón en Flash Lite, consulte el apartado “[Gestión de eventos de ratón](#)” en la página 22.

Comience con un documento de Flash parcialmente finalizado. Puede cambiar esta configuración para utilizar otro dispositivo y otro tipo de contenido (consulte *Uso de Adobe Device Central CS5*).

- 1 Descargue y abra el archivo llamado `simple_menu_start fla`, que se encuentra en www.adobe.com/go/learn_flt_samples_and_tutorials_es. En la página de tutoriales y ejemplos, busque, descargue y descomprima el archivo .zip de su versión de Flash Lite y, a continuación, acceda a la carpeta Samples para poder ejecutar el ejemplo.
- 2 Abra el panel Biblioteca (Ventana > Biblioteca).
Observe que la Biblioteca contiene tres símbolos de botón denominados News, Weather y Sports.
- 3 En la línea de tiempo (Ventana > Línea de tiempo), seleccione la capa Menu Buttons.
- 4 Arrastre una instancia del símbolo de botón News desde el panel Biblioteca hasta el escenario.
- 5 En la ficha Propiedades, en el campo de texto Nombre de instancia, escriba `btn_news`.

- 6 Repita los pasos 4 y 5 para los botones Sports y Weather, y denomínelos **btn_sports** y **btn_weather**, respectivamente.
- 7 Alinee los tres botones verticalmente, tal como se muestra en el ejemplo siguiente:
- 8 En el panel Herramientas, seleccione la herramienta Texto y cree un campo de texto en la parte inferior del escenario.

Este campo de texto muestra un breve mensaje cuando el usuario pasa el ratón por encima de cada opción de menú.

- 9 Con el nuevo campo de texto seleccionado, haga lo siguiente en la ficha Propiedades:
 - a Escriba **txt_status** en el campo de texto Nombre de instancia.
 - b Seleccione Texto dinámico en el menú emergente Tipo de texto.
 - c Seleccione Utilizar fuentes del dispositivo en el menú emergente Método de representación de fuentes.
- 10 En la línea de tiempo, seleccione el fotograma 1 en la capa denominada ActionScript.
- 11 Abra el panel Acciones (Ventana > Acciones) e introduzca el siguiente código:

```
//Disable the focus rectangle because buttons have an over state.
stage.stageFocusRect = false;

//Add event listeners for each button to listen for roll overs and clicks.
btn_news.addEventListener(MouseEvent.ROLL_OVER, mouserollhandler);
btn_news.addEventListener(MouseEvent.CLICK, mouseclickhandler);

btn_sports.addEventListener(MouseEvent.ROLL_OVER, mouserollhandler);
btn_sports.addEventListener(MouseEvent.CLICK, mouseclickhandler);

btn_weather.addEventListener(MouseEvent.ROLL_OVER, mouserollhandler);
btn_weather.addEventListener(MouseEvent.CLICK, mouseclickhandler);

//Set the function to respond to roll overs.
function mouserollhandler (myEvent:MouseEvent): void {
    //Check if the current button is the btn_news button.
    if (myEvent.target == btn_news){
        //Set the text field text.
        txt_status.text = "Press to select News"
    }
    //Check if the current button is the btn_sports button.
    else if (myEvent.target == btn_sports){
        txt_status.text = "Press to select Sports"
    }
    //Check if the current button is the btn_weather button.
```

```

        else if (myEvent.target == btn_weather){
            txt_status.text = "Press to select Weather"
        }
    }

    //Set the function to respond to clicks.
    function mouseclickhandler (myEvent:MouseEvent): void {
        //Check if the current button is the btn_news button.
        if (myEvent.target == btn_news){
            txt_status.text = "You pressed News"
        }
        //Check if the current button is the btn_sports button.
        else if (myEvent.target == btn_sports){
            txt_status.text = "You pressed Sports"
        }
        //Check if the current button is the btn_weather button.
        else if (myEvent.target == btn_weather){
            txt_status.text = "You pressed Weather"
        }
    }
}

```

12 Seleccione Control > Probar película > En Device Central para mostrar una vista previa de la aplicación en el emulador.

Haga clic en todas las opciones de menú. Al hacer clic en cada elemento, tenga en cuenta el mensaje que aparece.

Gestión de eventos de teclado

Flash Lite genera eventos de teclado como respuesta a la pulsación de teclas en el dispositivo. Puede escribir el código de un gestor de teclas para responder a estos eventos. Para obtener una lista de las teclas admitidas en Flash Lite, consulte el apartado [“Teclas que pueden utilizarse en Flash Lite”](#) en la página 15.

En la siguiente tabla se incluyen los eventos de teclado de ActionScript 3.0 que admite Flash Lite:

Evento de teclado	Descripción
KEY_DOWN	Define el valor de la propiedad type de un objeto de evento keyDown.
KEY_UP	Define el valor de la propiedad type de un objeto de evento keyUp.

Creación de un detector de eventos

Una forma sencilla de gestionar eventos de pulsación de tecla consiste en añadir un detector de eventos para el evento de teclado `keyDown` que haga referencia a una función de controlador. El siguiente código de ejemplo asocia una función de controlador de eventos para que responda cuando el usuario presione la tecla de desplazamiento adecuada en el dispositivo:

```
//Add an event listener to the stage for the keyDown event using the equivalent
//constant value KEY_DOWN from the KeyboardEvent class. This associates the event
//with a "handler" function to respond to the event.
stage.addEventListener(KeyboardEvent.KEY_DOWN, onKeyDownHandler);

//Create a handler function to respond to the event.
function onKeyDownHandler(event:KeyboardEvent):void {
    //Because the right navigation key has the key code 39,
    //check for the keycode value 39, then respond.
    if(event.keyCode == 39){
        trace("You pressed the Right Arrow key");
    }
}
```

Gestión de eventos de teclado con un detector de teclas

- 1 En Flash, cree un nuevo documento móvil y guárdelo como keylistener fla.
- 2 Seleccione Capa 1 en la línea de tiempo y cámbiele el nombre a Contenido.
- 3 En el panel Herramientas, seleccione la herramienta Óvalo y cree un óvalo o un círculo en el escenario. Después conviértalo en un clip de película.
- 4 Con el nuevo clip de película seleccionado, en la ficha Propiedades, escriba **circle** en el campo de texto de nombre de instancia.
- 5 En la línea de tiempo, inserte una capa y cámbiele el nombre como ActionScript. En esta capa, seleccione el primer fotograma.
- 6 Abra el panel Acciones (Ventana > Acciones) e introduzca el siguiente código:

```
//Add an event listener to the circle movie clip for the keyDown event using
//the equivalent constant value KEY_DOWN from the KeyboardEvent class.
//This associates the event with a "handler" function to respond to the event.
circle.addEventListener(KeyboardEvent.KEY_DOWN, onKeyDownHandler);
circle.focusRect = false;
stage.focus = circle;

//Create a handler function to respond to the event.
function onKeyDownHandler(event:KeyboardEvent):void {
    //Because the right navigation key has the key code 39,
    //check for the keycode value 39, then respond.
    if(event.keyCode == 39){
        circle.x++;
    }
    //Now respond to the left key.
    else if(event.keyCode == 37){
        circle.x--;
    }
    //Now respond to the up key.
    else if(event.keyCode == 38){
        circle.y--;
    }
    //Now respond to the down key.
    else if(event.keyCode == 40){
        circle.y++;
    }
}
```

7 Pruebe la aplicación seleccionando Control > Probar película > En Device Central.

Presione las cuatro teclas de navegación en el teclado del emulador (o las teclas de flecha correspondientes del teclado) para que el círculo se mueva por el escenario.

Compatibilidad con la tecnología multitáctil

Flash Lite 4 introduce la compatibilidad con la tecnología multitáctil para administrar eventos multitáctiles y eventos de gestos nativos. Esto significa que todos los dispositivos móviles compatibles se acomodan y responden a varios puntos de toque. Asimismo, los datos entrantes del punto de toque se procesan en gestos de mayor nivel. Para activar esta función, se han creado nuevas API de eventos de gesto y toque para ActionScript 3.0.

Con esta función, es posible interactuar con los dispositivos tocando la pantalla con uno o varios dedos. El sistema operativo subyacente realiza un seguimiento de cada punto de toque.

El contenido de Flash Lite 4 se ha creado específicamente para los dispositivos habilitados para varios toques. Las API de evento de toque están disponibles para exponer estos eventos en el código ActionScript.

La compatibilidad con la tecnología multitáctil se puede categorizar en dos amplias categorías. En la primera categoría, el contenido de Flash Lite responde directamente a los eventos cuando los controladores del dispositivo y el digitalizador los generan. Por ejemplo, es posible arrastrar los dedos por la pantalla en una aplicación de dibujo para dibujar líneas paralelas. En la segunda categoría, el sistema operativo sintetiza un flujo entrante de puntos de toque en gestos de mayor nivel. Las API exponen los gestos nativos del sistema operativo en el contenido de Flash Lite mediante las API de ActionScript 3.0. Por ejemplo, se puede reducir el tamaño de una imagen arrastrando los dedos por la pantalla con un movimiento de pellizco.

Gestión de eventos de toque

Los objetos de ActionScript 3.0 responden a entradas de toque detectando un conjunto de tipos de evento de toque. Estos objetos reflejan los eventos de ratón existentes, detectando eventos de desplazamiento, salida y superposición además de los eventos de estado básicos como, por ejemplo, abajo, movimiento o arriba. Los objetos de ActionScript 3.0 no admiten los estados de activación ni un sensor de proximidad.

Los eventos de toque y de ratón presentan una diferencia principal. Cada evento de toque contiene un ID de punto de toque exclusivo que identifica un conjunto de eventos relacionados. Esto significa que todos los eventos generados por un solo dedo se asignan a un ID exclusivo de cualquier otro dedo. Los eventos de toque abajo y toque arriba mantienen este ID entre ellos.

Para los sistemas que sólo son táctiles, el sistema operativo asigna un solo punto de toque como principal. También asigna un flujo paralelo de eventos de ratón de nivel de SO para asignar el punto de toque principal.

Gestión de eventos de gesto nativo

El hardware digitalizador genera gestos nativos en el nivel de SO y los reenvía al contenido de Flash Lite en forma de una nueva clase de evento. El digitalizador asigna una fase de gesto a cada evento de gesto. Estas fases son *begin*, *update* y *end*. Cuando el SO detecta un nuevo gesto, lo prueba con una pila de objetos en la ubicación de gesto. Si el SO identifica un detector, envía el evento de gesto inicial y todos los eventos posteriores asociados con el gesto. El SO omite las ubicaciones de los eventos posteriores aunque sean distintos al evento de gesto inicial.

API de ActionScript 3.0

Para esta función, ActionScript 3.0 introduce la clase Multitouch. Esta clase auxiliar administra el estado global del sistema multitáctil.

Para obtener más información sobre la clase Multitouch y sus subclases, propiedades y métodos, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Capítulo 5: Utilización de texto y fuentes

Se pueden añadir campos de texto dinámicos y estáticos o campos de introducción de texto a Adobe Flash Lite 4.

Texto

Texto en Flash Lite

Flash Lite 4 admite las siguientes funciones de texto:

- Campos estáticos, dinámicos y de introducción de texto.

En tiempo de ejecución, no se puede cambiar el contenido de un campo de texto estático, pero sí el de un campo dinámico o de introducción de texto. Los campos de introducción de texto permiten a los usuarios introducir texto. Flash Lite 4 admite la introducción de texto en línea en la mayoría de los dispositivos. En los dispositivos que admiten idiomas complejos, los campos de introducción de texto utilizan el mecanismo de introducción de texto genérico del dispositivo.

- Fuentes incorporadas y de dispositivo.

Flash Lite 4 puede representar campos de texto con contornos de fuentes incorporados en el archivo SWF o utilizando fuentes disponibles en el dispositivo. Para obtener más información, consulte [“Métodos de representación de fuentes en Flash Lite”](#) en la página 39.

- Codificación de texto Unicode.

Flash Lite 4 puede mostrar texto de cualquier idioma siempre y cuando las fuentes que contengan los glifos de los caracteres necesarios estén disponibles. Para obtener información sobre la edición en varios idiomas en Flash, consulte "Creación de texto en varios idiomas" en *Uso de Flash*.

- Compatibilidad parcial con formato HTML y propiedades de la clase TextFormat.
- Texto desplazable
- Introducción de texto en línea

La introducción de texto en línea permite que el usuario introduzca texto directamente en los campos de texto.

- Compatibilidad con el motor de texto de Flash Player 10.

Flash Lite admite `flash.text.engine` en esta versión. `flash.text.engine` ofrece una compatibilidad de bajo nivel para realizar un sofisticado control de las medidas y el formato del texto, además de admitir texto bidireccional. Para obtener más información sobre `flash.text.engine`, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

- Compatibilidad con texto previsible.

Esta función proporciona funcionalidad como la finalización de palabras y las listas de propuestas. Flash Lite 4 admite los mejores motores de texto previsible (por ejemplo, T9, eZiTap/eZiText y iTap) en cualquier plataforma. Esta compatibilidad se aplica siempre que las implementaciones del motor de texto sean similares a las API estándar que ofrecen los proveedores de motor de texto previsible.

- Compatibilidad con XMLSocket.

La compatibilidad con socket XML amplía la compatibilidad de escritorio de Flash en Flash Lite 4. Permite crear conexiones de datos continuas de baja latencia para aplicaciones tales como juegos y chat.

Flash Lite 4 no admite todas las funciones de texto que se ofrecen en la versión de escritorio de Flash Player. Flash Lite presenta las siguientes limitaciones:

- No se admite el suavizado avanzado, la tecnología de representación avanzada de fuentes disponible únicamente en Adobe Flash Player 8 y versiones posteriores.
- Se admite el formato de texto, pero sólo están disponibles las opciones color, tipo, tamaño, negrita y cursiva. Además, no se muestra el formato si la fuente del texto del dispositivo no contiene la opción seleccionada. Por ejemplo, un campo con formato de cursiva se muestra como texto sin formato si la fuente del dispositivo no contiene la versión en cursiva.
- El dispositivo no se puede enmascarar, utilizar como máscara o representar con transparencia.
- La opción de formato Generar texto como HTML es parcialmente compatible en los campos dinámicos y de introducción de texto. El texto se muestra sin etiquetas HTML visibles. No obstante, el formato sólo se permite para las siguientes etiquetas: `p`, `br`, `sbr`, `font` (con los atributos `face`, `color` y `size`), `b` y `i`.
- Flash Lite 4 no admite hojas de estilo en cascada (CSS).
- No se admiten componentes Flash, incluidos Label, TextArea y TextInput.
- Los objetos TextField y TextFormat son parcialmente compatibles y existen limitaciones adicionales para árabe, hebreo y tailandés. Para obtener más información, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Creación de texto y aplicación de formato al texto

Puede crear texto y aplicarle formato al texto en Flash Lite 4 del mismo modo que lo haría en una aplicación Flash de escritorio.

Para más información sobre la utilización del texto en Flash, consulte los temas siguientes en *Uso de Flash*:

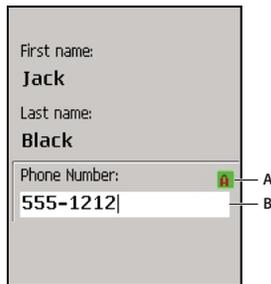
- “Creación de texto”
- “Establecimiento de atributos de texto”
- “Edición de texto”
- “Control de texto con ActionScript”

Para ver una lista de funciones de texto que Flash Lite no admite, consulte “[Texto en Flash Lite](#)” en la página 30.

Introducción de texto

Utilización de campos de introducción de texto

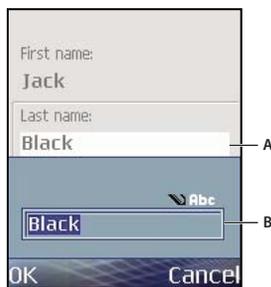
Flash Lite 4 admite la introducción de texto en línea. Esta función permite a los usuarios editar campos de texto directamente en la aplicación de Flash Lite, en lugar de en un cuadro de introducción de texto independiente, como en versiones anteriores. Por ejemplo, en la siguiente figura se muestra cómo aparece un campo de introducción de texto en línea en un dispositivo Symbian™ Series 60 que ejecuta Flash Lite 4:



A. Indicador de modo de entrada B. Campo de texto editándose en línea

Flash Lite 4 admite la introducción de texto en línea para los idiomas latinos y asiáticos. No obstante, esta función no admite idiomas complejos, entre los que se incluyen idiomas de escritura de derecha a izquierda como, por ejemplo, árabe, hebreo, urdu y yídish, así como algunos idiomas asiáticos.

Si el dispositivo no admite la introducción de texto en línea, aún se puede editar el contenido de los campos de introducción de texto. Para ello se puede utilizar un cuadro de texto modal que aparezca sobre el contenido de Flash Lite. Por ejemplo, la siguiente figura muestra un cuadro de diálogo de introducción de texto en un dispositivo Symbian Series 60 en el que se ejecuta Flash Lite 4:



A. Cuadro de diálogo de introducción de texto B. Campo de texto editándose

En general, el contenido creado para versiones anteriores de Flash Lite puede funcionar sin cualquier modificación en Flash Lite 4. En lugar de utilizar el cuadro de diálogo de modos, los usuarios se limitan a editar los campos de introducción de texto en línea. Sin embargo, compruebe que el contenido heredado se vuelva a crear para que utilice las funciones disponibles sólo en Flash Lite 4. Un ejemplo es la capacidad para ejecutar ActionScript y las animaciones de fondo durante la introducción de texto.

Para obtener más información acerca del cuadro de diálogo de introducción de texto, consulte [“Acerca del cuadro de diálogo de introducción de texto del dispositivo \(Flash Lite 4\)”](#) en la página 36.

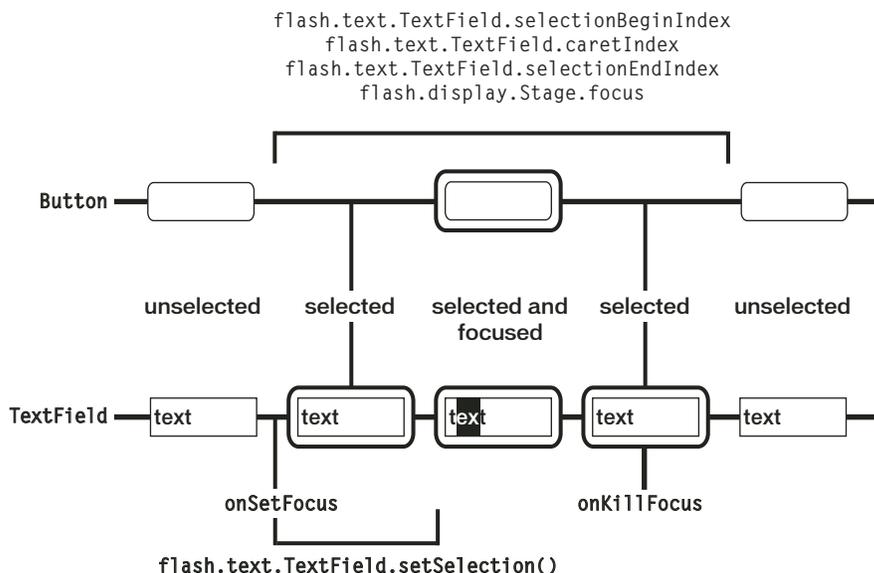
Utilización de la introducción de texto en línea

Flash Lite 4 admite la introducción de texto en línea, lo cual permite a los usuarios introducir y editar texto directamente en un campo de introducción de texto. Al escribir el texto en una aplicación de Flash Lite, los usuarios interactúan con el editor de métodos de entrada (IME) original del dispositivo. El reproductor de Flash Lite no procesa los datos introducidos por el usuario.

Nota: el emulador de Flash Lite de Adobe Device Central no muestra texto en línea.

Para que un campo de texto pueda admitir texto introducido por el usuario, debe estar *activado*. Se puede activar un campo de introducción de texto seleccionándolo. Posteriormente, se puede presionar la tecla de selección del dispositivo o una tecla programada para la que la aplicación establece la entrada de texto en el estado seleccionado. Para obtener más información, consulte “[Activación de campos de introducción de texto con ActionScript 3.0](#)” en la página 34.

En la siguiente figura se muestran los distintos estados de un campo de introducción de texto: sin seleccionar, seleccionado (pero no activado) y activado:



Contenido publicado para Flash Lite 2.1 y 3.x

Si el contenido de Flash Lite 2.1 y 3.x se ejecuta en el reproductor independiente de Flash Lite 4, se utiliza el modo restringido para la introducción de texto en línea. En este modo, no se procesa ActionScript y se omiten todas las animaciones, sonidos y eventos de detención de reproducción de vídeo. Tras desactivar un campo de texto, Flash Lite 4 reanuda el modo normal de funcionamiento.

Contenido publicado para Flash Lite 4

Si el contenido de Flash Lite 4 se ejecuta en el reproductor independiente de Flash Lite 4, se desactivará el modo restringido. Asimismo, ActionScript se procesa durante la introducción de texto en línea. Del mismo modo, si el reproductor es un plugin del navegador, el modo restringido no se aplica.

Impacto de la desactivación del modo restringido

La siguiente tabla describe el modo en que la desactivación del modo restringido influye en los diferentes componentes de Flash Lite.

Componentes de Flash Lite	Impacto
ActionScript	El procesamiento de ActionScript continúa cuando el campo de texto está activo y se produce la introducción de texto.
Eventos	Todas las callbacks de notificaciones de eventos registradas continuarán llamándose cuando se seleccione el campo de texto. Estas notificaciones no esperan que se anule la selección del campo de texto. Las notificaciones incluyen, pero sin limitación, los eventos de ratón y teclado.
Animaciones	Las interpolaciones de forma y movimiento continúan independientemente del estado del campo de introducción de texto.
Vídeo	La reproducción de vídeo continúa cuando el campo de texto está activo de modo que se puede introducir texto mientras se ve el vídeo. En este contexto, el vídeo hace referencia al vídeo remoto, incorporado y externo.
Audio	La reproducción de sonido continúa a pesar de la introducción de texto en línea.
Red	Para cualquier medio transmitido, la reproducción no se detiene si se selecciona un campo de texto. Asimismo, el contenido puede cargar datos en la red.

Evento onChange para campos de texto

En las versiones anteriores de Flash Lite, el evento `onChange` se envía una vez durante toda la sesión de introducción de texto y después de que se anule la selección del campo de texto. En Flash Lite 4, los eventos `onChange` se envían para cada cadena de conversión que se realice y cada evento de pulsación de tecla. Una cadena de conversión se completa si el usuario confirma la cadena. Considere el ejemplo siguiente.

Supongamos que está escribiendo en un dispositivo Symbian Series 60 que ejecuta Flash Lite 4 en modo de impresión en texto. Cuando se escriben unas cuantas letras de una cadena concreta, el dispositivo muestra las palabras posibles para completar la cadena. Asimismo, la cadena se subraya. Cuando se selecciona una cadena de las opciones y se presiona Intro, la cadena se confirma. Se debe tener en cuenta que la cadena confirmada no se subraya.

Campos de introducción de texto en línea y navegación

Un campo de texto activado contiene un puntero que señala el punto de inserción actual. El usuario puede cambiar la ubicación del puntero en el campo de texto mediante las teclas de navegación del dispositivo.

De forma predeterminada, Flash Lite dibuja un rectángulo de selección alrededor del campo de texto que está resaltado en ese momento. En ocasiones, este rectángulo de selección puede llegar a oscurecer el puntero del campo de texto activado o el indicador de inserción. Por ello, recomendamos desactivar el rectángulo de selección (estableciendo `_focusRect` en `false`) y utilizar un indicador de selección personalizado. Para ver un ejemplo de una aplicación que utiliza un indicador de selección personalizado, consulte la página de tutoriales y ejemplos de Flash Lite en www.adobe.com/go/learn_flt_samples_and_tutorials_es. Localice el archivo ZIP para su versión de ActionScript y descargue y descomprima este archivo. A continuación, desplácese a la carpeta Samples para acceder al ejemplo.

Activación de campos de introducción de texto con ActionScript 3.0

El usuario puede activar un campo de introducción de texto con selección del teclado pulsando la tecla de selección del dispositivo. Una aplicación de Flash Lite también puede activar automáticamente un campo de introducción de texto cuando se selecciona con el método `setSelection()`. Este método activa el campo de texto seleccionado en ese momento; si no hay ningún campo seleccionado al ejecutar el comando, no ocurre nada.

Para seleccionar un campo de texto y activarlo posteriormente, se puede utilizar la propiedad `focus` de la clase `Stage` junto con el método `setSelection()` de la clase `TextField`. El siguiente fragmento de código describe cómo utilizar esta propiedad y método.

```
//TextField will have focus after this call
stage.focus = txt;
//TextField will be activated after this call
txt.setSelection(0,0);
```

Para obtener una aplicación de muestra finalizada que utiliza esta técnica, consulte la muestra de introducción de texto en línea que encontrará en www.adobe.com/go/learn_flt_samples_and_tutorials_es. Localice el archivo ZIP para su versión de ActionScript y descargue y descomprima este archivo. A continuación, desplácese a la carpeta `Samples` para acceder al ejemplo.

Indicador de modo de introducción

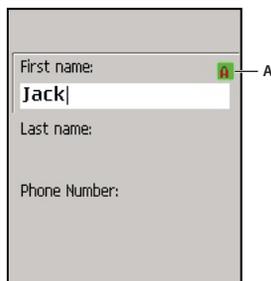
Para ayudar al usuario con las tareas de introducción de texto comunes, la mayoría de los dispositivos proporcionan varios modos de entrada. Estos modos pueden ser entradas de texto previsibles o manuales (*toque triple* o *varios toques*) o todos los modos de sólo números.

Si Flash Lite se ejecuta a pantalla completa, el dispositivo muestra una letra `A` para el modo de introducción alfabética y un símbolo de almohadilla (`#`) para el modo de introducción numérica. En otros modos de pantalla, el dispositivo dibuja un indicador de modo de entrada en una barra de estado o alguna otra ubicación en pantalla. Por ejemplo, la imagen siguiente muestra el indicador de modo de introducción en el modo de pantalla parcial en la versión independiente de Flash Lite 4 para dispositivos Series 60:



A. Indicador de modo de introducción, modo de pantalla parcial B. Campo de texto que se edita

Si el reproductor se ejecuta en modo de pantalla completa, el dispositivo puede dibujar un indicador de modo de introducción en el lugar de la pantalla que desee. Por ejemplo, la figura siguiente muestra el indicador de modo de introducción en el modo de pantalla completa en la versión independiente de Flash Lite 4 para dispositivos Series 60:



A. Indicador de modo de introducción, modo de pantalla completa.

El indicador de modo de introducción para aplicaciones de pantalla completa mostrado en la figura anterior es una implementación de referencia para el reproductor independiente de dispositivos Series 60. El dispositivo determina el indicador específico, si existe, que aparece en pantalla.

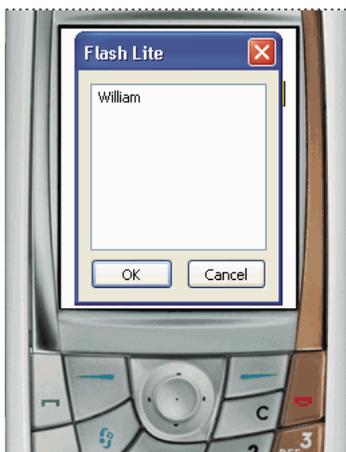
Acerca del cuadro de diálogo de introducción de texto del dispositivo (Flash Lite 4)

Para abrir el cuadro de diálogo de introducción de texto del dispositivo, el usuario debe primero resaltar un campo de introducción de texto y presionar la tecla de selección del dispositivo.

Este cuadro de diálogo es modal (es decir, el usuario no puede interactuar con el contenido mientras el cuadro de diálogo esté resaltado). Lite también inicia una pausa en la cabeza lectora de la aplicación mientras el cuadro de diálogo esté resaltado.

Si selecciona Aceptar (la tecla programable izquierda), el cuadro de diálogo de introducción de texto se cierra y Flash Lite asigna automáticamente el texto al campo de introducción de texto. Al seleccionar Cancelar (la tecla programable derecha), no se asignará ningún texto al campo de introducción de texto.

El emulador de Adobe Device Central imita las funciones del cuadro de diálogo de introducción de texto cuando se prueba la aplicación en la herramienta de creación Flash. La imagen siguiente muestra el cuadro de diálogo de introducción de texto en el emulador:



Cuadro de diálogo de introducción de texto en el emulador

Para ver un ejemplo de cómo utilizar un campo de introducción de texto en una aplicación, consulte [“Creación de una aplicación de ejemplo de campos de texto”](#) en la página 37.

Especificación de tipos de campos de introducción de texto

Flash Lite admite campos de introducción de texto de una línea, de varias líneas y de contraseña. Un tipo de campo de introducción de texto se especifica utilizando el menú desplegable Tipo de línea de la ficha Propiedades.

El tipo de línea especificado para un campo de introducción de texto determina el comportamiento del cuadro de diálogo de introducción de texto del dispositivo cuando el usuario edita el campo de texto.

Por ejemplo, cuando un usuario edita un campo de introducción de texto de una línea, el cuadro de diálogo de introducción de texto del dispositivo muestra un campo de texto de una línea. El campo de texto se desplaza horizontalmente si el usuario introduce más caracteres de los que se pueden mostrar.

Cuadros de diálogo de introducción de texto (Flash Lite 4)

En las versiones anteriores de Flash Lite, el texto se introduce en un cuadro de diálogo modal independiente en lugar de interactuar con el campo de texto en pantalla directamente. La aplicación host controla este cuadro de diálogo y no Flash Lite. Durante esta interacción, el reproductor Flash Lite se detiene hasta que el usuario sale del cuadro de diálogo.

(En Flash Lite 4, se puede introducir texto directamente en el campo de texto en pantalla).

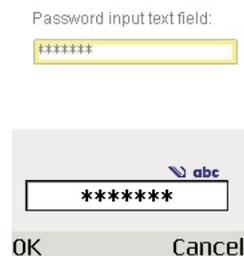
La imagen siguiente muestra el cuadro de diálogo de introducción de texto de un dispositivo para un campo de texto de una línea en una aplicación de Flash Lite 4:



Cuando se edita un campo de introducción de texto de varias líneas, el cuadro de diálogo de introducción de texto del dispositivo se expande para mostrar todo el texto introducido y se muestra la siguiente imagen:



Cuando se edita un campo de introducción de texto de contraseña, el cuadro de diálogo de introducción de texto del dispositivo muestra todos los caracteres introducidos. Cuando se hace clic en Aceptar, los caracteres de la contraseña se convierten en asteriscos y aparece la imagen siguiente:



Creación de una aplicación de ejemplo de campos de texto

Esta aplicación sencilla obtiene texto introducido por el usuario y, después, le aplica formato y lo muestra en un campo de texto dinámico compatible con HTML.

Para ver una aplicación de muestra finalizada (textfield_example fla) que utiliza esta técnica, consulte www.adobe.com/go/learn_ftl_samples_and_tutorials_es. Localice el archivo ZIP para su versión de ActionScript y descargue y descomprima este archivo. A continuación, desplácese a la carpeta Samples para acceder al ejemplo.

- 1 En Flash, cree un nuevo documento móvil y guárdelo como textfield fla.
- 2 Con ayuda de la herramienta Texto del panel Herramientas, cree un campo de texto de una línea en la parte superior del escenario.
- 3 Con el campo de texto aún seleccionado, en la ficha Propiedades, realice lo siguiente:
 - a Seleccione Texto clásico en el menú emergente Motor de texto.
 - b Seleccione Texto de entrada en el menú emergente Tipo de texto.
 - c Seleccione Utilizar fuentes del dispositivo en el menú emergente Método de representación de fuentes.
 - d Indique **inputText** en el campo de texto Nombre de instancia.
- 4 Cree otro campo de texto bajo el primero. Compruebe que el segundo campo de texto es mucho más largo que el primero.
- 5 Con el campo de texto aún seleccionado, en la ficha Propiedades, realice lo siguiente:
 - a Seleccione Texto clásico en el menú emergente Motor de texto.
 - b Seleccione Texto dinámico en el menú emergente Tipo de texto.
 - c Seleccione Utilizar fuentes del dispositivo en el menú emergente Método de representación de fuentes.
 - d Haga clic en el icono Generar texto como HTML para seleccionarlo.
 - e Indique **messageText** en el campo de texto Nombre de instancia.
- 6 En la línea de tiempo, seleccione el fotograma 1 en la capa 1.
- 7 Abra el panel Acciones (Ventana > Acciones) e introduzca el siguiente código:

```
messageText.htmlText = "";
inputText.addEventListener(TextEvent.TEXT_INPUT, onTextInput);
function onTextInput(e:TextEvent) {
    messageText.htmlText += "You Entered <i>" + e.text + "</i>\n";
}
```

El evento `TextInput` del campo de introducción de texto se puede detectar para cualquier entrada del campo de texto. En este ejemplo, el evento `TextInput` se utiliza para recibir el texto introducido y se muestra en el campo de texto dinámico `messageText`.

- 8 Guarde las modificaciones e inicie la aplicación (Control > Probar película).
- 9 Para utilizar la aplicación, pulse la tecla de selección del emulador para abrir el cuadro de diálogo de introducción de texto e introduzca algún texto desde el teclado del equipo. A continuación, haga clic en Aceptar para cerrar el cuadro de diálogo. El texto introducido aparecerá en cursiva en el campo de texto `messageText`.

Motor de texto de Flash

`flash.text.engine` se puede utilizar para crear y administrar elementos de texto sencillos, aunque se desarrolló para crear componentes de gestión de texto. `flash.text.engine` consta de clases que permiten crear, aplicar formato y controlar texto. El siguiente código de ejemplo muestra el uso de `flash.text.engine`:

```
import flash.text.engine.*;
var str:String="Hello World";
// Fontdescription represents properties necessary to describe a font, for example,
// font name. This example uses the embedded Arial font.
    var fontDescription:FontDescription = new FontDescription();
    fontDescription.fontName = "Arial"
    fontDescription.fontLookup = flash.text.engine.FontLookup.EMBEDDED_CFF;
// The Element format provides formatting information such as color, alpha, size, and
// so on. In this example, the color of text is green and the text size is 18.
    var format:ElementFormat=new ElementFormat(fontDescription);
    format.color = 0x00ff00;
    format.fontSize = 18;
// TextElement represents a formatted string, and is set to the content property
// of textBlock to create a block of text.
    var textElement:TextElement=new TextElement(str,format);
// textBlock represents a paragraph. It is a container for text and graphic elements.
    var textBlock:TextBlock = new TextBlock();
    textBlock.content=textElement;
// Creates a text line with a maximum of 500 characters.
    var textLine:TextLine=textBlock.createTextLine(null,500);
    textLine.y=20;
// Adds the text line to the display object for rendering to make the line display on
// the screen.
    addChild(textLine);
```

Para obtener más información sobre el modo en que funcionan las clases de `flash.text.engine`, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Representación de fuentes

Métodos de representación de fuentes en Flash Lite

Flash Lite puede representar las fuentes de los campos de texto de cualquiera de las formas siguientes:

Utilización de fuentes disponibles en el dispositivo Se puede aplicar una fuente a un campo de texto que se sepa que está disponible en el dispositivo. O bien, es posible especificar una de las tres fuentes del dispositivo genéricas (`_sans`, `_serif` o `_typewriter`) que se encuentran disponibles en el menú desplegable Fuente. Si selecciona una fuente del dispositivo genérica, Flash Lite intenta hacer coincidir la fuente genérica seleccionada con una fuente del dispositivo en tiempo de ejecución. Por ejemplo, `_sans` se asigna a una fuente *sans serif*, si está disponible.

Representación de fuentes como mapas de bits Flash Lite representa el texto de mapa de bits alineando los contornos de fuente con los límites de píxeles. Esto facilita la lectura del texto con tamaño pequeño (como diez puntos o menos). Esta opción requiere la inclusión de los contornos de fuente adecuados en el archivo SWF publicado. (Consulte [“Incorporación de contornos de fuente en archivos SWF”](#) en la página 41.)

Representación de fuentes como vectores suavizados Flash Lite representa el texto suavizado utilizando representaciones basadas en vectores de los contornos de fuente que se incorporan al archivo SWF publicado. (Consulte [“Incorporación de contornos de fuente en archivos SWF”](#) en la página 41.)

El método para representar fuentes de un campo de texto se debe seleccionar en el menú emergente Método de representación de fuentes que se encuentra en la ficha Propiedades. El menú emergente Método de representación de fuentes contiene cinco opciones de representación; sin embargo, los desarrolladores de Flash Lite sólo tienen tres disponibles. Los otros dos métodos (Suavizado para legibilidad y Suavizado personalizado) sólo están disponibles en aplicaciones destinadas a Flash Player 8 y versiones posteriores en equipos de escritorio.

A continuación se describen algunas directrices que deben tenerse en cuenta para elegir entre texto suavizado, de mapa de bits o de dispositivo:

- Si va a utilizar fuentes incorporadas con campos de texto dinámicos o de introducción de texto, incorpore los contornos únicamente para los caracteres que necesite mostrar. La reducción del número de contornos ayuda a reducir el tamaño del archivo. Por ejemplo, si utiliza un campo de introducción de texto para capturar la edad de un usuario (un número), incluya solamente el contorno de la fuente para caracteres numéricos (0-9). En este caso, también puede ser conveniente restringir la introducción de caracteres sólo a números.
- El emulador de Adobe Device Central no emula fuentes de dispositivo, a no ser que tenga instaladas en el equipo las mismas fuentes que va a utilizar para desarrollar el contenido. Por lo tanto, el diseño y el aspecto del campo de texto del dispositivo tal vez sea distinto en el emulador y en el dispositivo.
- Supongamos que se aplica uno de los tipos de fuentes del dispositivo genéricas (*_sans*, *_serif* o *_typewriter*). En este caso, Flash Lite intenta buscar una fuente similar en el dispositivo para mostrar el texto. Sin embargo, los dispositivos móviles suelen disponer de menos fuentes y estilos de fuente que un equipo de escritorio. Como resultado, es posible que una fuente como *_sans* no se asigne a una fuente sans serif. Asegúrese de probar la aplicación en cada dispositivo de destino para determinar los ajustes adecuados para cada fuente.

El texto suavizado en Flash Lite es, básicamente, una forma de vector complejo. Como ocurre con el resto de las formas de vector, necesita un buen procesado para poder representarse. Dado que la velocidad de procesado de la mayoría de los dispositivos es relativamente lenta, la animación de grandes cantidades de texto suavizado podría reducir el rendimiento de la aplicación. Para mejorar el rendimiento, intenten reducir temporalmente la calidad de representación del reproductor de Flash Lite durante la animación. Después, devuélvalo a una calidad de representación más elevada cuando se complete la animación.

Seleccionar un método de representación de fuentes para un campo de texto

- 1 Seleccione un campo de texto en el escenario.
- 2 En la ficha Propiedades, seleccione una de las opciones siguientes en el menú emergente Método de representación de fuentes:
 - Seleccione Utilizar fuentes del dispositivo para que Flash Lite utilice una fuente disponible en el dispositivo. No se incorpora ningún dato de fuentes en los archivos SWF publicados.
 - Seleccione Texto de mapa de bits (sin suavizado) para que Flash Lite alinee los contornos con los límites de píxeles (el texto pequeño aparecerá claro y nítido). Esta opción requiere la incorporación de los contornos de fuente al archivo SWF publicado. (Consulte ["Incorporación de contornos de fuente en archivos SWF"](#) en la página 41.)
 - Seleccione Suavizado para animación para que Flash Lite suavice la fuente del campo de texto de acuerdo con el ajuste actual de calidad de representación. Esta opción requiere la incorporación de los contornos de fuente al archivo SWF publicado.

Calidad de la representación y texto suavizado de Flash Lite

Utilice la propiedad `quality` del objeto Stage para controlar cómo representa Flash Lite los gráficos vectoriales y de mapas de bits. Los valores posibles son "LOW", "MEDIUM", "HIGH" y "BEST". El valor predeterminado es "HIGH".

Flash Lite representa el texto suavizado mediante representaciones de vectores de los contornos de fuente. Si desea que el texto suavizado aparezca lo más suave posible, debe definir la calidad de representación del reproductor como alta. La figura siguiente muestra un campo de texto suavizado (Arial, 24 puntos) con calidad alta:

*H i g h
q u a l i t y
s e t t i n g*

Es posible aumentar el rendimiento de la animación y la velocidad de los fotogramas. Observe el ejemplo siguiente. Durante una animación intensiva o secuencia de interpolación, la calidad de representación se puede establecer temporalmente en un valor inferior. Una vez completada la animación, se vuelve a la configuración anterior.

Incorporación de contornos de fuente en archivos SWF

Para representar una fuente de un campo de texto, Flash Lite puede utilizar las fuentes disponibles en el propio dispositivo o utilizar contornos de fuente incorporados en el archivo SWF publicado. Al incorporar contornos de fuentes en el archivo SWF, se garantiza que la fuente del campo de texto no parezca la misma en todas las plataformas de destino. Sin embargo, el tamaño del archivo aumenta considerablemente. Flash Lite necesita los contornos de fuente para representar texto en mapa de bits (sin suavizado) o texto suavizado.

En el caso de los campos de texto estático que utilicen los métodos de representación de fuentes suavizados o de mapa de bits, Flash incorpora automáticamente los contornos de fuente necesarios para mostrar el contenido de los campos de texto. Observe el ejemplo siguiente. Un campo de texto estático contiene la palabra *Submit*. En este caso, Flash incorpora automáticamente los contornos de fuente necesarios para mostrar los seis caracteres (*S, u, b, m, i y t*). El contenido de un campo de texto estático no va a cambiar, por lo que el archivo SWF sólo necesita incluir los contornos de fuente de esos caracteres.

Se especifican caracteres en los que se desea incorporar contornos de fuentes en el archivo SWF publicado. Esta especificación es necesaria para los campos de introducción de texto dinámicos que utilizan los métodos de suavizado o representación de fuentes de mapas de bits. El contenido de estos campos de texto puede cambiar durante la reproducción; por tanto, Flash no puede conocer de antemano los contornos de fuente que deben estar disponibles. Es posible incluir los contornos de todos los caracteres de la fuente seleccionada, un rango o unos caracteres específicos. Puede utilizar el cuadro de diálogo Incorporación de caracteres para especificar los caracteres que se incorporarán al archivo SWF publicado.

Incorporar contornos de fuente para un campo de texto dinámico o de introducción de texto

- 1 Seleccione el campo de texto dinámico o de introducción de texto en el escenario.
- 2 En el inspector de propiedades, seleccione Texto de mapa de bits (sin suavizado) o Suavizado para animación en el menú emergente Método de representación de fuentes.
- 3 Haga clic en el botón para incorporar para abrir el cuadro de diálogo de incorporación de fuentes.
- 4 Seleccione los caracteres de la lista que desee incorporar. También se pueden escribir los caracteres que se desean incorporar en el campo de texto para incluir también estos caracteres.
- 5 Haga clic en Aceptar.

Texto desplazable

Creación de texto desplazable

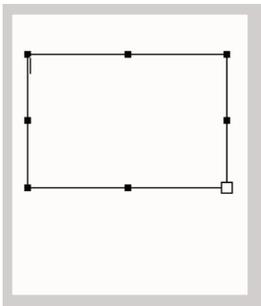
Flash Lite 4 admite las propiedades `TextField.scrollV`, `TextField.scrollH`, `TextField.maxscrollV` y `TextField.maxscrollH`, con las que es posible crear campos de texto desplazable. La propiedad `scrollV` especifica la primera línea visible de un bloque de texto. Es posible definir y obtener su valor. Por ejemplo, el código siguiente desplaza cinco líneas hacia abajo el campo de texto cuyo nombre de variable es `story_text`:

```
story_text.scrollV += 5;
```

La propiedad `maxscrollV` especifica la primera línea visible en un bloque de texto en el que la última línea del texto no es visible. Esta propiedad es de sólo lectura. Puede comparar la propiedad `maxscrollV` del campo de texto con su propiedad `scrollV` para averiguar hasta dónde se ha desplazado un usuario en un campo de texto. Resulta útil si desea crear una barra de desplazamiento vertical que ofrezca información sobre la posición de desplazamiento actual en relación con la posición de desplazamiento máximo. Las propiedades `TextField.scrollH` y `TextField.maxscrollH` presentan el mismo comportamiento.

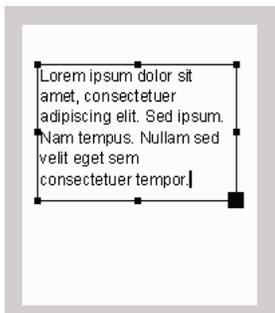
Crear un campo de texto desplazable y controlarlo con ActionScript

- 1 En Flash, cree un nuevo documento móvil.
- 2 Con la herramienta Texto, arrastre un campo de texto en el escenario que tenga un tamaño similar al mostrado en la imagen siguiente:



- 3 Seleccione Texto clásico en el menú emergente Motor de texto en la ficha Propiedades.
- 4 Seleccione Texto dinámico en el menú emergente Tipo de texto en la ficha Propiedades.
- 5 Seleccione Utilizar fuentes del dispositivo en el menú emergente Método de representación de fuentes en la ficha Propiedades.
- 6 En el inspector de propiedades, seleccione Multilínea en el menú emergente Tipo de línea de la ficha Propiedades.
- 7 Seleccione Texto > Desplazamiento permitido para que el usuario pueda desplazar el campo.
- 8 Escriba **story** en el campo de texto Nombre de instancia de la ficha Propiedades.

- 9 Haga doble clic en el interior del campo de texto e introduzca suficiente texto como para que una o varias líneas de texto se extiendan por debajo de su borde inferior.



- 10 En la línea de tiempo, seleccione el primer fotograma de la capa 1 y abra el panel Acciones. (Ventana > Acciones). Introduzca el código siguiente en el panel Acciones:

```
stage.focus = story;
story.addEventListener(KeyboardEvent.KEY_DOWN, onKeyEvent);

function onKeyEvent(e:KeyboardEvent): void {
    if (e.keyCode == 38) {
        story.scrollV++;
    }
    else if (e.keyCode == 40) {
        story.scrollV--;
    }
}
```

- 11 Seleccione Control > Probar película > En Device Central para probar la aplicación. Utilice las teclas arriba y abajo para mover el texto hacia arriba o hacia abajo.

Capítulo 6: Trabajo con sonido

Adobe Flash Lite admite dos tipos de sonido: sonido de dispositivo y sonido estándar (*nativo*) de Flash. Los sonidos del dispositivo se reproducen directamente en el dispositivo y no en el reproductor Flash Lite. Por el contrario, el sonido nativo se reproduce directamente en el reproductor Flash Lite, del mismo modo que en la versión de escritorio de Adobe Flash Player.

Algunos ejemplos de formato de sonido de dispositivo son MIDI o MFi, pero los formatos de sonido admitidos varían según el dispositivo.

Sonido de dispositivo

Sonido de dispositivo

Los sonidos del dispositivo hacen referencia al audio que se reproduce directamente en el dispositivo y no en el reproductor Flash Lite. Cada dispositivo admite distintos tipos de formato de sonido, incluidos MIDI, MFi o mp3. Para incorporar sonidos del dispositivo en su contenido de Flash Lite, puede incluirlos en el archivo SWF publicado. También puede cargar archivos de sonido externos en la red o desde el sistema de archivos local del dispositivo. Salvo algunas excepciones, puede utilizar la clase `Sound` de ActionScript para controlar el sonido del dispositivo, tal como lo haría en la versión de Flash Player para escritorio.

Reproducción de sonidos de dispositivo externos

En Flash Lite, puede cargar y reproducir archivos de sonido externos. Para reproducir sonidos de dispositivo externos, utilice el método `load()` de la clase `Sound`. Al igual que ocurre con el sonido de dispositivo empaquetado, el reproductor Flash Lite transfiere el audio cargado externamente al dispositivo para decodificarlo y reproducirlo.

Es importante recordar la siguiente información sobre la reproducción de sonido de dispositivo externo en Flash Lite:

- A diferencia de las aplicaciones de escritorio, los sonidos cargados de forma externa en el reproductor de Flash Lite siempre se tratan como sonidos de evento. Como resultado, los sonidos del dispositivo externo no se transmiten (se reproducen conforme se están cargando). Los datos de sonido deben descargarse completamente para poder reproducirlos. Por este mismo motivo, debe llamar al método `start()` de la clase `Sound` para reproducir el sonido una vez que esté completamente cargado (consulte el siguiente ejemplo de código).
- Flash Lite no reproduce de forma nativa archivos MP3 cargados externamente. Si la aplicación carga un archivo MP3 externo, Flash Lite transfiere los datos de sonido al dispositivo, del mismo modo que lo hace cualquier archivo de sonido cargado externamente. El dispositivo descodifica y reproduce este sonido.

El siguiente código muestra cómo cargar y reproducir un archivo de sonido externo:

```
//The following URLRequest assumes that the mp3 file is in the same directory as
//the SWF file when published. You can use a full URL for a remote file.
var MySound:Sound = new Sound(Areq);
MySound.addEventListener(Event.COMPLETE, onSoundLoaded);
var Areq:URLRequest = new URLRequest("click.mp3");
MySound.load(Areq);
function onSoundLoaded(event:Event):void
{
    var mysound:Sound = event.target as Sound;
    mysound.play();
}
```

Sonido nativo

Sonidos nativos de Flash

Además de admitir sonido de dispositivo, Flash Lite también admite sonidos de Flash estándar o *nativos*. Básicamente, un sonido nativo es un sonido en cualquier formato reconocible por la herramienta de edición de Flash. El sonido nativo de Flash Lite se puede reproducir como sonido de evento o como sonido sincronizado.

El flujo de trabajo general para utilizar sonidos nativos en aplicaciones de Lite 1.1 es el mismo que en las aplicaciones de escritorio Flash, con las siguientes excepciones:

- Flash Lite no admite la reproducción *nativa* de archivos MP3 externos. Sin embargo, si el dispositivo puede reproducir audio MP3, podrá cargar archivos MP3 externos en su aplicación.
- Flash Lite no admite la opción de compresión de audio Voz.

Utilización de la función de frecuencia de muestreo de 8 kHz

De forma predeterminada, la herramienta de edición de Flash exporta audio nativo a velocidades de muestreo de 5, 11, 22 o 44 kHz (kHz). Sin embargo, muchos dispositivos no admiten la reproducción de datos de audio a estas velocidades de muestreo. Por ejemplo, los dispositivos Symbian Series 60 admiten únicamente velocidades de 8 y 16 kHz. En versiones anteriores de Flash Lite, el reproductor volvía a muestrear el audio en tiempo de ejecución a 8 kHz antes de llamar a las API del sonido nativo del dispositivo para reproducir el sonido. Este proceso requería tiempo de procesamiento adicional e implicaba un consumo de memoria.

En Adobe Flash Professional CS5, puede elegir que la herramienta de edición vuelva a realizar el muestreo del audio nativo de la aplicación a 8 kHz durante el proceso de publicación del archivo SWF. Esta configuración se aplica a todos los sonidos de evento y transmisión en la aplicación.

El siguiente procedimiento explica cómo utilizar el cuadro de diálogo Configuración de publicación para establecer esta opción globalmente para todos los sonidos de flujo y evento.

Nota: en el caso de las secuencias de datos de sonido, Flash siempre aplica las opciones de compresión de sonido globales especificadas en el cuadro de diálogo Configuración de publicación. Las opciones de compresión por sonido especificadas en el cuadro de diálogo Propiedades de sonido se aplican únicamente a los sonidos de eventos.

Activación de la función de frecuencia de muestreo de 8 kHz para todos los sonidos nativos

- 1 En Flash, seleccione Archivo > Configuración de publicación.

- 2 En el cuadro de diálogo Configuración de publicación, haga clic en la ficha Flash.
- 3 Para activar la frecuencia de muestreo de 8 kHz para todos los flujos sonido de la aplicación, realice lo siguiente:
 - a Haga clic en el botón Establecer para definir las opciones de las secuencias de datos de sonido.
 - b En el cuadro de diálogo Configuración de sonido, seleccione mp3 en el menú emergente Compresión.
la función de muestreo de 8 kHz sólo está disponible para audio comprimido MP3.
 - c Seleccione la opción Usar frecuencia de muestreo de 8kHz.
 - d Haga clic en Aceptar para cerrar el cuadro de diálogo Configuración de publicación.
- 4 Para activar la frecuencia de muestreo de 8 kHz para todos los eventos de sonido, realice lo siguiente:
 - a Haga clic en el botón Establecer para definir las opciones de evento de audio.
 - b En el cuadro de diálogo Propiedades de sonido, seleccione mp3 en el menú emergente Compresión.
Nota: la función de muestreo de 8 kHz sólo está disponible para audio comprimido MP3.
 - c Seleccione la opción Usar frecuencia de muestreo de 8kHz.
 - d Haga clic en Aceptar.
- 5 Haga clic en Aceptar para cerrar el cuadro de diálogo Configuración de publicación.

Capítulo 7: Trabajo con vídeo

Flash Lite 4 descodifica y representa los formatos de vídeo del códec *On2* y *Sorenson Spark*. Esta versión de Flash Lite es compatible con la transmisión de vídeo que utiliza protocolos RTMP, RTMPE, RTMPT y RTMPTE. La función de transmisión de velocidad multibit ayuda a desconectar el vídeo en condiciones fluctuantes de red sin ninguna pausa en su reproducción.

Trabajo con archivos FLV

Compatibilidad de archivos FLV en Flash Lite 4

Flash Lite 4 admite archivos FLV utilizando versiones de los códecs *On2* y *Sorenson Spark* optimizados para dispositivos móviles. Flash Lite 4, no el dispositivo, representa los archivos FLV. Si un dispositivo admite Flash Lite 4, puede añadir archivos FLV a su aplicación.

Nota: los fabricantes de algunos dispositivos pueden seguir admitiendo archivos FLV si ya han implementado un mecanismo de reproducción de vídeo eficiente. En este caso, use vídeo de dispositivo en sus aplicaciones. Compruebe la ficha *Perfiles de dispositivos de Adobe Device Central* para ver si se admiten los archivos FLV () en un dispositivo determinado.

Puede incluir archivos FLV en una aplicación de cualquiera de las siguientes maneras:

- Incorporación de archivos FLV en la aplicación.
- Reproducción de archivos FLV externos desde una unidad de disco duro local o un servidor web a través de HTTP.
- Reproducción de archivos FLV externos desde Flash Media Server y a través de RTMP, RTMPE, RTMPT o RTMPTE.

También se puede convertir el vídeo (por ejemplo, archivos de vídeo de QuickTime o Windows Media) en archivos FLV. Para ello:

- Use un codificador (como Adobe Media Encoder CS5 o Sorenson™ Squeeze).
- Use el plug-in FLV Export para exportar archivos FLV desde aplicaciones de edición de vídeo admitidas.

Las siguientes funciones de vídeo no están disponibles en Flash Lite 4:

- La clase Camera o la grabación de vídeo.
- Conexiones RTMPS y RTMFP
- Vídeo de canal alfa
- El componente FLVPlayback. Para obtener más información, consulte [“Acerca de los componentes de Flash Lite”](#) en la página 1.

Incorporación de archivos FLV en una aplicación

Para incorporar un archivo FLV, impórtelo a la biblioteca de un documento. Al publicar la aplicación, la herramienta de edición incorpora el archivo FLV en el archivo SWF publicado.

Los archivos FLV incorporados están asociados a la línea de tiempo. Por ejemplo, si un documento de Flash se configura para reproducirse a 10 fotogramas por segundo, un archivo FLV de 10 segundos rellena la línea de tiempo del fotograma 1 al 100. Para controlar la reproducción de un archivo FLV incorporado, use ActionScript de para reproducir, detener y mover la cabeza lectora por la línea de tiempo. También puede asignar al objeto de vídeo incorporado un nombre de instancia y controlarlo con las propiedades del objeto Video de ActionScript.

Al incorporar archivos FLV, puede sincronizarlos con otros elementos del documento. Por ejemplo, puede añadir elementos interactivos a determinados fotogramas que vinculen a otros documentos. Se recomienda incorporar archivos FLV si los archivos son pequeños y no incluyen ninguna pista de sonido. La incorporación de archivos FLV puede incrementar de manera significativa el tamaño del archivo SWF publicado.

- 1 En Flash, haga clic en Flash Lite 4 en Crear nuevo. Se abrirá un nuevo archivo.
- 2 Seleccione Archivo > Importar > Importar vídeo.
Aparecerá el asistente para importación de vídeo.
- 3 En la pantalla Seleccionar vídeo, seleccione la opción para importar un archivo de su equipo y haga clic en Examinar.
- 4 Acceda a la carpeta que contiene el archivo FLV y selecciónelo.
Si no ve el archivo de vídeo en la lista (o bien, lo ve pero no lo puede seleccionar), seleccione Todos los archivos (*.*) en el menú desplegable Tipo de archivo (Windows) o seleccione Todos los archivos en el menú emergente Activar (Macintosh).
- 5 Haga clic en Abrir. Seleccione la opción Incorporar FLV en SWF y reproducir en la línea de tiempo y haga clic en Siguiente.
- 6 En la pantalla Incorporación, seleccione vídeo incorporado como tipo de símbolo y haga clic en Siguiente.
- 7 En la pantalla Finalizar importación de vídeo, haga clic en Finalizar para importar el vídeo.
Aparecerá un nuevo símbolo de vídeo vinculado al archivo FLV en el panel Biblioteca y en el escenario.
- 8 Seleccione el objeto Video del escenario e introduzca un nombre de instancia para el objeto Video en la ficha Propiedades.
Puede utilizar el nombre de instancia para controlar el objeto Video con las propiedades de la clase Video de ActionScript.

Reproducción de archivos FLV externos en una aplicación

Un archivo FLV externo no se incorpora a un archivo SWF. Un archivo FLV externo reside en la tarjeta de memoria de un dispositivo, en un servidor web remoto o en Adobe® Flash® Media Server®. Puede utilizar ActionScript para reproducir y controlar archivos FLV externos en Flash Lite 4 en tiempo de ejecución.

La reproducción de archivos FLV externos aporta varias ventajas con respecto a la incorporación de vídeo en un documento de Flash. Puede usar clips de vídeo largos sin ralentizar la reproducción, ya que los archivos FLV externos necesitan menos cantidad de memoria que los archivos de vídeo incorporados. Los archivos FLV externos pueden tener velocidades de fotogramas distintas a las del documento de Flash en el que se reproduzcan. Por ejemplo, puede establecer la velocidad de fotogramas del documento de Flash en 30 fotogramas por segundo (fps) y la velocidad de fotogramas del vídeo en 21 fps. Este control garantiza una reproducción del vídeo suave. Este control también le permite reproducir archivos FLV a velocidades de fotogramas diferentes sin tener que modificar contenido multimedia enriquecido existente.

Puede manipular objetos Video al igual que puede manipular objetos MovieClip. Puede rotar y escalar el vídeo. Puede configurar el valor de transparencia alfa. Puede componer y mezclar el vídeo con otro contenido multimedia enriquecido. También puede usar la clase Sound para controlar el sonido del vídeo.

- 1 En la biblioteca de Flash, haga clic con el botón derecho y en las opciones seleccione Nuevo vídeo para crear un símbolo de vídeo. Asegúrese de que el tipo de vídeo es “Vídeo (controlado por ActionScript)” y haga clic en Aceptar.
- 2 Arrastre el símbolo de vídeo al escenario e indique una instancia de nombre en la ficha Propiedades. Este ejemplo utiliza el nombre de instancia MyVideo.
- 3 En la línea de tiempo, seleccione el primer fotograma y abra el panel Acciones.
- 4 Introduzca el código siguiente :

```
var MyNC:NetConnection = new NetConnection();
MyNC.connect(null);

var MyNS:NetStream = new NetStream(MyNC);
MyNS.play("http://www.helpexamples.com/flash/video/clouds.flv");

MyVideo.attachNetStream(MyNS);

//The clouds.flv video has metadata that Adobe is not using, so create
//an error handler to ignore the message generated by the runtime
//about the metadata.
MyNS.addEventListener(AsyncErrorEvent.ASYNC_ERROR, asyncErrorHandler);
function asyncErrorHandler(event:AsyncErrorEvent):void
{
    //Ignore metadata error message.
}
```

Nota: para cargar archivos FLV desde un servidor web, registre la extensión de nombre de archivo y el tipo MIME con el servidor web. El tipo MIME para los archivos flv es video/x-flv. Es posible que el tipo MIME ya se haya registrado con el servidor web. Para obtener más información, consulte la documentación del servidor web.

Control de los archivos FLV con ActionScript

Use ActionScript para conectarse a, ver y controlar archivos FLV externos. Use la clase NetConnection para establecer una conexión. Utilice la clase Video para ver el vídeo. Use la clase NetStream para controlar la reproducción. Para controlar los archivos FLV, llame a los métodos de la clase NetStream; por ejemplo, el método `NetStream.pause()`.

Consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform* para obtener descripciones detalladas de estas clases, junto con ejemplos de código que ilustren su uso. Asimismo, Flash Lite 4 admite todos los métodos, propiedades y eventos incluidos en *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*). La única excepción es la propiedad `MovieClipLoader.checkPolicyFile`.

RTMP

El canal de datos del protocolo de mensajería en tiempo real (Real Time Messaging Protocol, RTMP) se utiliza para transmitir datos de vídeo entre un cliente (reproductor de Flash Lite) y el servidor (Flash Media Server). El reproductor de Flash Lite y el servidor comunican estos datos en forma de objetos ActionScript.

Para serializar y deserializar objetos de ActionScript mientras se envían a través de la red, utilice un formato binario compacto denominado Action Message Format (AMF). La primera versión del formato, AMF 0, se utiliza en ActionScript 2.0. ActionScript 3.0 emplea el formato AMF3 para serializar y deserializar objetos de ActionScript. AMF 3 envía cadenas y características de objetos además de instancias de objetos. AMF 3 también admite los tipos de datos introducidos en ActionScript 3.0.

Para obtener más información sobre el formato AMF 3, consulte la [especificación AMF 3](#) (en inglés). De forma predeterminada, RTMP utiliza el puerto 1935.

En el siguiente ejemplo se utiliza la instancia `MyVideo` para cargar vídeo desde Flash Media Server utilizando RTMP:

```
var MyVideo:Video = new Video();
addChild(MyVideo);
var nc:NetConnection = new NetConnection();
var ns:NetStream;
// Add event listeners for the connection status and errors.
nc.addEventListener(NetStatusEvent.NET_STATUS, netStatusHandler)
nc.addEventListener(AsyncErrorEvent.ASYNC_ERROR, asyncErrorHandler);
// Provide the video location.
nc.connect("rtmp://fmsexamples.adobe.com/vod");

function netStatusHandler(event:NetStatusEvent):void {
    trace(event.info.code);
    if(event.info.code == "NetConnection.Connect.Success"){
        // Create a NetStream object and pass it the NetConnection object.
        ns = new NetStream(nc);
        // Assign the ns.client property to the object that is handling onMetaData.
        ns.client = this;
        // Attach the NetStream object to the video object on Stage.
        MyVideo.attachNetStream(ns);
        // Play the file "sample.flv". Do not include ".flv" when
        // playing the file on Flash Media Server.
        ns.play("sample");
    }
}

function asyncErrorHandler(event:AsyncErrorEvent):void {
    // ignore error
}

function onMetaData(item:Object):void {
    // See the metadata in the trace output.
    var key:String;
    for (key in item){
        trace(key + ": " + item[key]);
    }
}
```

Mejoras

En esta versión de Flash Lite 4 se han actualizado algunas funciones relacionadas con el vídeo.

RTMPE

Si desea proteger el transporte de los datos de vídeo de transmisión de Flash Media Server al reproductor de Flash Lite, puede activar el cifrado. Los datos se pueden cifrar mediante RTMPE.

Flash Media Server cifra todo el contenido en tiempo de ejecución, lo que significa que no es necesario cifrar el archivo de origen. RTMPE utiliza los métodos originales de cifrado estándar de la industria que constan del *intercambio de claves Diffie-Hellman* y *HMACSHA256*. Mientras que se transmiten los datos, RTMPE genera un par de claves RC4. Una clave cifra los datos enviados por el servidor y la otra descifra la información enviada al servidor.

RTMPE evita que las aplicaciones de terceros puedan detectar la transferencia de datos entre el cliente y el servidor. Para cifrar un flujo de datos, se utiliza la siguiente sintaxis:

```
nc.connect("rtmpe://localhost/vod/");
```

Si utiliza el parámetro de origen del componente FLVPlayback, la ruta es la siguiente:

```
rtmpe://localhost/vod/flvFileName
```

Al añadir la letra “e”, se indica a Flash Media Server que añada el cifrado en tiempo real al flujo de datos. RTMPE cifra los datos mientras se transmiten. No se requiere ninguna clave para descifrar los datos.

Únicamente el reproductor de Flash Lite puede finalizar RTMPE. Si una aplicación utiliza RTMPE sin especificar ningún puerto, Flash Player detecta los puertos en el siguiente orden: 1935 (RTMPE), 443 (RTMPE) y 80 (RTMPE). RTMPE se activa de forma predeterminada. Para desactivar RTMPE, abra el archivo `fms.ini` ubicado en `C:\Program Files\Adobe\Flash Media Server 3\conf` y establezca el parámetro `ADAPTOR.RTMPE_ENABLED` en `off`. Reinicie el servidor tras el cambio.

RTMPT

Muchas organizaciones cuentan con firewalls y servidores proxy que no permiten las conexiones de socket RTMP directas. Como resultado, gran parte de estas organizaciones no pueden utilizar las aplicaciones de Flash Media Server.

Este problema se aborda enviando datos RTMP incluidos en solicitudes HTTP. Este método se denomina tunelación. En el reproductor de Flash Lite, es posible cargar la tunelación de forma explícita o automática.

En esta versión de Flash Lite, `netConnection.connect()` se amplía para admitir un nuevo protocolo, Tunelación RTMP (RTMPT). RTMPT significa RTMP tunelado mediante HTTP; los datos RTMP se encapsulan como datos HTTP. De forma predeterminada, RTMPT utiliza el puerto 80. Para llamar a RTMPT de forma explícita, se utiliza la siguiente sintaxis:

```
nc.connect("rtmpt://myserver/myapp"); // defaults to port 80
```

Asimismo se especifica el puerto:

```
nc.connect("rtmpt://myserver:port/myapp");
```

Reintentos automáticos

La función de "reintento automático" del reproductor de Flash Lite se amplía para incluir un escenario de conexión especial. Los reintentos automáticos se producen únicamente cuando una conexión se especifica como RTMP con un puerto predeterminado. Considere la siguiente sintaxis:

```
nc.connect("rtmpt://myserver/myapp");
```

Este comando prueba los siguientes puertos en secuencia:

```
nc.connect("rtmp://myserver/myapp"); // Defaults to port 1935.  
nc.connect("rtmp://myserver:443/myapp");  
nc.connect("rtmp://myserver:80/myapp");
```

RTMPTE

Flash Media Server 3 introduce RTMPTE, que permite cifrar RTMP. El puerto predeterminado es 80. RTMPTE asegura los datos incluso cuando se transfieren desde Flash Media Server a servidores de terceros mediante la tunelación.

En esta versión de Flash Lite, `netConnection.connect()` se amplía para admitir un nuevo protocolo, Cifrado de tunelación RTMP (RTMPTE). Significa que los datos RTMP cifrados se encapsulan como datos HTTP.

Para cifrar un flujo de datos tunelado, se emplea la siguiente sintaxis:

```
nc.connect("rtmpte://localhost/vod/");
```

Si utiliza el parámetro de origen del componente `FLVPlayback`, la ruta es la siguiente:

```
rtmpte://localhost/vod/flvFileName
```

En el protocolo RTMPTE, se indica a Flash Media Server que añada cifrado en tiempo real al flujo de datos de tunelación. RTMPTE cifra los datos mientras se transmiten. No se requiere ninguna clave para descifrar los datos.

Transmisión con varias velocidades de bits

Flash Lite 4 admite el cambio de velocidad de bits dinámico cuando la transmisión se realiza desde Flash Media Server. Como resultado, la aplicación puede adaptarse a los cambios dinámicos en las condiciones de la red mientras se transmite el contenido grabado.

Los editores de contenido en línea se encuentran con un problema principal al intentar transmitir vídeo que es de calidad más elevada (especialmente vídeo HD) y de mayor duración. El problema es el de las conexiones de red fluctuantes de los usuarios. El problema puede verse aumentado cuando se aplican las siguientes condiciones a una sola conexión:

- Usuarios multitarea.
- Redes inalámbricas fluctuantes.
- Usuarios simultáneos.

El resultado es un destino es un destino con movimiento para el ancho de banda que esté disponible. Como resultado, los usuarios terminan con un nuevo almacenamiento en búfer y esperando la reproducción del vídeo si el ancho de banda seleccionado no es compatible con la red.

La transmisión de flujo dinámica resuelve este problema. Detecta el ancho de banda fluctuante y cambia entre los flujos de diferentes velocidades de bits para coincidir con la transmisión de contenido en el ancho de banda del usuario.

Por otra parte, algunos usuarios inician la transmisión con un ancho de banda bajo disponible y, posteriormente, liberan más ancho de banda una vez que se inicia la reproducción de vídeo. En este escenario, la transmisión de flujo dinámica puede ofrecer la capacidad de aumentar la calidad de vídeo a un mayor nivel, mejorando una vez más la experiencia del usuario.

Otro problema que puede dificultar la reproducción (especialmente con vídeo HD de grandes dimensiones y la reproducción en pantalla completa) puede ser las limitaciones de rendimiento de hardware del usuario. Si la CPU no puede decodificar el flujo de vídeo con la rapidez adecuada, se eliminarán algunos fotogramas. Esto afecta negativamente a la agilidad de reproducción del vídeo del usuario. En este caso, la utilización de un archivo de vídeo menor calidad permite una menor carga en la CPU a la hora de decodificar en sincronización y mantener el rendimiento.

Con anterioridad, la alternativa era realizar rutinas de detección de ancho de banda frecuentes o iniciales. Sin embargo, estas pruebas no resultaban rentables y en ocasiones no proporcionaban la precisión necesaria en las fluctuaciones normales y los cambios de ancho de banda. Ahora, con las capacidades de transmisión de flujo dinámica y la información de la calidad del servicio (QoS) disponible, las pruebas de detección de ancho de banda han perdido su vigencia.

La transmisión de flujo dinámica puede suponer una solución óptima frente a las fluctuaciones de red y la sobrecarga de la CPU. Con el control continuo de la medición de la calidad del servicio (QoS) clave en el cliente, la transmisión de flujo dinámica puede identificar de forma eficaz cuándo aumentar o disminuir los flujos de calidad diferente.

Capítulo 8: Trabajo con las API del dispositivo

Con Flash Lite 4 es posible obtener la localización geográfica y los valores de aceleración direccional del dispositivo. Para este propósito, Flash Lite 4 admite dos API de dispositivo.

Localización geográfica

La clase Geolocation permite registrarse para las actualizaciones de localización desde el sensor de localización integrado de un dispositivo. Si un dispositivo que aloja Flash Lite admite esta función, puede obtener la localización geográfica actual del dispositivo. La localización geográfica se devuelve en forma de coordenadas longitudinales y latitudinales.

Si la localización cambia, se recibirán actualizaciones sobre los cambios en el dispositivo. Con el uso de esta función, se puede obtener la información siguiente:

- **Latitud:** especifica la latitud de la localización del dispositivo. Los valores de latitud presentan el siguiente intervalo:
 $-90^\circ \leq \text{Lat} \leq +90^\circ$
La latitud negativa es “Sur” y la positiva es “Norte”.
- **Longitud:** especifica la longitud de la localización del dispositivo. Los valores de longitud presentan el siguiente intervalo:
 $-180^\circ \leq \text{Long} < +180^\circ$
La longitud negativa es “Oeste” y la positiva es “Este”.
- **Altitud:** especifica la altitud de la localización del dispositivo. La altitud se mide en metros (m).
- **Precisión horizontal:** especifica el intervalo de desviación a partir de las coordenadas longitudinales exactas y devuelve el valor en metros. Por ejemplo, 50 m desde 5° O.
- **Precisión vertical:** especifica el intervalo de desviación a partir de las coordenadas latitudinales exactas y devuelve el valor en metros. Por ejemplo, 10 m desde 20 N.
- **Dirección:** especifica la dirección de movimiento respecto al norte verdadero o geográfico y devuelve el valor en grados. Por ejemplo, 10° significa que el dispositivo se está moviendo con un ángulo de 10° respecto al norte verdadero.
- **Velocidad:** especifica la velocidad con la que se mueve el dispositivo y devuelve el valor en m/seg. Por ejemplo, 0,010 m/seg.
- **Marca de tiempo:** especifica la duración de tiempo desde que se inició la aplicación. El valor se mide en milisegundos. La marca de tiempo es dinámica y el valor continúa cambiando en función del cambio de localización que se produce durante un intervalo de tiempo especificado.

Distribución de eventos

La clase Geolocation distribuye eventos basados en cualquier actividad detectada por el sensor de localización del dispositivo.

Es posible controlar el intervalo de tiempo para las actualizaciones de localización geográfica utilizando el método `setRequestedUpdateInterval()`. El intervalo de tiempo se mide en milisegundos.

Nota: el intervalo de actualización sólo se utiliza como sugerencia para conservar la batería. El tiempo real entre las actualizaciones de localización puede ser superior o inferior a este valor.

Si no se utiliza el método `setRequestedUpdateInterval()`, el dispositivo de host devuelve actualizaciones en función de su intervalo predeterminado.

Utilización de la localización geográfica

El siguiente ejemplo de código muestra el funcionamiento de la localización geográfica. En primer lugar, el código comprueba si el dispositivo admite la localización geográfica. Si es así, se registra un detector de eventos para detectar los cambios en la localización una vez iniciada la aplicación. Siempre que se detecte un cambio, se llama a la función `gpsListener` y se devuelve la información de coordenada diferente. El intervalo de tiempo para las actualizaciones de aceleración se establece en 5 segundos. Para obtener más información, consulte “[Localización geográfica](#)” en la página 54.

```
import flash.sensors.*;
if (Geolocation.isSupported)
{
    var gps:Geolocation = new Geolocation();
    function gpsListener(evt:GeolocationEvent):void
    {
        trace("Latitude: " + evt.latitude.toString());
        trace("Longitude: " + evt.longitude.toString());
        trace("Altitude: " + evt.altitude.toString());
        trace("Horizontal Accuracy: " + evt.horizontalAccuracy.toString());
        trace("Vertical Accuracy: " + evt.verticalAccuracy.toString());
        trace("Speed: " + evt.speed.toString());
        trace("Heading: " + evt.heading.toString());
        trace("Timestamp: " + evt.timestamp.toString());
    }
    gps.addEventListener(GeolocationEvent.UPDATE, gpsListener);
    gps.setRequestedUpdateInterval(5000);
    gps.removeEventListener(GeolocationEvent.UPDATE, gpsListener);
}
else
{
    trace("GPS not supported.");
}
```

Para obtener información sobre las clases `Geolocation` y `GeolocationEvent`, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Acelerómetro

La clase `Accelerometer` permite obtener los datos de aceleración de un dispositivo mediante un sensor acelerómetro instalado en el dispositivo. Estos datos se pueden representar de la siguiente manera:

- La localización del dispositivo en el espacio a lo largo de un eje tridimensional.
- El movimiento del dispositivo a lo largo del eje tridimensional.

El eje tridimensional consta de los ejes X, Y y Z. Cuando el dispositivo se mueve, el sensor detecta este movimiento y devuelve las coordenadas de aceleración del dispositivo. Para obtener más información sobre la orientación de los ejes tridimensionales, consulte “Orientación del eje” en la página 56.

También se puede obtener la marca de tiempo de los cambios de aceleración. La marca de tiempo especifica la duración de tiempo desde que se inició la aplicación. El valor se mide en milisegundos. La marca de tiempo es dinámica y el valor continúa cambiando en función del cambio de aceleración que se produce para un intervalo de tiempo especificado.

Distribución de eventos

La clase Accelerometer distribuye eventos basados en cualquier actividad detectada por el sensor de movimiento del dispositivo.

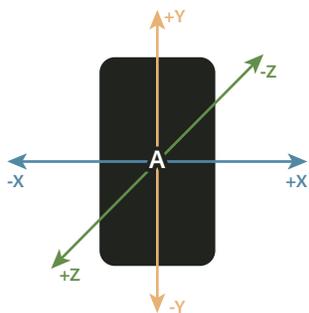
Es posible controlar el intervalo de tiempo para las actualizaciones de aceleración utilizando el método `setRequestedUpdateInterval()`. El intervalo de tiempo se mide en milisegundos.

Nota: el intervalo de actualización sólo se utiliza como sugerencia para conservar la batería. El tiempo real entre las actualizaciones de aceleración puede ser superior o inferior a este valor.

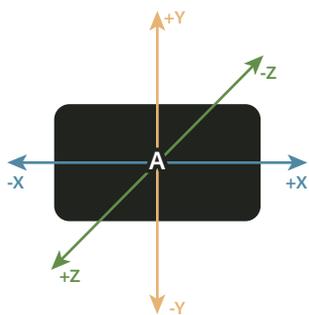
Si no utiliza el método `setRequestedUpdateInterval()`, el dispositivo host devuelve las actualizaciones en función de su intervalo predeterminado.

Orientación del eje

Los ejes tridimensionales se basan en la orientación del contenido mostrado en el dispositivo, tal y como se muestra en los siguientes mensajes.



A. Orientación del eje en modo vertical



A. Orientación del eje en modo horizontal

La aceleración a través de de los ejes se define de la siguiente manera:

- Aceleración a través del eje X: el valor de aceleración se mide en Gs (9,81 m/seg²). El eje X comienza desde la izquierda del dispositivo hasta su derecha cuando el dispositivo es paralelo al suelo. La aceleración es positiva si el dispositivo se mueve hacia la derecha.
- Aceleración a través del eje Y: el valor de aceleración se mide en Gs (9,81 m/seg²). El eje Y comienza desde la parte inferior del dispositivo hasta su parte superior cuando el dispositivo es paralelo al suelo. La aceleración es positiva si el dispositivo se mueve hacia el frente.
- Aceleración a través del eje Z: el valor de aceleración se mide en Gs (9,81 m/seg²). El eje Y es perpendicular a la cara del dispositivo cuando éste es paralelo al suelo. La aceleración es positiva si el dispositivo se aleja del suelo.

Utilización del acelerómetro

El siguiente ejemplo de código muestra el funcionamiento del acelerómetro. El primer lugar, el código comprueba si el dispositivo admite el acelerómetro. Si es así, un detector de eventos se registra para detectar los cambios en la aceleración una vez se haya iniciado la aplicación. Siempre que se detecte un cambio, se llama a la función `accListener` y se devuelve la información de aceleración diferente. El intervalo de tiempo para las actualizaciones de aceleración se establece en 5 segundos. Para obtener más información, consulte “[Acelerómetro](#)” en la página 55.

```
import flash.sensors.*;
if ((Accelerometer.isSupported)
{
    var acc:Accelerometer = new Accelerometer();
    function accListener(evt:AccelerometerEvent)
    {
        trace("AccelerationX: " + evt.accelerationX.toString());
        trace("AccelerationY: " + evt.accelerationY.toString());
        trace("AccelerationZ: " + evt.accelerationZ.toString());
        trace("Timestamp: "+evt.timestamp.toString());
    }
    acc.addEventListener(AccelerometerEvent.UPDATE, accListener);
    acc.setRequestedUpdateInterval(5000);
    acc.removeEventListener(AccelerometerEvent.UPDATE, accListener);
}
else
{
    trace("Accelerometer not supported.");
}
```

Para obtener más información sobre las clases `Accelerometer` y `AccelerometerEvent`, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Capítulo 9: Introducción a ActionScript en Flash Lite 4

El software de Adobe Flash Lite 4 se basa en Flash Player 10, pero difiere en los siguientes puntos:

- Flash Lite admite algunas funciones sólo de forma parcial.
- Flash Lite añade algunas funciones específicas para dispositivos móviles.

Clases y elementos de idioma de ActionScript admitidos, parcialmente admitidos y no admitidos

Acelerómetro

La siguiente tabla muestra los miembros de la clase Accelerometer que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
Accelerometer()	Sí	Sí
isSupported : Boolean	Sí	Sí
muted : Boolean	Sí	No
setRequestedUpdateInterval(interval:Number):void	Sí	Sí
update	Sí	Sí

AccelerometerEvent

La siguiente tabla muestra los miembros de la clase AccelerometerEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
accelerationX : Number	Sí	Sí
accelerationY : Number	Sí	Sí
accelerationZ : Number	Sí	Sí
AccelerometerEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, timestamp:Number = 0, accelerationX:Number = 0, accelerationY:Number = 0, accelerationZ:Number = 0)	Sí	Sí
timestamp : Number	Sí	Sí
clone():Event	Sí	No
toString() : String	Sí	No
UPDATE : String = "accelerometerUpdate"	Sí	Sí

ActionGenerator

La siguiente tabla muestra los miembros de la clase ActionGenerator que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ActionGenerator()	Sí	Sí
generateAction(a:AutomationAction):void	Sí	Sí
generateActions(actionList:Array):void	Sí	Sí

ActionScriptVersion

La siguiente tabla muestra los miembros de la clase ActionScriptVersion que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ACTIONSCRIPT3 : uint = 3	Sí	Sí

ApplicationDomain

La siguiente tabla muestra los miembros de la clase ApplicationDomain que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
currentDomain : ApplicationDomain	Sí	Sí
domainMemory : ByteArray	Sí	Sí
MIN_DOMAIN_MEMORY_LENGTH : uint	Sí	Sí
parentDomain : ApplicationDomain	Sí	Sí
ApplicationDomain(parentDomain:ApplicationDomain = null)	Sí	Sí
constructApplicationDomainObject(parentDomain:ApplicationDomain = null):*	Sí	Sí
getDefinition(name:String):Object	Sí	Sí
hasDefinition(name:String):Boolean	Sí	Sí

ArgumentError

La siguiente tabla muestra los miembros de la clase ArgumentError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ArgumentError(message:String = "")	Sí	Sí

arguments

La siguiente tabla muestra los miembros de la clase arguments que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
callee : Function	Sí	Sí
length: Number	Sí	Sí

Array

La siguiente tabla muestra los miembros de la clase Array que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
length : uint	Sí	Sí
Array(... values)	Sí	Sí
Array(numElements:int = 0)	Sí	Sí
concat(... args):Array	Sí	Sí
every(callback:Function, thisObject:* = null):Boolean	Sí	Sí
filter(callback:Function, thisObject:* = null):Array	Sí	Sí
forEach(callback:Function, thisObject:* = null):void	Sí	Sí
indexOf(searchElement:* , fromIndex:int = 0):int	Sí	Sí
join(sep:String):String	Sí	Sí
lastIndexOf(searchElement:* , fromIndex:int = 0x7fffffff):int	Sí	Sí
map(callback:Function, thisObject:* = null):Array	Sí	Sí
pop():*	Sí	Sí
push(... args):uint	Sí	Sí
reverse():Array	Sí	Sí
shift():*	Sí	Sí
slice(startIndex:int = 0, endIndex:int = 16777215):Array	Sí	Sí
some(callback:Function, thisObject:* = null):Boolean	Sí	Sí
sort(... args):Array	Sí	Sí
sortOn(fieldName:Object, options:Object = null):Array	Sí	Sí
splice(startIndex:int, deleteCount:uint, ... values):Array	Sí	Sí
toLocaleString():String	Sí	Sí
toString() : String	Sí	Sí
unshift(... args):uint	Sí	Sí
CASEINSENSITIVE : uint = 1	Sí	Sí
DESCENDING : uint = 2	Sí	Sí
NUMERIC : uint = 16	Sí	Sí
RETURNINDEXEDARRAY : uint = 8	Sí	Sí
UNIQUESORT : uint = 4	Sí	Sí

AsyncErrorEvent

La siguiente tabla muestra los miembros de la clase AsyncErrorEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
AsyncErrorEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "", error:Error = null)	Sí	Sí
clone():Event	Sí	Sí
toString(): String	Sí	Sí
ASYNC_ERROR : String = "asynError"	Sí	Sí

AutomationAction

La siguiente tabla muestra los miembros de la clase AutomationAction que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
type: String	Sí	Sí
AutomationAction(type:String)	Sí	Sí

AVM1Movie

Para obtener información sobre los miembros de la clase AVM1Movie que admiten ActionScript 3.0 y Flash Lite 4, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Bitmap

La siguiente tabla muestra los miembros de la clase Bitmap que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
bitmapData : BitmapData	Sí	Sí
pixelSnapping : String	Sí	Sí
smoothing: Boolean	Sí	Sí
Bitmap(bitmapData:BitmapData = null, pixelSnapping:String = "auto", smoothing:Boolean = false)	Sí	Sí
constructBitmapObject(bitmapData:BitmapData = null, pixelSnapping:String = "auto", smoothing:Boolean = false):void	Sí	Sí

BitmapData

La siguiente tabla muestra los miembros de la clase BitmapData que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
height : int	Sí	Sí
rect : Rectangle	Sí	Sí
transparent : Boolean	Sí	Sí
width : int	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
BitmapData(width:int, height:int, transparent:Boolean = true, fillColor:uint = 0xFFFFFFFF)	Sí	Sí
applyFilter(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, filter:BitmapFilter):void	Sí	No
clone() : BitmapData	Sí	Sí
colorTransform(rect:Rectangle, colorTransform:flash.geom.ColorTransform):void	Sí	Sí
compare(otherBitmapData:BitmapData) : Object	Sí	Sí
constructBitmapDataObject(width:int, height:int, transparent:Boolean = true, fillColor:uint = 0xFFFFFFFF):void	Sí	Sí
copyChannel(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, sourceChannel:uint, destChannel:uint):void	Sí	Sí
copyPixels(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, alphaBitmapData:BitmapData = null, alphaPoint:Point = null, mergeAlpha:Boolean = false):void	Sí	Sí
dispose():void	Sí	Sí
draw(source:IBitmapDrawable, matrix:Matrix = null, colorTransform:flash.geom.ColorTransform = null, blendMode:String = null, clipRect:Rectangle = null, smoothing:Boolean = false):void	Sí	Sí
fillRect(rect:Rectangle, color:uint):void	Sí	Sí
floodFill(x:int, y:int, color:uint):void	Sí	Sí
generateFilterRect(sourceRect:Rectangle, filter:BitmapFilter) : Rectangle	Sí	No
getColorBoundsRect(mask:uint, color:uint, findColor:Boolean = true):Rectangle	Sí	Sí
getPixel(x:int, y:int):uint	Sí	Sí
getPixel32(x:int, y:int):uint	Sí	Sí
getPixels(rect:Rectangle):ByteArray	Sí	Sí
getVector(rect:Rectangle):Vector.<uint>	Sí	Sí
histogram(hRect:Rectangle = null):Vector.<Vector.<Number>>	Sí	Sí
hitTest(firstPoint:Point, firstAlphaThreshold:uint, secondObject:Object, secondBitmapDataPoint:Point = null, secondAlphaThreshold:uint = 1):Boolean	Sí	Sí
lock():void	Sí	No
merge(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, redMultiplier:uint, greenMultiplier:uint, blueMultiplier:uint, alphaMultiplier:uint):void	Sí	Sí
noise(randomSeed:int, low:uint = 0, high:uint = 255, channelOptions:uint = 7, grayScale:Boolean = false):void	Sí	No

Nombre de miembro	ActionScript 3.0	Flash Lite 4
paletteMap(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, redArray:Array = null, greenArray:Array = null, blueArray:Array = null, alphaArray:Array = null):void	Sí	Sí
perlinNoise(baseX:Number, baseY:Number, numOctaves:uint, randomSeed:int, stitch:Boolean, fractalNoise:Boolean, channelOptions:uint = 7, grayScale:Boolean = false, offsets:Array = null):void	Sí	No
pixelDissolve(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, randomSeed:int = 0, numPixels:int = 0, fillColor:uint = 0):int	Sí	No
scroll(x:int, y:int):void	Sí	No
setPixel(x:int, y:int, color:uint):void	Sí	Sí
setPixel32(x:int, y:int, color:uint):void	Sí	Sí
setPixels(rect:Rectangle, inputByteArray:ByteArray):void	Sí	Sí
setVector(rect:Rectangle, inputVector:Vector.<uint>):void	Sí	Sí
threshold(sourceBitmapData:BitmapData, sourceRect:Rectangle, destPoint:Point, operation:String, threshold:uint, color:uint = 0, mask:uint = 0xFFFFFFFF, copySource:Boolean = false):uint	Sí	No
unlock(changeRect:Rectangle = null):void	Sí	No

BitmapDataChannel

La siguiente tabla muestra los miembros de la clase BitmapDataChannel que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ALPHA : uint = 8	Sí	Sí
BLUE : uint = 4	Sí	Sí
GREEN : uint = 2	Sí	Sí
RED : uint = 1	Sí	Sí

Boolean

La siguiente tabla muestra los miembros de la clase Boolean que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
Boolean(expression:Object = false)	Sí	Sí
toString() : String	Sí	Sí
valueOf() : Boolean	Sí	Sí

BreakOpportunity

La siguiente tabla muestra los miembros de la clase BreakOpportunity que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ALL : String = "all"	Sí	Sí
ANY : String = "any"	Sí	Sí
AUTO : String = "auto"	Sí	Sí
NONE : String = "none"	Sí	Sí

ByteArray

La siguiente tabla muestra los miembros de la clase ByteArray que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ByteArray()	Sí	Sí
bytesAvailable : uint	Sí	Sí
clear():void	Sí	Sí
compress():void	Sí	No
defaultObjectEncoding : uint	Sí	Sí
deflate():void	Sí	Sí
endian : String	Sí	Sí
inflate():void	Sí	No
length : uint	Sí	Sí
objectEncoding : uint	Sí	Sí
position : uint	Sí	Sí
readBoolean():Boolean	Sí	Sí
readByte():int	Sí	Sí
readBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Sí	Sí
readDouble():Number	Sí	Sí
readFloat():Number	Sí	Sí
readInt():int	Sí	Sí
readMultiByte(length:uint, charSet:String):String	Sí	Sí
readObject():*	Sí	Sí
readShort():int	Sí	Sí
readUnsignedByte():uint	Sí	Sí
readUnsignedInt():uint	Sí	Sí
readUnsignedShort():uint	Sí	Sí
readUTF():String	Sí	Sí
readUTFBytes(length:uint):String	Sí	Sí
toString() : String	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
uncompress():void	Sí	Sí
writeBoolean(value:Boolean):void	Sí	Sí
writeByte(value:int):void	Sí	Sí
writeBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Sí	Sí
writeDouble(value:Number):void	Sí	Sí
writeFloat(value:Number):void	Sí	Sí
writeInt(value:int):void	Sí	Sí
writeMultiByte(value:String, charSet:String):void	Sí	Sí
writeObject(object:*):void	Sí	Sí
writeShort(value:int):void	Sí	Sí
writeUnsignedInt(value:uint):void	Sí	Sí
writeUTF(value:String):void	Sí	Sí
writeUTFBytes(value:String):void	Sí	Sí

Capabilities

La siguiente tabla muestra los miembros de la clase Capabilities que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
avHardwareDisable : Boolean	Sí	Sí
cpuArchitecture	Sí	No
hasAccessibility : Boolean	Sí	Sí
hasAudio : Boolean	Sí	Sí
hasAudioEncoder : Boolean	Sí	Sí
hasEmbeddedVideo : Boolean	Sí	Sí
hasIME : Boolean	Sí	Sí
hasMP3 : Boolean	Sí	Sí
hasPrinting : Boolean	Sí	Sí
hasScreenBroadcast : Boolean	Sí	Sí
hasScreenPlayback : Boolean	Sí	Sí
hasStreamingAudio : Boolean	Sí	Sí
hasStreamingVideo : Boolean	Sí	Sí
hasTLS : Boolean	Sí	Sí
hasVideoEncoder : Boolean	Sí	Sí
isDebugger : Boolean	Sí	Sí
isEmbeddedInAcrobat : Boolean	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
language : String	Sí	Sí
localFileReadDisable : Boolean	Sí	Sí
manufacturer : String	Sí	Sí
maxLevelIDC : String	Sí	Sí
os : String	Sí	Sí
pixelAspectRatio : Number	Sí	Sí
playerType : String	Sí	Sí
screenColor : String	Sí	Sí
screenDPI : Number	Sí	Sí
screenResolutionX : Number	Sí	Sí
screenResolutionY : Number	Sí	Sí
serverString : String	Sí	Sí
supports32BitProcesses : Boolean	Sí	No
supports64BitProcesses : Boolean	Sí	No
touchscreenType : String	Sí	No
version : String	Sí	Sí

CFFHinting

La siguiente tabla muestra los miembros de la clase CFFHinting que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
HORIZONTAL_STEM : String = "horizontalStem"	Sí	Sí
NONE : String = "none"	Sí	Sí

Class (ID de clase)

Para obtener información sobre los miembros de la clase Class que admiten ActionScript 3.0 y Flash Lite 4, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

ColorTransform

La siguiente tabla muestra los miembros de la clase ColorTransform que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
alphaMultiplier: Number	Sí	Sí
alphaOffset: Number	Sí	Sí
blueMultiplier: Number	Sí	Sí
blueOffset: Number	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
color : uint	Sí	Sí
ColorTransform(redMultiplier:Number = 1.0, greenMultiplier:Number = 1.0, blueMultiplier:Number = 1.0, alphaMultiplier:Number = 1.0, redOffset:Number = 0, greenOffset:Number = 0, blueOffset:Number = 0, alphaOffset:Number = 0)	Sí	Sí
concat(second:flash.geom:ColorTransform):void	Sí	Sí
greenMultiplier: Number	Sí	Sí
greenOffset: Number	Sí	Sí
redMultiplier: Number	Sí	Sí
redOffset: Number	Sí	Sí
toString() : String	Sí	Sí

ContentElement

La siguiente tabla muestra los miembros de la clase ContentElement que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ContentElement(elementFormat:ElementFormat = null, eventMirror:EventDispatcher = null, textRotation:String = "rotate0")	Sí	Sí
elementFormat : ElementFormat	Sí	Sí
eventMirror : EventDispatcher	Sí	Sí
GRAPHIC_ELEMENT : uint = 0xFDEF	Sí	Sí
groupElement : GroupElement	Sí	Sí
rawText : String	Sí	Sí
text: String	Sí	Sí
textBlock : flash.text.engine:TextBlock	Sí	Sí
textBlockBeginIndex : int	Sí	Sí
textRotation : String	Sí	Sí
userData : *	Sí	Sí

DataEvent

La siguiente tabla muestra los miembros de la clase DataEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
data : String	Sí	Sí
DATA : String = "data"	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
DataEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, data:String = "")	Sí	Sí
toString() : String	Sí	Sí
UPLOAD_COMPLETE_DATA : String = "uploadCompleteData"	Sí	Sí

Date

La siguiente tabla muestra los miembros de la clase Date que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
date : Number	Sí	Sí
Date(yearOrTimevalue:Object, month:Number, date:Number = 1, hour:Number = 0, minute:Number = 0, second:Number = 0, millisecond:Number = 0)	Sí	Sí
dateUTC : Number	Sí	Sí
day : Number	Sí	Sí
dayUTC : Number	Sí	Sí
fullYear : Number	Sí	Sí
fullYearUTC : Number	Sí	Sí
getDate() : Number	Sí	Sí
getDay() : Number	Sí	Sí
getFullYear() : Number	Sí	Sí
getHours() : Number	Sí	Sí
getMilliseconds() : Number	Sí	Sí
getMinutes() : Number	Sí	Sí
getMonth() : Number	Sí	Sí
getSeconds() : Number	Sí	Sí
getTime() : Number	Sí	Sí
getTimezoneOffset() : Number	Sí	Sí
getUTCDate() : Number	Sí	Sí
getUTCDay() : Number	Sí	Sí
getUTCFullYear() : Number	Sí	Sí
getUTCHours() : Number	Sí	Sí
getUTCMilliseconds() : Number	Sí	Sí
getUTCMinutes() : Number	Sí	Sí
getUTCMonth() : Number	Sí	Sí
getUTCSeconds() : Number	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
hours : Number	Sí	Sí
hoursUTC : Number	Sí	Sí
milliseconds : Number	Sí	Sí
millisecondsUTC : Number	Sí	Sí
minutes : Number	Sí	Sí
minutesUTC : Number	Sí	Sí
month : Number	Sí	Sí
monthUTC : Number	Sí	Sí
parse(date:String):Number	Sí	Sí
seconds : Number	Sí	Sí
secondsUTC : Number	Sí	Sí
setDate(day:Number):Number	Sí	Sí
setFullYear(year:Number, month:Number, day:Number):Number	Sí	Sí
setHours(hour:Number, minute:Number, second:Number, millisecond:Number):Number	Sí	Sí
setMilliseconds(millisecond:Number) : Number	Sí	Sí
setMinutes(minute:Number, second:Number, millisecond:Number):Number	Sí	Sí
setMonth(month:Number, day:Number):Number	Sí	Sí
setSeconds(second:Number, millisecond:Number):Number	Sí	Sí
setTime(millisecond:Number) : Number	Sí	Sí
setUTCDate(day:Number):Number	Sí	Sí
setUTCFullYear(year:Number, month:Number, day:Number):Number	Sí	Sí
setUTCHours(hour:Number, minute:Number, second:Number, millisecond:Number):Number	Sí	Sí
setUTCMilliseconds(millisecond:Number) : Number	Sí	Sí
setUTCMinutes(minute:Number, [second:Number], [millisecond:Number]) : Number	Sí	Sí
setUTCMinutes(minute:Number, second:Number, millisecond:Number):Number	Sí	Sí
setUTCMonth(month:Number, day:Number):Number	Sí	Sí
setUTCSeconds(second:Number, millisecond:Number):Number	Sí	Sí
setYear(year:Number) : Number	Sí	Sí
time : Number	Sí	Sí
timezoneOffset : Number	Sí	Sí
toDateString():String	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
toLocaleDateString():String	Sí	Sí
toLocaleString():String	Sí	Sí
toLocaleTimeString():String	Sí	Sí
toString() : String	Sí	Sí
getTimeString():String	Sí	Sí
toUTCString():String	Sí	Sí
UTC(year:Number, month:Number, date:Number = 1, hour:Number = 0, minute:Number = 0, second:Number = 0, millisecond:Number = 0):Number	Sí	Sí
valueOf() : Number	Sí	Sí

DefinitionError

La siguiente tabla muestra los miembros de la clase DefinitionError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
DefinitionError(message:String = "")	Sí	Sí

Diccionario

La siguiente tabla muestra los miembros de la clase Dictionary que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
Dictionary(weakKeys:Boolean = false)	Sí	Sí

DigitCase

La siguiente tabla muestra los miembros de la clase DigitCase que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
DEFAULT : String = "default"	Sí	Sí
LINING : String = "lining"	Sí	Sí
OLD_STYLE : String = "oldStyle"	Sí	Sí

DigitWidth

La siguiente tabla muestra los miembros de la clase DigitWidth que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
DEFAULT : String = "default"	Sí	Sí
PROPORTIONAL : String = "proportional"	Sí	Sí
TABULAR : String = "tabular"	Sí	Sí

DisplayObject

La siguiente tabla muestra los miembros de la clase DisplayObject que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
accessibilityProperties : AccessibilityProperties	Sí	No
added	Sí	Sí
addedToStage	Sí	Sí
alpha : Number	Sí	Sí
blendMode : String	Sí	No
blendShader : Shader	Sí	No
cacheAsBitmap: Boolean	Sí	Sí
enterFrame	Sí	Sí
exitFrame	Sí	Sí
filters: Array	Sí	No
frameConstructed	Sí	Sí
getBounds(targetCoordinateSpace:DisplayObject):Rectangle	Sí	Sí
getRect(targetCoordinateSpace:DisplayObject):Rectangle	Sí	Sí
globalToLocal(point:Point):Point	Sí	Sí
globalToLocal3D(point:Point):Vector3D	Sí	Sí
height: Number	Sí	Sí
hitTestObject(obj:DisplayObject):Boolean	Sí	Sí
hitTestPoint(x:Number, y:Number, shapeFlag:Boolean = false):Boolean	Sí	Sí
loaderInfo : LoaderInfo	Sí	Sí
local3DToGlobal(point3d:Vector3D):Point	Sí	Sí
localToGlobal(point:Point):Point	Sí	Sí
mask : DisplayObject	Sí	Sí
modified : Boolean	Sí	Sí
mouseX : Number	Sí	Sí
mouseY : Number	Sí	Sí
name: String	Sí	Sí
opaqueBackground : Object	Sí	Sí
parent : DisplayObjectContainer	Sí	Sí
removed	Sí	Sí
removedFromStage	Sí	Sí
render	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
root : DisplayObject	Sí	Sí
rotation : Number	Sí	Sí
rotationX : Number	Sí	Sí
rotationY : Number	Sí	Sí
rotationZ : Number	Sí	Sí
scale9Grid:Rectangle	Sí	Sí
scaleX:Number	Sí	Sí
scaleY:Number	Sí	Sí
scaleZ : Number	Sí	No
scrollRect : Rectangle	Sí	No
stage : Stage	Sí	Sí
transform : flash.geom:Transform	Sí	Sí
visible : Boolean	Sí	Sí
width: Number	Sí	Sí
x: Number	Sí	Sí
y: Number	Sí	Sí
z: Number	Sí	No

DisplayObjectContainer

La siguiente tabla muestra los miembros de la clase DisplayObjectContainer que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
addChild(child:DisplayObject):DisplayObject	Sí	Sí
addChildAt(child:DisplayObject, index:int):DisplayObject	Sí	Sí
areInaccessibleObjectsUnderPoint(point:Point):Boolean	Sí	Sí
contains(child:DisplayObject):Boolean	Sí	Sí
getChildAt(index:int):DisplayObject	Sí	Sí
getChildByName(name:String):DisplayObject	Sí	Sí
getChildIndex(child:DisplayObject):int	Sí	Sí
getObjectsUnderPoint(point:Point):Array	Sí	Sí
mouseChildren : Boolean	Sí	Sí
numChildren : int	Sí	Sí
removeChild(child:DisplayObject):DisplayObject	Sí	Sí
removeChildAt(index:int):DisplayObject	Sí	Sí
setChildIndex(child:DisplayObject, index:int):void	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
swapChildren(child1:DisplayObject, child2:DisplayObject):void	Sí	Sí
swapChildrenAt(index1:int, index2:int):void	Sí	Sí
tabChildren: Boolean	Sí	Sí
textSnapshot : flash.text:TextSnapshot	Sí	No

EastAsianJustifier

La siguiente tabla muestra los miembros de la clase EastAsianJustifier que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():flash.text.engine:TextJustifier	Sí	Sí
EastAsianJustifier(locale:String = "ja", lineJustification:String = "allButLast", justificationStyle:String = "pushInKinsoku")	Sí	Sí
justificationStyle : String	Sí	Sí

ElementFormat

La siguiente tabla muestra los miembros de la clase ElementFormat que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
alignmentBaseline : String	Sí	Sí
alpha : Number	Sí	Sí
baselineShift : Number	Sí	Sí
breakOpportunity : String	Sí	Sí
clone():ElementFormat	Sí	Sí
color : uint	Sí	Sí
digitCase : String	Sí	Sí
digitWidth : String	Sí	Sí
dominantBaseline : String	Sí	Sí
ElementFormat(fontDescription:FontDescription = null, fontSize:Number = 12.0, color:uint = 0x000000, alpha:Number = 1.0, textRotation:String = "auto", dominantBaseline:String = "roman", alignmentBaseline:String = "useDominantBaseline", baselineShift:Number = 0.0, kerning:String = "on", trackingRight:Number = 0.0, trackingLeft:Number = 0.0, locale:String = "en", breakOpportunity:String = "auto", digitCase:String = "default", digitWidth:String = "default", ligatureLevel:String = "common", typographicCase:String = "default")	Sí	Sí
fontDescription : FontDescription	Sí	Sí
fontSize : Number	Sí	Sí
getFontMetrics():FontMetrics	Sí	Sí
kerning : String	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ligatureLevel : String	Sí	Sí
locale : String	Sí	Sí
locked : Boolean	Sí	Sí
textRotation : String	Sí	Sí
trackingLeft : Number	Sí	Sí
trackingRight : Number	Sí	Sí
typographicCase : String	Sí	Sí

Endian

La siguiente tabla muestra los miembros de la clase Endian que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
BIG_ENDIAN : String = "bigEndian"	Sí	Sí
LITTLE_ENDIAN : String = "littleEndian"	Sí	Sí

EOFError

La siguiente tabla muestra los miembros de la clase EOFError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
EOFError(message:String = "")	Sí	Sí

Error

La siguiente tabla muestra los miembros de la clase Error que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
Error(message:String = "", id:int = 0)	Sí	Sí
errorID : int	Sí	Sí
getStackTrace():String	Sí	Sí
message: String	Sí	Sí
name: String	Sí	Sí
toString() : String	Sí	Sí

ErrorEvent

La siguiente tabla muestra los miembros de la clase ErrorEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
ERROR : String = "error"	Sí	Sí
ErrorEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Sí	Sí
errorID : int	Sí	No
toString() : String	Sí	Sí

EvalError

La siguiente tabla muestra los miembros de la clase EvalError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
EvalError(message:String = "")	Sí	Sí

Evento

La siguiente tabla muestra los miembros de la clase Event que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ACTIVATE : String = "activate"	Sí	Sí
ADDED : String = "added"	Sí	Sí
ADDED_TO_STAGE : String = "addedToStage"	Sí	Sí
bubbles : Boolean	Sí	Sí
CANCEL : String = "cancel"	Sí	Sí
cancelable : Boolean	Sí	Sí
CHANGE : String = "change"	Sí	Sí
CLEAR : String = "clear"	Sí	Sí
clone():Event	Sí	Sí
CLOSE : String = "close"	Sí	Sí
COMPLETE : String = "complete"	Sí	Sí
CONNECT : String = "connect"	Sí	Sí
COPY : String = "copy"	Sí	Sí
currentTarget : Object	Sí	Sí
CUT : String = "cut"	Sí	Sí
DEACTIVATE : String = "deactivate"	Sí	Sí
ENTER_FRAME : String = "enterFrame"	Sí	Sí
Event(type:String, bubbles:Boolean = false, cancelable:Boolean = false)	Sí	Sí
eventPhase : uint	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
EXIT_FRAME : String = "exitFrame"	Sí	Sí
formatToString(className:String, ... arguments):String	Sí	Sí
FRAME_CONSTRUCTED : String = "frameConstructed"	Sí	Sí
FULLSCREEN : String = "fullScreen"	Sí	Sí
ID3 : String = "id3"	Sí	Sí
INIT : String = "init"	Sí	Sí
isDefaultPrevented():Boolean	Sí	Sí
MOUSE_LEAVE : String = "mouseLeave"	Sí	Sí
OPEN : String = "open"	Sí	Sí
PASTE : String = "paste"	Sí	Sí
preventDefault():void	Sí	Sí
REMOVED : String = "removed"	Sí	Sí
REMOVED_FROM_STAGE : String = "removedFromStage"	Sí	Sí
RENDER : String = "render"	Sí	Sí
RESIZE : String = "resize"	Sí	Sí
SCROLL : String = "scroll"	Sí	Sí
SELECT : String = "select"	Sí	Sí
SELECT_ALL : String = "selectAll"	Sí	Sí
SOUND_COMPLETE : String = "soundComplete"	Sí	Sí
stopImmediatePropagation():void	Sí	Sí
stopPropagation():void	Sí	Sí
TAB_CHILDREN_CHANGE : String = "tabChildrenChange"	Sí	Sí
TAB_ENABLED_CHANGE : String = "tabEnabledChange"	Sí	Sí
TAB_INDEX_CHANGE : String = "tabIndexChange"	Sí	Sí
target:Object	Sí	Sí
toString() : String	Sí	Sí
type:String	Sí	Sí
UNLOAD : String = "unload"	Sí	Sí

EventDispatcher

La siguiente tabla muestra los miembros de la clase EventDispatcher que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
activate	Sí	Sí
addEventListener(type:String, listener:Function, useCapture:Boolean = false, priority:int = 0, useWeakReference:Boolean = false):void	Sí	Sí
deactivate	Sí	Sí
dispatchEvent(event:Event):Boolean	Sí	Sí
EventDispatcher(target:EventDispatcher = null)	Sí	Sí
hasEventListener(type:String):Boolean	Sí	Sí
removeEventListener(type:String, listener:Function, useCapture:Boolean = false):void	Sí	Sí
willTrigger(type:String):Boolean	Sí	Sí

EventPhase

La siguiente tabla muestra los miembros de la clase EventPhase que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
AT_TARGET : uint = 2	Sí	Sí
BUBBLING_PHASE : uint = 3	Sí	Sí
CAPTURING_PHASE : uint = 1	Sí	Sí

Extensión

La siguiente tabla muestra los miembros de la clase Extension que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
callMethod(methodName:String, ... rest):*	Sí	Sí
Extension(extensionName:String, ... rest)	Sí	Sí

ExternalInterface

La siguiente tabla muestra los miembros de la clase ExternalInterface que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
addCallback(functionName:String, closure:Function):void	Sí	Sí
available : Boolean	Sí	Sí
call(functionName:String, ... arguments):*	Sí	Sí
marshallExceptions : Boolean = false	Sí	Sí
objectID : String	Sí	Sí

FocusDirection

La siguiente tabla muestra los miembros de la clase FocusEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
BOTTOM : String = "bottom"	Sí	No
NONE : String = "none"	Sí	No
TOP : String = "top"	Sí	No

FocusEvent

La siguiente tabla muestra los miembros de la clase FocusEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
FocusEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, relatedObject:InteractiveObject = null, shiftKey:Boolean = false, keyCode:uint = 0)	Sí	Sí
FOCUS_IN : String = "focusIn"	Sí	Sí
FOCUS_OUT : String = "focusOut"	Sí	Sí
isRelatedObjectInaccessible : Boolean	Sí	Sí
keyCode : uint	Sí	Sí
KEY_FOCUS_CHANGE : String = "keyFocusChange"	Sí	Sí
MOUSE_FOCUS_CHANGE : String = "mouseFocusChange"	Sí	Sí
relatedObject : InteractiveObject	Sí	Sí
shiftKey : Boolean	Sí	Sí
toString() : String	Sí	Sí

Font

La siguiente tabla muestra los miembros de la clase Font que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
enumerateFonts(enumerateDeviceFonts:Boolean = false):Array	Sí	Parcial
fontName : String	Sí	Sí
fontStyle : String	Sí	Sí
fontType : String	Sí	Sí
hasGlyphs(str:String):Boolean	Sí	Sí
registerFont(font:Class):void	Sí	Sí

FontDescription

La siguiente tabla muestra los miembros de la clase FontDescription que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
cffHinting : String	Sí	Sí
clone():FontDescription	Sí	Sí
FontDescription(fontName:String = "_serif", fontWeight:String = "normal", fontPosture:String = "normal", fontLookup:String = "device", renderingMode:String = "cff", cffHinting:String = "horizontalStem")	Sí	Sí
fontLookup : String	Sí	Sí
fontName : String	Sí	Sí
fontPosture : String	Sí	Sí
fontWeight : String	Sí	Sí
isFontCompatible(fontName:String, fontWeight:String, fontPosture:String):Boolean	Sí	Sí
relatedObject : InteractiveObject	Sí	Sí
locked : Boolean	Sí	Sí
renderingMode : String	Sí	Sí

FontLookup

La siguiente tabla muestra los miembros de la clase FontLookup que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
DEVICE : String = "device"	Sí	Sí
EMBEDDED_CFF : String = "embeddedCFF"	Sí	Sí

FontMetrics

La siguiente tabla muestra los miembros de la clase FontMetrics que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
emBox : Rectangle	Sí	Sí
FontMetrics(emBox:Rectangle, strikethroughOffset:Number, strikethroughThickness:Number, underlineOffset:Number, underlineThickness:Number, subscriptOffset:Number, subscriptScale:Number, superscriptOffset:Number, superscriptScale:Number)	Sí	Sí
strikethroughOffset : Number	Sí	Sí
strikethroughThickness : Number	Sí	Sí
subscriptOffset : Number	Sí	Sí
subscriptScale : Number	Sí	Sí
superscriptOffset : Number	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
superscriptScale : Number	Sí	Sí
underlineOffset : Number	Sí	Sí
underlineThickness : Number	Sí	Sí

FontPosture

La siguiente tabla muestra los miembros de la clase FontPosture que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ITALIC : String = "italic"	Sí	Sí
NORMAL : String = "normal"	Sí	Sí

FontType

La siguiente tabla muestra los miembros de la clase FontType que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
DEVICE : String = "device"	Sí	Sí
EMBEDDED : String = "embedded"	Sí	Sí
EMBEDDED_CFF : String = "embeddedCFF"	Sí	Sí

FontWeight

La siguiente tabla muestra los miembros de la clase FontWeight que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
BOLD : String = "bold"	Sí	Sí
NORMAL : String = "normal"	Sí	Sí

FrameLabel

La siguiente tabla muestra los miembros de la clase FrameLabel que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
frame : int	Sí	Sí
name: String	Sí	Sí

FullScreenEvent

La siguiente tabla muestra los miembros de la clase FullScreenEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
fullScreen : Boolean	Sí	Sí
FullScreenEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, fullScreen:Boolean = false)	Sí	Sí
FULL_SCREEN : String = "fullScreen"	Sí	Sí
toString() : String	Sí	Sí

Function

La siguiente tabla muestra los miembros de la clase Function que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
apply(thisArg:*, argArray:*):*	Sí	Sí
call(thisArg:*, ... args):*	Sí	Sí

Geolocation

La siguiente tabla muestra los miembros de la clase Geolocation que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
Geolocation()	Sí	Sí
muted : Boolean	Sí	No
update	Sí	Sí
isSupported : Boolean	Sí	Sí
setRequestedUpdateInterval(time:Number):*	Sí	Sí
status	Sí	Sí

GeolocationEvent

La siguiente tabla muestra los miembros de la clase GeolocationEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
altitude : Number	Sí	Sí
GeolocationEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, latitude:Number = 0, longitude:Number = 0, altitude:Number = 0, hAccuracy:Number = 0, vAccuracy:Number = 0, speed:Number = 0, heading:Number = 0, timestamp:int = 0)	Sí	Sí
heading : Number	Sí	Sí
horizontalAccuracy : Number	Sí	Sí
latitude : Number	Sí	Sí
longitude : Number	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
speed : Number	Sí	Sí
timestamp : Number	Sí	Sí
verticalAccuracy : Number	Sí	Sí
clone() : Event	Sí	No
toString() : String	Sí	No
UPDATE : String = "geolocationUpdate"	Sí	Sí

GestureEvent

La siguiente tabla muestra los miembros de la clase GestureEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
altKey : Boolean	Sí	No
clone():Event	Sí	Sí
commandKey : Boolean	Sí	No
controlKey : Boolean	Sí	No
ctrlKey : Boolean	Sí	No
GestureEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, phase:String = null, localX:Number = 0, localY:Number = 0, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false)	Sí	Sí
GESTURE_TWO_FINGER_TAP : String = "gestureTwoFingerTap"	Sí	Sí
localX : Number	Sí	Sí
localY : Number	Sí	Sí
phase : String	Sí	Sí
shiftKey : Boolean	Sí	No
stageX : Number	Sí	Sí
stageY : Number	Sí	Sí
toString() : String	Sí	Sí
updateAfterEvent():void	Sí	Sí

GesturePhase

La siguiente tabla muestra los miembros de la clase GesturePhase que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ALL : String = "all"	Sí	No

Nombre de miembro	ActionScript 3.0	Flash Lite 4
BEGIN : String = "begin"	Sí	Sí
END : String = "end"	Sí	Sí
UPDATE : String = "update"	Sí	Sí

GradientType

La siguiente tabla muestra los miembros de la clase GradientType que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
LINEAR : String = "linear"	Sí	Sí
RADIAL : String = "radial"	Sí	Sí

GraphicElement

La siguiente tabla muestra los miembros de la clase GraphicElement que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
elementHeight : Number	Sí	Sí
elementWidth : Number	Sí	Sí
graphic : DisplayObject	Sí	Sí
GraphicElement(graphic:DisplayObject = null, elementWidth:Number = 15.0, elementHeight:Number = 15.0, elementFormat:ElementFormat = null, eventMirror:EventDispatcher = null, textRotation:String = "rotate0")	Sí	Sí

Graphics

La siguiente tabla muestra los miembros de la clase Graphics que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
beginBitmapFill(bitmap:BitmapData, matrix:Matrix = null, repeat:Boolean = true, smooth:Boolean = false):void	Sí	Sí
beginFill(color:uint, alpha:Number = 1.0):void	Sí	Sí
beginGradientFill(type:String, colors:Array, alphas:Array, ratios:Array, matrix:Matrix = null, spreadMethod:String = "pad", interpolationMethod:String = "rgb", focalPointRatio:Number = 0):void	Sí	Sí
beginShaderFill(shader:Shader, matrix:Matrix = null):void	Sí	No
clear():void	Sí	Sí
copyFrom(sourceGraphics:Graphics):void	Sí	Sí
curveTo(controlX:Number, controlY:Number, anchorX:Number, anchorY:Number):void	Sí	Sí
drawCircle(x:Number, y:Number, radius:Number):void	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
drawEllipse(x:Number, y:Number, width:Number, height:Number):void	Sí	Sí
drawGraphicsData(graphicsData:Vector.<IGraphicsData>):void	Sí	No
drawPath(commands:Vector.<int>, data:Vector.<Number>, winding:String = "evenOdd"):void	Sí	No
drawRect(x:Number, y:Number, width:Number, height:Number):void	Sí	Sí
drawRoundRect(x:Number, y:Number, width:Number, height:Number, ellipseWidth:Number, ellipseHeight:Number = NaN):void	Sí	Sí
drawTriangles(vertices:Vector.<Number>, indices:Vector.<int> = null, uvData:Vector.<Number> = null, culling:String = "none"):void	Sí	No
endFill():void	Sí	Sí
lineBitmapStyle(bitmap:BitmapData, matrix:Matrix = null, repeat:Boolean = true, smooth:Boolean = false):void	Sí	No
lineGradientStyle(type:String, colors:Array, alphas:Array, ratios:Array, matrix:Matrix = null, spreadMethod:String = "pad", interpolationMethod:String = "rgb", focalPointRatio:Number = 0):void	Sí	No
lineShaderStyle(shader:Shader, matrix:Matrix = null):void	Sí	No
lineStyle(thickness:Number = NaN, color:uint = 0, alpha:Number = 1.0, pixelHinting:Boolean = false, scaleMode:String = "normal", caps:String = null, joints:String = null, miterLimit:Number = 3):void	Sí	Parcial
lineTo(x:Number, y:Number):void	Sí	Sí
moveTo(x:Number, y:Number):void	Sí	Sí

GroupElement

La siguiente tabla muestra los miembros de la clase GroupElement que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
elementCount : int	Sí	Sí
getElementAt(index:int):ContentElement	Sí	Sí
getElementAtCharIndex(charIndex:int):ContentElement	Sí	Sí
getElementIndex(element:ContentElement):int	Sí	Sí
GroupElement(elements:Vector.<ContentElement> = null, elementFormat:ElementFormat = null, eventMirror:EventDispatcher = null, textRotation:String = "rotate0")	Sí	Sí
groupElements(beginIndex:int, endIndex:int):GroupElement	Sí	Sí
mergeTextElements(beginIndex:int, endIndex:int):flash.text.engine:TextElement	Sí	Sí
replaceElements(beginIndex:int, endIndex:int, newElements:Vector.<ContentElement>):Vector.<ContentElement>	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
setElements(value:Vector.<ContentElement>):void	Sí	Sí
splitTextElement(elementIndex:int, splitIndex:int):flash.text.engine:TextElement	Sí	Sí
ungroupElements(groupIndex:int):void	Sí	Sí

HTTPStatusEvent

La siguiente tabla muestra los miembros de la clase HTTPStatusEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
HTTPStatusEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, status:int = 0)	Sí	Sí
HTTP_STATUS : String = "httpStatus"	Sí	Sí
status : int	Sí	Sí
toString() : String	Sí	Sí

IBitmapDrawable

Para obtener información sobre la interfaz IBitmapDrawable que admiten ActionScript 3.0 y Flash Lite 4, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

ID3Info

La siguiente tabla muestra los miembros de la clase ID3Info que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
album : String	Sí	Sí
artist : String	Sí	Sí
comment : String	Sí	Sí
genre : String	Sí	Sí
songName : String	Sí	Sí
track : String	Sí	Sí
year : String	Sí	Sí

IDataInput

La siguiente tabla muestra los miembros de la interfaz IDataInput que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
bytesAvailable : uint	Sí	Sí
endian : String	Sí	Sí
objectEncoding : uint	Sí	Sí
readBoolean():Boolean	Sí	Sí
readByte():int	Sí	Sí
readBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Sí	Sí
readDouble():Number	Sí	Sí
readFloat():Number	Sí	Sí
readInt():int	Sí	Sí
readMultiByte(length:uint, charSet:String):String	Sí	Sí
readObject():*	Sí	Sí
readShort():int	Sí	Sí
readUnsignedByte():uint	Sí	Sí
readUnsignedInt():uint	Sí	Sí
readUnsignedShort():uint	Sí	Sí
readUTF():String	Sí	Ye
readUTFBytes(length:uint):String	Sí	Sí

IDataOutput

La siguiente tabla muestra los miembros de la interfaz IDataInput que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
endian : String	Sí	Sí
objectEncoding : uint	Sí	Sí
writeBoolean(value:Boolean):void	Sí	Sí
writeByte(value:int):void	Sí	Sí
writeBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Sí	Sí
writeDouble(value:Number):void	Sí	Sí
writeFloat(value:Number):void	Sí	Sí
writeInt(value:int):void	Sí	Sí
writeMultiByte(value:String, charSet:String):void	Sí	Sí
writeObject(object:*) :void	Sí	Sí
writeShort(value:int):void	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
writeUnsignedInt(value:uint):void	Sí	Sí
writeUTF(value:String):void	Sí	Sí
writeUTFBytes(value:String):void	Sí	Sí

IDynamicPropertyOutput

La siguiente tabla muestra los miembros de la interfaz IDynamicPropertyOutput que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
writeDynamicProperty(name:String, value:*)void	Sí	Sí

IDynamicPropertyWriter

La siguiente tabla muestra los miembros de la interfaz IDynamicPropertyWriter que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
writeDynamicProperties(obj:Object, output:IDynamicPropertyOutput):void	Sí	Sí

IEventDispatcher

La siguiente tabla muestra los miembros de la interfaz IEventDispatcher que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
addEventListener(type:String, listener:Function, useCapture:Boolean = false, priority:int = 0, useWeakReference:Boolean = false):void	Sí	Sí
dispatchEvent(event:Event):Boolean	Sí	Sí
hasEventListener(type:String):Boolean	Sí	Sí
removeEventListener(type:String, listener:Function, useCapture:Boolean = false):void	Sí	Sí
willTrigger(type:String):Boolean	Sí	Sí

IExternalizable

La siguiente tabla muestra los miembros de la interfaz IExternalizable que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
readExternal(input:IDataInput):void	Sí	Sí
writeExternal(output:IDataOutput):void	Sí	Sí

IllegalOperationError

La siguiente tabla muestra los miembros de la clase IllegalOperationError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
IllegalOperationError(message:String = "")	Sí	Sí

IMEEvent

La siguiente tabla muestra los miembros de la clase IMEEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
imeClient : IMEClient	Sí	No
IMEEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Sí	Sí
IME_COMPOSITION : String = "imeComposition"	Sí	Sí
IME_START_COMPOSITION : String = "imeStartComposition"	Sí	No
toString() : String	Sí	Sí

int

La siguiente tabla muestra los miembros de la clase int que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
int(num:Object)	Sí	Sí
MAX_VALUE : int = 2147483648	Sí	Sí
MIN_VALUE : int = -2147483648	Sí	Sí
toExponential(fractionDigits:uint):String	Sí	Sí
toFixed(fractionDigits:uint):String	Sí	Sí
toPrecision(precision:uint):String	Sí	Sí
toString(radix:uint):String	Sí	Sí
valueOf():int	Sí	Sí

InteractiveObject

La siguiente tabla muestra los miembros de la clase InteractiveObject que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clear	Sí	Sí
click	Sí	Sí
contextMenu : ContextMenu	Sí	No

Nombre de miembro	ActionScript 3.0	Flash Lite 4
copy	Sí	No
cut	Sí	No
doubleClick	Sí	No
doubleClickEnabled : Boolean	Sí	No
focusIn	Sí	Sí
focusOut	Sí	Sí
focusRect : Object	Sí	Sí
InteractiveObject()	Sí	Sí
keyDown	Sí	Sí
keyFocusChange	Sí	Sí
keyUp	Sí	Sí
mouseDown	Sí	Sí
mouseEnabled : Boolean	Sí	Sí
mouseFocusChange	Sí	Sí
mouseMove	Sí	Sí
mouseOut	Sí	Sí
mouseOver	Sí	Sí
mouseUp	Sí	Sí
mouseWheel	Sí	No
paste	Sí	No
rollOut	Sí	Sí
rollOver	Sí	Sí
selectAll	Sí	No
tabChildrenChange	Sí	No
tabEnabled: Boolean	Sí	No
tabEnabledChange	Sí	No
tabIndex : int	Sí	No
tabIndexChange	Sí	No

InvalidSWFError

La siguiente tabla muestra los miembros de la clase InvalidSWFError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
InvalidSWFError(message:String = "", id:int = 0)	Sí	Sí

IOError

La siguiente tabla muestra los miembros de la clase IOError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
IOError(message:String = "")	Sí	Sí

IOErrorEvent

La siguiente tabla muestra los miembros de la clase IOErrorEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
IOErrorEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Sí	Sí
IO_ERROR : String = "ioError"	Sí	Sí
toString() : String	Sí	Sí

JPEGLoaderContext

La siguiente tabla muestra los miembros de la clase JPEGLoaderContext que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
deblockingFilter : Number = 0.0	Sí	Sí
JPEGLoaderContext(deblockingFilter:Number = 0.0, checkPolicyFile:Boolean = false, applicationDomain:ApplicationDomain = null, securityDomain:SecurityDomain = null)	Sí	Sí

JustificationStyle

La siguiente tabla muestra los miembros de la clase JustificationStyle que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
PRIORITIZE_LEAST_ADJUSTMENT : String = "prioritizeLeastAdjustment"	Sí	Sí
PUSH_IN_KINSOKU : String = "pushInKinsoku"	Sí	Sí
PUSH_OUT_ONLY : String = "pushOutOnly"	Sí	Sí

Ajuste entre caracteres

La siguiente tabla muestra los miembros de la clase Kerning que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
AUTO : String = "auto"	Sí	Sí
OFF : String = "off"	Sí	Sí
ON : String = "on"	Sí	Sí

Keyboard

La siguiente tabla muestra los miembros de la clase Keyboard que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
BACKSPACE : uint = 8	Sí	Sí
capsLock : Boolean	Sí	Sí
CAPS_LOCK : uint = 20	Sí	Sí
CONTROL : uint = 17	Sí	Sí
DELETE : uint = 46	Sí	Sí
DOWN : uint = 40	Sí	Sí
END : uint = 35	Sí	Sí
ENTER : uint = 13	Sí	Sí
ESCAPE : uint = 27	Sí	Sí
F1 : uint = 112	Sí	Sí
F10 : uint = 121	Sí	Sí
F11 : uint = 122	Sí	Sí
F12 : uint = 123	Sí	Sí
F13 : uint = 124	Sí	Sí
F14 : uint = 125	Sí	Sí
F15 : uint = 126	Sí	Sí
F2 : uint = 113	Sí	Sí
F3 : uint = 114	Sí	Sí
F4 : uint = 115	Sí	Sí
F5 : uint = 116	Sí	Sí
F6 : uint = 117	Sí	Sí
F7 : uint = 118	Sí	Sí
F8 : uint = 119	Sí	Sí
F9 : uint = 120	Sí	Sí
hasVirtualKeyboard : Boolean	Sí	No
HOME : uint = 36	Sí	Sí
INSERT : uint = 45	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
isAccessible() : Boolean	Sí	No
LEFT : uint = 37	Sí	Sí
numLock : Boolean	Sí	Sí
NUMPAD_0 : uint = 96	Sí	Sí
NUMPAD_1 : uint = 97	Sí	Sí
NUMPAD_2 : uint = 98	Sí	Sí
NUMPAD_3 : uint = 99	Sí	Sí
NUMPAD_4 : uint = 100	Sí	Sí
NUMPAD_5 : uint = 101	Sí	Sí
NUMPAD_6 : uint = 102	Sí	Sí
NUMPAD_7 : uint = 103	Sí	Sí
NUMPAD_8 : uint = 104	Sí	Sí
NUMPAD_9 : uint = 105	Sí	Sí
NUMPAD_ADD : uint = 107	Sí	Sí
NUMPAD_DECIMAL : uint = 110	Sí	Sí
NUMPAD_DIVIDE : uint = 111	Sí	Sí
NUMPAD_ENTER : uint = 108	Sí	Sí
NUMPAD_MULTIPLY : uint = 106	Sí	Sí
NUMPAD_SUBTRACT : uint = 109	Sí	Sí
PAGE_DOWN : uint = 34	Sí	Sí
PAGE_UP : uint = 33	Sí	Sí
physicalKeyboardType : String	Sí	No
RIGHT : uint = 39	Sí	Sí
SHIFT : uint = 16	Sí	Sí
SPACE : uint = 32	Sí	Sí
TAB : uint = 9	Sí	Sí
UP : uint = 38	Sí	Sí

KeyboardAutomationAction

La siguiente tabla muestra los miembros de la clase KeyboardAutomationAction que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
KeyboardAutomationAction(type:String, keyCode:uint = 0)	Sí	Sí
IOErrorEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Sí	Sí
IO_ERROR : String = "ioError"	Sí	Sí
toString() : String	Sí	Sí

KeyboardEvent

La siguiente tabla muestra los miembros de la clase KeyboardEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
altKey : Boolean	Sí	No
charCode : uint	Sí	Sí
clone():Event	Sí	Sí
ctrlKey : Boolean	Sí	No
KeyboardEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, charCode:uint = 0, keyCode:uint = 0, keyLocation:uint = 0, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false)	Sí	Sí
keyCode : uint	Sí	Sí
KEY_DOWN : String = "keyDown"	Sí	Sí
keyLocation : uint	Sí	Sí
KEY_UP : String = "keyUp"	Sí	Sí
shiftKey : Boolean	Sí	No
toString() : String	Sí	Sí
updateAfterEvent():void	Sí	No

LigatureLevel

La siguiente tabla muestra los miembros de la clase LigatureLevel que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
COMMON : String = "common"	Sí	Sí
EXOTIC : String = "exotic"	Sí	Sí
MINIMUM : String = "minimum"	Sí	Sí
NONE : String = "none"	Sí	Sí
UNCOMMON : String = "uncommon"	Sí	Sí

LineJustification

La siguiente tabla muestra los miembros de la clase LineJustification que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ALL_BUT_LAST : String = "allButLast"	Sí	Sí
ALL_INCLUDING_LAST : String = "allIncludingLast"	Sí	Sí
UNJUSTIFIED : String = "unjustified"	Sí	Sí

Loader

La siguiente tabla muestra los miembros de la clase Loader que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
close():void	Sí	Sí
content : DisplayObject	Sí	Sí
contentLoaderInfo : LoaderInfo	Sí	Sí
load(request:URLRequest, context:LoaderContext = null):void	Sí	Sí
loadBytes(bytes:ByteArray, context:LoaderContext = null):void	Sí	Sí
Loader()	Sí	Sí
uncaughtErrorEvents : UncaughtErrorEvents	Sí	No
unload():void	Sí	Sí
unloadAndStop(gc:Boolean = true):void	Sí	Sí

LoaderContext

La siguiente tabla muestra los miembros de la clase LoaderContext que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
applicationDomain : ApplicationDomain = null	Sí	Sí
checkPolicyFile : Boolean = false	Sí	Sí
LoaderContext(checkPolicyFile:Boolean = false, applicationDomain:ApplicationDomain = null, securityDomain:SecurityDomain = null)	Sí	Sí
securityDomain : SecurityDomain = null	Sí	Sí

LoaderInfo

La siguiente tabla muestra los miembros de la clase LoaderInfo que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
actionScriptVersion : uint	Sí	Sí
applicationDomain : ApplicationDomain	Sí	Sí
bytes : ByteArray	Sí	Sí
bytesLoaded : uint	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
bytesTotal : uint	Sí	Sí
childAllowsParent : Boolean	Sí	Sí
complete	Sí	Sí
content : DisplayObject	Sí	Sí
contentType: String	Sí	Sí
frameRate : Number	Sí	Sí
getLoaderInfoByDefinition(object:Object):LoaderInfo	Sí	Sí
height : int	Sí	Sí
httpStatus	Sí	Sí
init	Sí	Sí
ioError	Sí	Sí
loader : Loader	Sí	Sí
loaderURL : String	Sí	Sí
open	Sí	Sí
parameters : Object	Sí	Sí
parentAllowsChild : Boolean	Sí	Sí
progress	Sí	Sí
sameDomain : Boolean	Sí	Sí
sharedEvents : EventDispatcher	Sí	Sí
swfVersion : uint	Sí	Sí
uncaughtErrorEvents : UncaughtErrorEvents	Sí	No
unload	Sí	Sí
url: String	Sí	Sí
width : int	Sí	Sí

LocalConnection

La siguiente tabla muestra los miembros de la clase LocalConnection que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
allowDomain(... domains):void	Sí	Sí
allowInsecureDomain(... domains):void	Sí	Sí
asyncError	Sí	Sí
client : Object	Sí	Sí
close():void	Sí	Sí
connect(connectionName:String):void	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
domain : String	Sí	Sí
isPerUser : Boolean	Sí	No
isSupported : Boolean	Sí	No
LocalConnection()	Sí	Sí
securityError	Sí	Sí
send(connectionName:String, methodName:String, ... arguments):void	Sí	Sí
status	Sí	Sí

Math

La siguiente tabla muestra los miembros de la clase Math que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
abs(val:Number):Number	Sí	Sí
acos(val:Number):Number	Sí	Sí
asin(val:Number):Number	Sí	Sí
atan(val:Number):Number	Sí	Sí
atan2(y:Number, x:Number) : Number	Sí	Sí
ceil(val:Number):Number	Sí	Sí
cos(angleRadians:Number):Number	Sí	Sí
E : Number = 2.71828182845905	Sí	Sí
exp(val:Number):Number	Sí	Sí
floor(val:Number):Number	Sí	Sí
LN10 : Number = 2.302585092994046	Sí	Sí
LN2 : Number = 0.6931471805599453	Sí	Sí
log(val:Number):Number	Sí	Sí
LOG10E : Number = 0.4342944819032518	Sí	Sí
LOG2E : Number = 1.442695040888963387	Sí	Sí
max(val1:Number, val2:Number, ... rest):Number	Sí	Sí
min(val1:Number, val2:Number, ... rest):Number	Sí	Sí
PI : Number = 3.141592653589793	Sí	Sí
pow(base:Number, pow:Number):Number	Sí	Sí
random() : Number	Sí	Sí
round(val:Number):Number	Sí	Sí
sin(angleRadians:Number):Number	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
sqrt(val:Number):Number	Sí	Sí
SQRT1_2 : Number = 0.7071067811865476	Sí	Sí
SQRT2 : Number = 1.4142135623730951	Sí	Sí
tan(angleRadians:Number):Number	Sí	Sí

Matrix

La siguiente tabla muestra los miembros de la clase Matrix que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
a: Number	Sí	Sí
b: Number	Sí	Sí
c: Number	Sí	Sí
clone() : Matrix	Sí	Sí
concat(m:Matrix):void	Sí	Sí
createBox(scaleX:Number, scaleY:Number, rotation:Number = 0, tx:Number = 0, ty:Number = 0):void	Sí	Sí
createGradientBox(width:Number, height:Number, [rotation:Number], [tx:Number], [ty:Number]) : Void	Sí	Sí
d: Number	Sí	Sí
deltaTransformPoint(point:Point):Point	Sí	Sí
identity():void	Sí	Sí
invert():void	Sí	Sí
Matrix(a:Number = 1, b:Number = 0, c:Number = 0, d:Number = 1, tx:Number = 0, ty:Number = 0)	Sí	Sí
rotate(angle:Number):void	Sí	Sí
scale(sx:Number, sy:Number):void	Sí	Sí
toString() : String	Sí	Sí
transformPoint(point:Point):Point	Sí	Sí
translate(dx:Number, dy:Number):void	Sí	Sí
tx: Number	Sí	Sí
ty: Number	Sí	Sí

MemoryError

La siguiente tabla muestra los miembros de la clase MemoryError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
MemoryError(message:String = "")	Sí	Sí

MorphShape

Para obtener información sobre los miembros de la clase MorphShape que admiten ActionScript 3.0 y Flash Lite 4, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

MouseAutomationAction

La siguiente tabla muestra los miembros de la clase MouseAutomationAction que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
delta : int	Sí	Sí
MIDDLE_MOUSE_DOWN : String = "middleMouseDown"	Sí	Sí
MIDDLE_MOUSE_UP : String = "middleMouseUp"	Sí	Sí
MouseAutomationAction(type:String, stageX:Number = 0, stageY:Number = 0, delta:int = 0)	Sí	Sí
MOUSE_DOWN : String = "mouseDown"	Sí	Sí
MOUSE_MOVE : String = "mouseMove"	Sí	Sí
MOUSE_UP : String = "mouseUp"	Sí	Sí
MOUSE_WHEEL : String = "mouseWheel"	Sí	Sí
RIGHT_MOUSE_DOWN : String = "rightMouseDown"	Sí	Sí
RIGHT_MOUSE_UP : String = "rightMouseUp"	Sí	Sí
stageX : Number	Sí	Sí
stageY : Number	Sí	Sí

MouseEvent

La siguiente tabla muestra los miembros de la clase MouseEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
altKey : Boolean	Sí	Sí
buttonDown : Boolean	Sí	Sí
CLICK : String = "click"	Sí	Sí
clone():Event	Sí	Sí
ctrlKey : Boolean	Sí	Sí
delta : int	Sí	Sí
DOUBLE_CLICK : String = "doubleClick"	Sí	Sí
isRelatedObjectInaccessible : Boolean	Sí	Sí
localX : Number	Sí	Sí
localY : Number	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
MouseEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, localX:Number = NaN, localY:Number = NaN, relatedObject:InteractiveObject = null, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false, buttonDown:Boolean = false, delta:int = 0)	Sí	Sí
MOUSE_DOWN : String = "mouseDown"	Sí	Sí
MOUSE_MOVE : String = "mouseMove"	Sí	Sí
MOUSE_OUT : String = "mouseOut"	Sí	Sí
MOUSE_OVER : String = "mouseOver"	Sí	Sí
MOUSE_UP : String = "mouseUp"	Sí	Sí
MOUSE_WHEEL : String = "mouseWheel"	Sí	No
relatedObject : InteractiveObject	Sí	Sí
ROLL_OUT : String = "rollOut"	Sí	Sí
ROLL_OVER : String = "rollOver"	Sí	Sí
shiftKey : Boolean	Sí	Sí
stageX : Number	Sí	Sí
stageY : Number	Sí	Sí
toString() : String	Sí	Sí
updateAfterEvent():void	Sí	No

MovieClip

La siguiente tabla muestra los miembros de la clase MovieClip que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
currentFrame : int	Sí	Sí
currentFrameLabel : String	Sí	Sí
currentLabel : String	Sí	Sí
currentLabels : Array	Sí	Sí
currentScene : Scene	Sí	Sí
enabled: Boolean	Sí	Sí
framesLoaded : int	Sí	Sí
gotoAndPlay(frame:Object, scene:String = null):void	Sí	Sí
gotoAndStop(frame:Object, scene:String = null):void	Sí	Sí
MovieClip()	Sí	Sí
nextFrame():void	Sí	Sí
nextScene():void	Sí	Sí
play():void	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
prevFrame():void	Sí	Sí
prevScene():void	Sí	Sí
scenes : Array	Sí	Sí
stop():void	Sí	Sí
totalFrames : int	Sí	Sí
trackAsMenu: Boolean	Sí	No

Multitouch

La siguiente tabla muestra los miembros de la clase Multitouch que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
inputMode : String	Sí	Sí
maxTouchPoints : int	Sí	Sí
supportedGestures : Vector.<String>	Sí	Sí
supportsGestureEvents : Boolean	Sí	Sí
supportsTouchEvent : Boolean	Sí	Sí

MultitouchInputMode

La siguiente tabla muestra los miembros de la clase MultitouchInputMode que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
GESTURE : String = "gesture"	Sí	Sí
NONE : String = "none"	Sí	Sí
TOUCH_POINT : String = "touchPoint"	Sí	Sí

Namespace

La siguiente tabla muestra los miembros de la clase Namespace que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
Namespace(prefixValue:*, uriValue:*)	Sí	Sí
Namespace(uriValue:*)	Sí	Sí
prefix : String	Sí	Sí
toString() : String	Sí	Sí
uri : String	Sí	Sí
valueOf() : String	Sí	Sí

NetConnection

La siguiente tabla muestra los miembros de la clase NetConnection que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
asyncError	Sí	Sí
addHeader(operation:String, mustUnderstand:Boolean = false, param:Object = null):void	Sí	Sí
call(command:String, responder:Responder, ... arguments):void	Sí	Sí
client : Object	Sí	Sí
close():void	Sí	Sí
connect(command:String, ... arguments):void	Sí	Sí
connected : Boolean	Sí	Sí
connectedProxyType : String	Sí	Sí
defaultObjectEncoding : uint	Sí	Sí
farID : String	Sí	No
farNonce : String	Sí	No
ioError	Sí	Sí
maxPeerConnections : uint	Sí	No
nearID : String	Sí	No
nearNonce : String	Sí	No
NetConnection()	Sí	Sí
netStatus	Sí	Sí
objectEncoding : uint	Sí	Sí
protocol : String	Sí	No
proxyType : String	Sí	Sí
securityError	Sí	Sí
unconnectedPeerStreams : Array	Sí	No
uri : String	Sí	Sí
usingTLS : Boolean	Sí	Sí

NetGroup

La siguiente tabla muestra los miembros de la clase NetGroup que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
estimatedMemberCount : Number	Sí	No
info : NetGroupInfo	Sí	No
localCoverageFrom : String	Sí	No

Nombre de miembro	ActionScript 3.0	Flash Lite 4
localCoverageTo : String	Sí	No
neighborCount : Number	Sí	No
receiveMode : String	Sí	No
replicationStrategy : String	Sí	No
NetGroup(connection:NetConnection, groupspec:String)	Sí	No
addHaveObjects(startIndex:Number, endIndex:Number):void	Sí	No
addMemberHint(peerID:String):Boolean	Sí	No
addNeighbor(peerID:String):Boolean	Sí	No
addWantObjects(startIndex:Number, endIndex:Number):void	Sí	No
close():void	Sí	No
convertPeerIDToGroupAddress(peerID:String):String	Sí	No
denyRequestedObject(requestID:int):void	Sí	No
post(message:Object):String	Sí	No
removeHaveObjects(startIndex:Number, endIndex:Number):void	Sí	No
removeWantObjects(startIndex:Number, endIndex:Number):void	Sí	No
sendToAllNeighbors(message:Object):String	Sí	No
sendToNearest(message:Object, groupAddress:String):String	Sí	No
sendToNeighbor(message:Object, sendMode:String):String	Sí	No
writeRequestedObject(requestID:int, object:Object):void	Sí	No

NetGroupInfo

La siguiente tabla muestra los miembros de la clase NetGroupInfo que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
objectReplicationReceiveBytesPerSecond : Number	Sí	No
objectReplicationSendBytesPerSecond : Number	Sí	No
postingReceiveControlBytesPerSecond : Number	Sí	No
postingReceiveDataBytesPerSecond : Number	Sí	No
postingSendControlBytesPerSecond : Number	Sí	No
postingSendDataBytesPerSecond : Number	Sí	No
routingReceiveBytesPerSecond : Number	Sí	No
routingSendBytesPerSecond : Number	Sí	No
toString() : String	Sí	No

NetGroupReceiveMode

La siguiente tabla muestra los miembros de la clase NetGroupReceiveMode que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
EXACT : String = "exact"	Sí	No
NEAREST : String = "nearest"	Sí	No

NetGroupReplicationStrategy

La siguiente tabla muestra los miembros de la clase NetGroupReplicationStrategy que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
LOWEST_FIRST : String = "lowestFirst"	Sí	No
RAREST_FIRST : String = "rarestFirst"	Sí	No

NetGroupSendMode

La siguiente tabla muestra los miembros de la clase NetGroupSendMode que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
NEXT_DECREASING : String = "nextDecreasing"	Sí	No
NEXT_INCREASING : String = "nextIncreasing"	Sí	No

NetGroupSendResult

La siguiente tabla muestra los miembros de la clase NetGroupSendResult que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ERROR : String = "error"	Sí	No
NO_ROUTE : String = "no route"	Sí	No
SENT : String = "sent"	Sí	No

NetStatusEvent

La siguiente tabla muestra los miembros de la clase NetStatusEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
info : Object	Sí	Sí
NetStatusEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, info:Object = null)	Sí	Sí
NET_STATUS : String = "netStatus"	Sí	Sí
toString() : String	Sí	Sí

NetStream

La siguiente tabla muestra los miembros de la clase NetStream que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
asyncError	Sí	Sí
attach(connection:NetConnection) : void	Sí	No
attachAudio	Sí	No
attachCamera	Sí	No
audioReliable : Boolean	Sí	No
audioSampleAccess : Boolean	Sí	No
backBufferLength : Number	Sí	No
backBufferTime : Number	Sí	No
bufferLength : Number	Sí	Sí
bufferTime : Number	Sí	Sí
bufferTimeMax : Number	Sí	No
bytesLoaded : uint	Sí	Sí
bytesTotal : uint	Sí	Sí
checkPolicyFile: Boolean	Sí	No
client : Object	Sí	Sí
close() : Void	Sí	Sí
CONNECT_TO_FMS : String = "connectToFMS"	Sí	Sí
currentFps : Number	Sí	Sí
dataReliable : Boolean	Sí	No
DIRECT_CONNECTIONS : String = "directConnections"	Sí	Sí
farID : String	Sí	No
farNonce : String	Sí	No
inBufferSeek : Boolean	Sí	No
info : NetStreamInfo	Sí	Sí
ioError	Sí	Sí
liveDelay : Number	Sí	Sí
maxPauseBufferTime : Number	Sí	Sí
multicastAvailabilitySendToAll : Boolean	Sí	No
multicastAvailabilityUpdatePeriod : Number	Sí	No
multicastFetchPeriod : Number	Sí	No
multicastInfo : NetStreamMulticastInfo	Sí	No
multicastPushNeighborLimit : Number	Sí	No

Nombre de miembro	ActionScript 3.0	Flash Lite 4
multicastRelayMarginDuration : Number	Sí	No
multicastWindowDuration : Number	Sí	No
nearNonce : String	Sí	Sí
netStatus	Sí	Sí
objectEncoding : uint	Sí	Sí
onPeerConnect(subscriber:NetStream):Boolean	Sí	Sí
onCuePoint	Sí	Sí
onImageData	Sí	Sí
onMetaData	Sí	Sí
onPlayStatus	Sí	Sí
onPlayStatus	Sí	Sí
onXMPData	Sí	Sí
pause():void	Sí	Sí
play(... arguments):void	Sí	Sí
peerStreams : Array	Sí	No
play2(param:NetStreamPlayOptions):void	Sí	Sí
publish(name:String = null, type:String = null):void	Sí	No
receiveAudio(flag:Boolean):void	Sí	Sí
receiveVideo(flag:Boolean):void	Sí	Sí
receiveVideoFPS(FPS:Number):void	Sí	Sí
resume():void	Sí	Sí
seek(offset:Number):void	Sí	Sí
send(handlerName:String, ... arguments):void	Sí	No
soundTransform : flash.media:SoundTransform	Sí	No
step(frames:int):void	Sí	No
time : Number	Sí	Sí
togglePause():void	Sí	Sí
videoReliable : Boolean	Sí	No
videoSampleAccess : Boolean	Sí	No

NetStreamInfo

La siguiente tabla muestra los miembros de la clase NetStreamInfo que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
audioBufferByteLength : Number	Sí	Sí
audioBufferLength : Number	Sí	Sí
audioByteCount : Number	Sí	Sí
audioBytesPerSecond : Number	Sí	Sí
audioLossRate : Number	Sí	Sí
byteCount : Number	Sí	Sí
currentBytesPerSecond : Number	Sí	Sí
dataBufferByteLength : Number	Sí	Sí
dataBufferLength : Number	Sí	Sí
dataByteCount : Number	Sí	Sí
dataBytesPerSecond : Number"	Sí	Sí
droppedFrames : Number	Sí	Sí
NetStreamInfo(curBPS:Number, byteCount:Number, maxBPS:Number, audioBPS:Number, audioByteCount:Number, videoBPS:Number, videoByteCount:Number, dataBPS:Number, dataByteCount:Number, playbackBPS:Number, droppedFrames:Number, audioBufferByteLength:Number, videoBufferByteLength:Number, dataBufferByteLength:Number, audioBufferLength:Number, videoBufferLength:Number, dataBufferLength:Number, srtp:Number, audioLossRate:Number)	Sí	Sí
maxBytesPerSecond : Number	Sí	Sí
playbackBytesPerSecond : Number	Sí	Sí
toString() : String	Sí	Sí
SRTT : Number	Sí	Sí
videoBufferByteLength : Number	Sí	Sí
videoBufferLength : Number	Sí	Sí
videoByteCount : Number	Sí	Sí
videoBytesPerSecond : Number	Sí	Sí
videoLossRate : Number	Sí	No

NetStreamPlayOptions

La siguiente tabla muestra los miembros de la clase NetStreamPlayOptions que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
len:Number	Sí	Sí
NetStreamPlayOptions()	Sí	Sí
oldStreamName : String	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
start:Number	Sí	Sí
streamName : String	Sí	Sí
transition : String	Sí	Sí

NetStreamPlayTransitions

La siguiente tabla muestra los miembros de la clase NetStreamPlayTransitions que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
APPEND : * = append	Sí	Sí
RESET : * = reset	Sí	Sí
STOP : * = stop	Sí	Sí
SWAP : * = swap	Sí	Sí
SWITCH : * = switch	Sí	Sí

Number

La siguiente tabla muestra los miembros de la clase Number que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
MAX_VALUE: Number	Sí	Sí
MIN_VALUE: Number	Sí	Sí
NaN: Number	Sí	Sí
NEGATIVE_INFINITY: Number	Sí	Sí
Number(num:Object)	Sí	Sí
POSITIVE_INFINITY: Number	Sí	Sí
toExponential(fractionDigits:uint):String	Sí	Sí
toFixed(fractionDigits:uint):String	Sí	Sí
toPrecision(precision:uint):String	Sí	Sí
toString(radix:Number) : String	Sí	Sí
valueOf() : Number	Sí	Sí

Object

La siguiente tabla muestra los miembros de la clase Object que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
constructor: Object	Sí	Sí
hasOwnProperty(name:String) : Boolean	Sí	Sí
isPrototypeOf(theClass:Object) : Boolean	Sí	Sí
Object()	Sí	Sí
propertyIsEnumerable(name:String):Boolean	Sí	Sí
prototype: Object	Sí	Sí
prototype:Object	Sí	Sí
setPropertyIsEnumerable(name:String, isEnum:Boolean = true):void	Sí	Sí
toLocaleString():String	Sí	Sí
toString() : String	Sí	Sí
valueOf() : Object	Sí	Sí

ObjectEncoding

La siguiente tabla muestra los miembros de la clase ObjectEncoding que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
AMF0 : uint = 0	Sí	Sí
AMF3 : uint = 3	Sí	Sí
DEFAULT : uint = 3	Sí	Sí
dynamicPropertyWriter : IDynamicPropertyWriter	Sí	Sí

PixelSnapping

La siguiente tabla muestra los miembros de la clase PixelSnapping que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ALWAYS : String = "always"	Sí	Sí
AUTO : String = "auto"	Sí	Sí
NEVER : String = "never"	Sí	Sí

Point

La siguiente tabla muestra los miembros de la clase Point que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
add(v:Point) : Point	Sí	Sí
clone() : Point	Sí	Sí
distance(pt1:Point, pt2:Point) : Number	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
equals(toCompare:Point):Boolean	Sí	Sí
interpolate(pt1:Point, pt2:Point, f:Number) : Point	Sí	Sí
length: Number	Sí	Sí
normalize(thickness:Number):void	Sí	Sí
offset(dx:Number, dy:Number):void	Sí	Sí
Point(x:Number = 0, y:Number = 0)	Sí	Sí
polar(len:Number, angle:Number) : Point	Sí	Sí
subtract(v:Point) : Point	Sí	Sí
toString() : String	Sí	Sí
x: Number	Sí	Sí
y: Number	Sí	Sí

PressAndTapGestureEvent

La siguiente tabla muestra los miembros de la clase PressAndTapGestureEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
GESTURE_PRESS_AND_TAP : String = "gesturePressAndTap"	Sí	Sí
PressAndTapGestureEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, phase:String = null, localX:Number = 0, localY:Number = 0, tapLocalX:Number = 0, tapLocalY:Number = 0, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false)	Sí	Sí
tapLocalX : Number	Sí	Sí
tapLocalY : Number	Sí	Sí
tapStageX : Number	Sí	Sí
tapStageY : Number	Sí	Sí
toString() : String	Sí	Sí

ProgressEvent

La siguiente tabla muestra los miembros de la clase ProgressEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
bytesLoaded : uint	Sí	Sí
bytesTotal : uint	Sí	Sí
clone():Event	Sí	Sí
PROGRESS : String = "progress"	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ProgressEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, bytesLoaded:uint = 0, bytesTotal:uint = 0)	Sí	Sí
SOCKET_DATA : String = "socketData"	Sí	Sí
toString() : String	Sí	Sí

Proxy

La siguiente tabla muestra los miembros de la clase Proxy que admiten ActionScript 3.0 y Flash Lite 4.

Nota: Los miembros incluidos en la siguiente tabla son todos métodos protegidos.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
callProperty(name:*, ... rest):*	Sí	Sí
deleteProperty(name:*) : Boolean	Sí	Sí
getDescendants(name:*) : *	Sí	Sí
getProperty(name:*) : *	Sí	Sí
hasProperty(name:*) : Boolean	Sí	Sí
isAttribute(name:*) : Boolean	Sí	Sí
nextName(index:int) : String	Sí	Sí
nextNameIndex(index:int) : int	Sí	Sí
nextValue(index:int) : *	Sí	Sí
setProperty(name:*, value:*) : void	Sí	Sí
valueOf() : Object	Sí	Sí

QName

La siguiente tabla muestra los miembros de la clase QName que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
localName : String	Sí	Sí
QName(qname:QName)	Sí	Sí
QName(uri:Namespace, localName:QName)	Sí	Sí
toString() : String	Sí	Sí
uri : String	Sí	Sí
valueOf():QName	Sí	Sí

RangeError

La siguiente tabla muestra los miembros de la clase RangeError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
RangeError(message:String = "")	Sí	Sí

Rectángulo

La siguiente tabla muestra los miembros de la clase Rectangle que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
bottom: Number	Sí	Sí
bottomRight: Point	Sí	Sí
clone(): Rectangle	Sí	Sí
contains(x:Number, y:Number): Boolean	Sí	Sí
containsPoint(point:Point):Boolean	Sí	Sí
containsRect(rect:Rectangle):Boolean	Sí	Sí
equals(toCompare:Rectangle):Boolean	Sí	Sí
height: Number	Sí	Sí
inflate(dx:Number, dy:Number):void	Sí	Sí
inflatePoint(point:Point):void	Sí	Sí
intersection(toIntersect:Rectangle): Rectangle	Sí	Sí
intersects(toIntersect:Rectangle): Boolean	Sí	Sí
isEmpty(): Boolean	Sí	Sí
left: Number	Sí	Sí
offset(dx:Number, dy:Number):void	Sí	Sí
offsetPoint(point:Point):void	Sí	Sí
Rectangle(x:Number = 0, y:Number = 0, width:Number = 0, height:Number = 0)	Sí	Sí
right: Number	Sí	Sí
setEmpty():void	Sí	Sí
size: Point	Sí	Sí
top: Number	Sí	Sí
topLeft: Point	Sí	Sí
toString(): String	Sí	Sí
union(toUnion:Rectangle): Rectangle	Sí	Sí
width: Number	Sí	Sí
x: Number	Sí	Sí
y: Number	Sí	Sí

ReferenceError

La siguiente tabla muestra los miembros de la clase ReferenceError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ReferenceError(message:String = "")	Sí	Sí

RegExp

La siguiente tabla muestra los miembros de la clase RegExp que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
dotall : Boolean	Sí	Sí
exec(str:String):Object	Sí	Sí
extended : Boolean	Sí	Sí
global : Boolean	Sí	Sí
ignoreCase : Boolean	Sí	Sí
lastIndex : Number	Sí	Sí
multiline: Boolean	Sí	Sí
RegExp(re:String, flags:String)	Sí	Sí
source : String	Sí	Sí
test(str:String):Boolean	Sí	Sí

RenderingMode

La siguiente tabla muestra los miembros de la clase RenderingMode que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
CFF : String = "cff"	Sí	Sí
NORMAL : String = "normal"	Sí	Sí

Responder

La siguiente tabla muestra los miembros de la clase Responder que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
constructResponderObject(result:Function, status:Function = null):void	Sí	Sí
Responder(result:Function, status:Function = null)	Sí	Sí

Scene

La siguiente tabla muestra los miembros de la clase Scene que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
labels : Array	Sí	Sí
name: String	Sí	Sí
numFrames : int	Sí	Sí

ScriptTimeoutError

La siguiente tabla muestra los miembros de la clase ScriptTimeoutError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ScriptTimeoutError(message:String = "")	Sí	Sí

Security

La siguiente tabla muestra los miembros de la clase Security que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
allowDomain(... domains):void	Sí	Sí
allowInsecureDomain(... domains):void	Sí	Sí
exactSettings: Boolean	Sí	Sí
loadPolicyFile(url:String):void	Sí	Sí
LOCAL_TRUSTED : String = "localTrusted"	Sí	Sí
LOCAL_WITH_FILE : String = "localWithFile"	Sí	Sí
LOCAL_WITH_NETWORK : String = "localWithNetwork"	Sí	Sí
REMOTE : String = "remote"	Sí	Sí
sandboxType : String	Sí	Sí
showSettings(panel:String = "default"):void	Sí	Sí

SecurityDomain

La siguiente tabla muestra los miembros de la clase SecurityDomain que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
currentDomain : SecurityDomain	Sí	Sí

SecurityError

La siguiente tabla muestra los miembros de la clase SecurityError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
SecurityError(message:String = "")	Sí	Sí

SecurityErrorEvent

La siguiente tabla muestra los miembros de la clase SecurityErrorEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
SecurityErrorEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Sí	Sí
SECURITY_ERROR : String = "securityError"	Sí	Sí
toString() : String	Sí	Sí

SecurityPanel

La siguiente tabla muestra los miembros de la clase SecurityPanel que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
CAMERA : String = "camera"	Sí	Sí
DEFAULT : String = "default"	Sí	Sí
DISPLAY : String = "display"	Sí	Sí
LOCAL_STORAGE : String = "localStorage"	Sí	Sí
MICROPHONE : String = "microphone"	Sí	Sí
PRIVACY : String = "privacy"	Sí	Sí
SETTINGS_MANAGER : String = "settingsManager"	Sí	Sí

Shape

La siguiente tabla muestra los miembros de la clase Shape que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
graphics : Graphics	Sí	Sí
Shape()	Sí	Sí

SharedObject

La siguiente tabla muestra los miembros de la clase SharedObject que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
asyncError	Sí	Sí
client : Object	Sí	Sí
clear():void	Sí	Sí
close():void	Sí	Sí
connect(myConnection:NetConnection, params:String = null):void	Sí	Sí
data: Object	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
defaultObjectEncoding : uint	Sí	Sí
flush(minDiskSpace:int = 0):String	Sí	Sí
fps : Number	Sí	Sí
getLocal(name:String, localPath:String = null, secure:Boolean = false):SharedObject	Sí	Sí
getRemote(name:String, remotePath:String = null, persistence:Object = false, secure:Boolean = false):SharedObject	No	Sí
netStatus	Sí	Sí
objectEncoding : uint	Sí	Sí
send(... arguments):void	Sí	Sí
setDirty(propertyName:String):void	Sí	Sí
setProperty(propertyName:String, value:Object = null):void	Sí	Sí
size : uint	Sí	Sí
sync	Sí	Sí

SharedObjectFlushStatus

La siguiente tabla muestra los miembros de la clase SharedObjectFlushStatus que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
FLUSHED : String = "flushed"	Sí	Sí
PENDING : String = "pending"	Sí	Sí

SimpleButton

La siguiente tabla muestra los miembros de la clase SimpleButton que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
downState : DisplayObject	Sí	Sí
enabled: Boolean	Sí	Sí
hitTestState : DisplayObject	Sí	Sí
overState : DisplayObject	Sí	Sí
SimpleButton(upState:DisplayObject = null, overState:DisplayObject = null, downState:DisplayObject = null, hitTestState:DisplayObject = null)	Sí	Sí
soundTransform : flash.media.SoundTransform	Sí	No
trackAsMenu: Boolean	Sí	No
upState : DisplayObject	Sí	Sí
useHandCursor: Boolean	Sí	No

Socket

La siguiente tabla muestra los miembros de la clase Socket que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
bytesAvailable : uint	Sí	Sí
close	Sí	Sí
close():void	Sí	Sí
connect	Sí	Sí
connect(host:String, port:int):void	Sí	Sí
connected : Boolean	Sí	Sí
endian : String	Sí	Sí
flush():void	Sí	Sí
ioError	Sí	Sí
objectEncoding : uint	Sí	Sí
readBoolean():Boolean	Sí	Sí
readByte():int	Sí	Sí
readBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Sí	Sí
readDouble():Number	Sí	Sí
readFloat():Number	Sí	Sí
readInt():int	Sí	Sí
readMultiByte(length:uint, charSet:String):String	Sí	Sí
readObject():*	Sí	Sí
readShort():int	Sí	Sí
readUnsignedByte():uint	Sí	Sí
readUnsignedInt():uint	Sí	Sí
readUnsignedShort():uint	Sí	Sí
readUTF():String	Sí	Sí
readUTFBytes(length:uint):String	Sí	Sí
securityError	Sí	Sí
Socket(host:String = null, port:int = 0)	Sí	Sí
socketData	Sí	Sí
timeout : uint	Sí	Sí
writeBoolean(value:Boolean):void	Sí	Sí
writeByte(value:int):void	Sí	Sí
writeBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
writeDouble(value:Number):void	Sí	Sí
writeFloat(value:Number):void	Sí	Sí
writeInt(value:int):void	Sí	Sí
writeMultiByte(value:String, charSet:String):void	Sí	Sí
writeObject(object:*):void	Sí	Sí
writeShort(value:int):void	Sí	Sí
writeUnsignedInt(value:uint):void	Sí	Sí
writeUTF(value:String):void	Sí	Sí
writeUTFBytes(value:String):void	Sí	Sí

Sonido

La siguiente tabla muestra los miembros de la clase Sound que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
bytesLoaded : uint	Sí	Sí
bytesTotal : int	Sí	Sí
close():void	Sí	Sí
complete	Sí	Sí
extract(target:ByteArray, length:Number, startPosition:Number = -1):Number	Sí	No
id3 : Event	Sí	Sí
id3 : ID3Info	Parcial	Parcial
isBuffering : Boolean	Sí	Sí
ioError	Sí	Sí
length: Number	Sí	Sí
load(stream:URLRequest, context:SoundLoaderContext = null):void	Sí	Sí
open	Sí	Sí
play(startTime:Number = 0, loops:int = 0, sndTransform:flash.media:SoundTransform = null):SoundChannel	Sí	Sí
progress	Sí	Sí
sampleData	Sí	No
Sound(stream:URLRequest = null, context:SoundLoaderContext = null)	Sí	Sí
url: String	Sí	Sí

SoundChannel

La siguiente tabla muestra los miembros de la clase SoundChannel que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
leftPeak : Number	Sí	Sí
position : Number	Sí	Sí
rightPeak : Number	Sí	Sí
soundComplete	Sí	Sí
soundTransform : flash.media:SoundTransform	Sí	Sí
stop():void	Sí	Sí

SoundLoaderContext

La siguiente tabla muestra los miembros de la clase SoundLoaderContext que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
bufferTime : Number = 1000	Sí	Sí
checkPolicyFile : Boolean = false	Sí	Sí
SoundLoaderContext(bufferTime:Number = 1000, checkPolicyFile:Boolean = false)	Sí	Sí

SoundMixer

La siguiente tabla muestra los miembros de la clase SoundMixer que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
areSoundsInaccessible():Boolean	Sí	No
bufferTime : int	Sí	Sí
computeSpectrum(outputArray:ByteArray, FFTMode:Boolean = false, stretchFactor:int = 0):void	Sí	No
stopAll():void	Sí	Sí

SoundTransform

La siguiente tabla muestra los miembros de la clase SoundTransform que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
leftToLeft : Number	Sí	Sí
leftToRight : Number	Sí	Sí
pan : Number	Sí	Sí
rightToLeft : Number	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
rightToLeft : Number	Sí	Sí
SoundTransform(vol:Number = 1, panning:Number = 0)	Sí	Sí
volume : Number	Sí	Sí

SpaceJustifier

La siguiente tabla muestra los miembros de la clase SpaceJustifier que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():flash.text.engine.TextJustifier	Sí	Sí
letterSpacing : Boolean	Sí	Sí
maximumSpacing : Number	Sí	No
minimumSpacing : Number	Sí	No
optimumSpacing : Number	Sí	No
SpaceJustifier(locale:String = "en", lineJustification:String = "unjustified", letterSpacing:Boolean = false)	Sí	Sí
volume : Number	Sí	Sí

Sprite

La siguiente tabla muestra los miembros de la clase Sprite que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
buttonMode : Boolean	Sí	Sí
dropTarget : DisplayObject	Sí	Sí
graphics : Graphics	Sí	Sí
hitArea : Sprite	Sí	Sí
soundTransform : flash.media.SoundTransform	Sí	Sí
Sprite()	Sí	Sí
startDrag(lockCenter:Boolean = false, bounds:Rectangle = null):void	Sí	Sí
startTouchDrag(touchPointID:int, lockCenter:Boolean = false, bounds:Rectangle = null):void	Sí	Sí
stopDrag():void	Sí	Sí
stopTouchDrag(touchPointID:int):void	Sí	Sí
useHandCursor : Boolean	Sí	No

StackOverflowError

La siguiente tabla muestra los miembros de la clase StackOverflowError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
StackOverflowError(message:String = "")	Sí	Sí

Stage

La siguiente tabla muestra los miembros de la clase Stage que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
addChild(child:DisplayObject):DisplayObject	Sí	Sí
addChildAt(child:DisplayObject, index:int):DisplayObject	Sí	Sí
addEventListener(type:String, listener:Function, useCapture:Boolean = false, priority:int = 0, useWeakReference:Boolean = false):void	Sí	Sí
align: String	Sí	Sí
assignFocus(objectToFocus:InteractiveObject, direction:String):void	Sí	No
colorCorrection : String		No
colorCorrectionSupport : String		No
dispatchEvent(event:Event):Boolean	Sí	Sí
displayState: String	Sí	Sí
focus : InteractiveObject	Sí	Sí
frameRate : Number	Sí	Sí
fullScreen	Sí	Sí
fullScreenHeight : uint	Sí	Sí
fullScreenSourceRect : Rectangle	Sí	Sí
fullScreenWidth : uint	Sí	Sí
hasEventListener(type:String):Boolean	Sí	Sí
height: Number	Sí	Sí
isFocusInaccessible():Boolean	Sí	Sí
mouseChildren : Boolean	Sí	Sí
mouseLeave	Sí	Sí
numChildren : int	Sí	Sí
quality : String	Sí	Sí
removeChildAt(index:int):DisplayObject	Sí	Sí
resize	Sí	Sí
scaleMode: String	Sí	Sí
setChildIndex(child:DisplayObject, index:int):void	Sí	Sí
showDefaultContextMenu : Boolean	Sí	No

Nombre de miembro	ActionScript 3.0	Flash Lite 4
stageFocusRect : Boolean	Sí	Sí
stageHeight : int	Sí	Sí
stageWidth : int	Sí	Sí
swapChildrenAt(index1:int, index2:int):void	Sí	Sí
tabChildren: Boolean	Sí	No
textSnapshot : flash.text:TextSnapshot	Sí	No
width: Number	Sí	Sí
willTrigger(type:String):Boolean	Sí	No

StageAlign

La siguiente tabla muestra los miembros de la clase StageAlign que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
BOTTOM : String = "B"	Sí	Sí
BOTTOM_LEFT : String = "BL"	Sí	Sí
BOTTOM_RIGHT : String = "BR"	Sí	Sí
LEFT : String = "L"	Sí	Sí
RIGHT : String = "R"	Sí	Sí
TOP : String = "T"	Sí	Sí
TOP_LEFT : String = "TL"	Sí	Sí
TOP_RIGHT : String = "TR"	Sí	Sí

StageCapture

La siguiente tabla muestra los miembros de la clase StageCapture que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
addEventListener(type:String, listener:Function, useCapture:Boolean = false, priority:int = 0, useWeakReference:Boolean = false):void	Sí	Sí
cancel():void	Sí	Sí
capture	Sí	Sí
capture(type:String):void	Sí	Sí
clipRect : Rectangle	Sí	Sí
CURRENT : String = "current"	Sí	Sí
fileNameBase : String	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
MULTIPLE : String = "multiple"	Sí	Sí
NEXT : String = "next"	Sí	Sí
removeEventListener(type:String, listener:Function, useCapture:Boolean = false):void	Sí	Sí

StageCaptureEvent

La siguiente tabla muestra los miembros de la clase StageCaptureEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
CAPTURE : String = "capture"	Sí	Sí
checksum : uint	Sí	Sí
StageCaptureEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, url:String = null, checksum:uint = 0)	Sí	Sí
url: String	Sí	Sí

StageDisplayState

La siguiente tabla muestra los miembros de la clase StageDisplayState que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
FULL_SCREEN : String = "fullScreen"	Sí	Sí
NORMAL : String = "normal"	Sí	Sí

StageQuality

La siguiente tabla muestra los miembros de la clase StageQuality que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
BEST : String = "best"	Sí	Sí
HIGH : String = "high"	Sí	Sí
LOW : String = "low"	Sí	Sí
MEDIUM : String = "medium"	Sí	Sí

StageScaleMode

La siguiente tabla muestra los miembros de la clase StageScaleMode que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
EXACT_FIT : String = "exactFit"	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
NO_BORDER : String = "noBorder"	Sí	Sí
NO_SCALE : String = "noScale"	Sí	Sí
SHOW_ALL : String = "showAll"	Sí	Sí

StaticText

La siguiente tabla muestra los miembros de la clase StaticText que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
text: String	Sí	Sí

StatusEvent

La siguiente tabla muestra los miembros de la clase StatusEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
code : String	Sí	Sí
level : String	Sí	Sí
STATUS : String = "status"	Sí	Sí
StatusEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, code:String = "", level:String = "")	Sí	Sí
toString() : String	Sí	Sí

String

La siguiente tabla muestra los miembros de la clase String que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
charAt(index:Number = 0):String	Sí	Sí
charCodeAt(index:Number = 0):Number	Sí	Sí
concat(... args):String	Sí	Sí
fromCharCode(... charCodes):String	Sí	Sí
indexOf(val:String, startIndex:Number = 0):int	Sí	Sí
lastIndexOf(val:String, startIndex:Number = 0x7FFFFFFF):int	Sí	Sí
length : int	Sí	Sí
localeCompare(other:String, ... values):int	Sí	Sí
match(pattern:*):Array	Sí	Sí
replace(pattern:*, repl:Object):String	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
search(pattern:*) : int	Sí	Sí
slice(startIndex:Number = 0, endIndex:Number = 0x7fffffff):String	Sí	Sí
split(delimiter:*, limit:Number = 0x7fffffff):Array	Sí	Sí
String(val:String)	Sí	Sí
substr(startIndex:Number = 0, len:Number = 0x7fffffff):String	Sí	Sí
substring(startIndex:Number = 0, endIndex:Number = 0x7fffffff):String	Sí	Sí
toLocaleLowerCase():String	Sí	Sí
toLocaleUpperCase():String	Sí	Sí
toLowerCase() : String	Sí	Sí
toUpperCase() : String	Sí	Sí
valueOf() : String	Sí	Sí

StyleSheet

La siguiente tabla muestra los miembros de la clase StyleSheet que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clear():void	Sí	Sí
getStyle(styleName:String):Object	Sí	Sí
parseCSS(CSSText:String):void	Sí	Sí
setStyle(styleName:String, styleObject:Object):void	Sí	Sí
styleNames : Array	Sí	Sí
transform(formatObject:Object):flash.text:TextFormat	Sí	Sí

SWFVersion

La siguiente tabla muestra los miembros de la clase SWFVersion que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
FLASH1 : uint = 1	Sí	Sí
FLASH10 : uint = 10	Sí	Sí
FLASH2 : uint = 2	Sí	Sí
FLASH3 : uint = 3	Sí	Sí
FLASH4 : uint = 4	Sí	Sí
FLASH5 : uint = 5	Sí	Sí
FLASH6 : uint = 6	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
FLASH7 : uint = 7	Sí	Sí
FLASH8 : uint = 8	Sí	Sí
FLASH9 : uint = 9	Sí	Sí

SyncEvent

La siguiente tabla muestra los miembros de la clase SyncEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
changeList : Array	Sí	Sí
clone():Event	Sí	Sí
SYNC : String = "sync"	Sí	Sí
SyncEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, changeList:Array = null)	Sí	Sí
toString() : String	Sí	Sí

SyntaxError

La siguiente tabla muestra los miembros de la clase SyntaxError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
SyntaxError(message:String = "")	Sí	Sí

System

La siguiente tabla muestra los miembros de la clase System que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
disposeXML	Sí	No
exit(code:uint):void	Sí	No
freeMemory : Number	Sí	No
gc():void	Sí	Sí
ime : IME	Sí	Sí
pause():void	Sí	Sí
privateMemory : Number	Sí	No
resume():void	Sí	Sí
setClipboard(string:String):void	Sí	No

Nombre de miembro	ActionScript 3.0	Flash Lite 4
totalMemory : uint	Sí	Sí
totalMemoryNumber : Number	Sí	No
useCodePage : Boolean	Sí	Sí

TabAlignment

La siguiente tabla muestra los miembros de la clase TabAlignment que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
CENTER : String = "center"	Sí	Sí
DECIMAL : String = "decimal"	Sí	Sí
END : String = "end"	Sí	Sí
START : String = "start"	Sí	Sí

TabStop

La siguiente tabla muestra los miembros de la clase TabStop que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
alignment : String	Sí	Sí
decimalAlignmentToken : String	Sí	Sí
position : Number	Sí	Sí
TabStop(alignment:String = "start", position:Number = 0.0, decimalAlignmentToken:String = "")	Sí	Sí

TextBaseline

La siguiente tabla muestra los miembros de la clase TextBaseline que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ASCENT : String = "ascent"	Sí	Sí
DESCENT : String = "descent"	Sí	Sí
IDEOGRAPHIC_BOTTOM : String = "ideographicBottom"	Sí	Sí
IDEOGRAPHIC_CENTER : String = "ideographicCenter"	Sí	Sí
IDEOGRAPHIC_TOP : String = "ideographicTop"	Sí	Sí
ROMAN : String = "roman"	Sí	Sí
USE_DOMINANT_BASELINE : String = "useDominantBaseline"	Sí	Sí

TextBlock

La siguiente tabla muestra los miembros de la clase TextBlock que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
applyNonLinearFontScaling : Boolean	Sí	Sí
baselineFontDescription : FontDescription	Sí	Sí
baselineFontSize : Number	Sí	Sí
baselineZero : String	Sí	Sí
bidilLevel : int	Sí	Sí
content : ContentElement	Sí	Sí
createTextLine(previousLine:flash.text.engine:TextLine = null, width:Number = 1000000, lineOffset:Number = 0.0, fitSomething:Boolean = false):flash.text.engine:TextLine	Sí	Sí
dump():String	Sí	No
findNextAtomBoundary(afterCharIndex:int):int	Sí	Sí
findNextWordBoundary(afterCharIndex:int):int	Sí	Sí
findPreviousAtomBoundary(beforeCharIndex:int):int	Sí	Sí
findPreviousWordBoundary(beforeCharIndex:int):int	Sí	Sí
firstInvalidLine : flash.text.engine:TextLine	Sí	Sí
firstLine : flash.text.engine:TextLine	Sí	Sí
getTextLineAtCharIndex(charIndex:int):flash.text.engine:TextLine	Sí	Sí
lastLine : flash.text.engine:TextLine	Sí	Sí
lineRotation : String	Sí	Sí
recreateTextLine(textLine:flash.text.engine:TextLine, previousLine:flash.text.engine:TextLine = null, width:Number = 1000000, lineOffset:Number = 0.0, fitSomething:Boolean = false):flash.text.engine:TextLine	Sí	Sí
releaseLineCreationData():void	Sí	Sí
releaseLines(firstLine:flash.text.engine:TextLine, lastLine:flash.text.engine:TextLine):void	Sí	Sí
tabStops : Vector.<flash.text.engine:TabStop>	Sí	Sí
TextBlock(content:ContentElement = null, tabStops:Vector.<flash.text.engine:TabStop> = null, textJustifier:flash.text.engine:TextJustifier = null, lineRotation:String = "rotate0", baselineZero:String = "roman", bidilLevel:int = 0, applyNonLinearFontScaling:Boolean = true, baselineFontDescription:FontDescription = null, baselineFontSize:Number = 12.0)	Sí	Sí
textJustifier : flash.text.engine:TextJustifier	Sí	Sí
textLineCreationResult : String	Sí	Sí
userData : *	Sí	Sí

TextElement

La siguiente tabla muestra los miembros de la clase TextElement que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
replaceText(beginIndex:int, endIndex:int, newText:String):void	Sí	Sí
text: String	Sí	Sí
TextElement(text:String = null, elementFormat:ElementFormat = null, eventMirror:EventDispatcher = null, textRotation:String = "rotate0")	Sí	Sí

TextEvent

La siguiente tabla muestra los miembros de la clase TextEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
LINK : String = "link"	Sí	Sí
text: String	Sí	Sí
TextEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false, text:String = "")	Sí	Sí
TEXT_INPUT : String = "textInput"	Sí	Sí
toString() : String	Sí	Sí

TextField

La siguiente tabla muestra los miembros de la clase TextField que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
alwaysShowSelection : Boolean	Sí	Sí
antiAliasType: String	Sí	No
appendText(newText:String):void	Sí	Sí
autoSize : String	Sí	Sí
background: Boolean	Sí	Sí
backgroundColor : uint	Sí	Sí
border: Boolean	Sí	Sí
borderColor : uint	Sí	Sí
bottomScrollV : int	Sí	Sí
caretIndex : int	Sí	Sí
change	Sí	Sí
condenseWhite: Boolean	Sí	Sí
defaultTextFormat : flash.text:TextFormat	Sí	Sí
displayAsPassword : Boolean	Sí	Sí
embedFonts: Boolean	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
getCharBoundaries(charIndex:int):Rectangle	Sí	Sí
getCharIndexAtPoint(x:Number, y:Number):int	Sí	Sí
getFirstCharInParagraph(charIndex:int):int	Sí	Sí
getImageReference(id:String):DisplayObject	Sí	Sí
getLineIndexAtPoint(x:Number, y:Number):int	Sí	Sí
getLineIndexOfChar(charIndex:int):int	Sí	Sí
getLineLength(lineIndex:int):int	Sí	Sí
getLineMetrics(lineIndex:int):flash.text:TextLineMetrics	Sí	Sí
getLineOffset(lineIndex:int):int	Sí	Sí
getLineText(lineIndex:int):String	Sí	Sí
getParagraphLength(charIndex:int):int	Sí	Sí
getTextFormat(beginIndex:int = -1, endIndex:int = -1):flash.text:TextFormat	Sí	Sí
gridFitType: String	Sí	No
htmlText: String	Sí	Sí
isFontCompatible(fontName:String, fontStyle:String):Boolean	Sí	Sí
length : int	Sí	Sí
link	Sí	Sí
maxChars : int	Sí	Sí
maxScrollH : int	Sí	Sí
maxScrollV : int	Sí	Sí
mouseWheelEnabled: Boolean	Sí	No
multiline: Boolean	Sí	Sí
numLines : int	Sí	Sí
replaceSelectedText(value:String):void	Sí	Sí
replaceText(beginIndex:int, endIndex:int, newText:String):void	Sí	Sí
restrict: String	Sí	Sí
scroll	Sí	Sí
scrollH : int	Sí	Sí
scrollV : int	Sí	Sí
selectable: Boolean	Sí	Sí
selectionBeginIndex : int	Sí	Sí
selectionEndIndex : int	Sí	Sí
setSelection(beginIndex:int, endIndex:int):void	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
setTextFormat(format:flash.text:TextFormat, beginIndex:int = -1, endIndex:int = -1):void	Sí	Sí
sharpness: Number	Sí	No
styleSheet: StyleSheet	Sí	Sí
text: String	Sí	Sí
textColor : uint	Sí	Sí
TextField()	Sí	Sí
textHeight: Number	Sí	Sí
textInput	Sí	Sí
textWidth: Number	Sí	Sí
thickness: Number	Sí	No
type: String	Sí	Sí
useRichTextClipboard : Boolean	Sí	No
wordWrap: Boolean	Sí	Sí

TextFieldAutoSize

La siguiente tabla muestra los miembros de la clase TextFieldAutoSize que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
CENTER : String = "center"	Sí	Sí
LEFT : String = "left"	Sí	Sí
NONE : String = "none"	Sí	Sí
RIGHT : String = "right"	Sí	Sí

TextFieldType

La siguiente tabla muestra los miembros de la clase TextFieldType que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
DYNAMIC : String = "dynamic"	Sí	Sí
INPUT : String = "input"	Sí	Sí

TextFormat

La siguiente tabla muestra los miembros de la clase TextFormat que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
align: String	Sí	Sí
blockIndent: Object	Sí	Sí
bold: Object	Sí	Sí
bullet: Object	Sí	Sí
color: Object	Sí	Sí
font: String	Sí	Sí
indent: Object	Sí	Sí
italic: Object	Sí	Sí
kerning: Object	Sí	No
leading: Object	Sí	Sí
leftMargin: Object	Sí	Sí
letterSpacing: Object	Sí	No
rightMargin: Object	Sí	Sí
size: Object	Sí	Sí
tabStops: Array	Sí	Sí
target: String	Sí	Sí
TextFormat(font:String = null, size:Object = null, color:Object = null, bold:Object = null, italic:Object = null, underline:Object = null, url:String = null, target:String = null, align:String = null, leftMargin:Object = null, rightMargin:Object = null, indent:Object = null, leading:Object = null)	Sí	Sí
underline: Object	Sí	Sí
url: String	Sí	Sí

TextFormatAlign

La siguiente tabla muestra los miembros de la clase TextFormatAlign que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
CENTER: String = "center"	Sí	Sí
JUSTIFY: String = "justify"	Sí	Sí
LEFT: String = "left" NONE: String = "none"	Sí	Sí
RIGHT: String = "right"	Sí	Sí

TextJustifier

La siguiente tabla muestra los miembros de la clase TextJustifier que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():flash.text.engine:TextJustifier	Sí	Sí
getJustifierForLocale(locale:String):flash.text.engine:TextJustifier	Sí	Sí
lineJustification : String	Sí	Sí
locale : String	Sí	Sí
TextJustifier(locale:String, lineJustification:String)	Sí	Sí

TextLine

La siguiente tabla muestra los miembros de la clase TextLine que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ascent : Number	Sí	Sí
atomCount : int	Sí	Sí
descent : Number	Sí	Sí
dump():String	Sí	No
flushAtomData():void	Sí	Sí
getAtomBidiLevel(atomIndex:int):int	Sí	Sí
getAtomBounds(atomIndex:int):Rectangle	Sí	Sí
getAtomCenter(atomIndex:int):Number	Sí	Sí
getAtomGraphic(atomIndex:int):DisplayObject	Sí	Sí
getAtomIndexAtCharIndex(charIndex:int):int	Sí	Sí
getAtomIndexAtPoint(stageX:Number, stageY:Number):int	Sí	Sí
getAtomTextBlockBeginIndex(atomIndex:int):int	Sí	Sí
getAtomTextBlockEndIndex(atomIndex:int):int	Sí	Sí
getAtomTextRotation(atomIndex:int):String	Sí	Sí
getAtomWordBoundaryOnLeft(atomIndex:int):Boolean	Sí	Sí
getBaselinePosition(baseline:String):Number	Sí	Sí
getMirrorRegion(mirror:EventDispatcher):flash.text.engine:TextLineMirrorRegion	Sí	Sí
hasGraphicElement : Boolean	Sí	Sí
hasTabs : Boolean	Sí	No
MAX_LINE_WIDTH : int = 1000000	Sí	Sí
mirrorRegions : Vector.<flash.text.engine:TextLineMirrorRegion>	Sí	Sí
nextLine : flash.text.engine:TextLine	Sí	Sí
previousLine : flash.text.engine:TextLine	Sí	Sí
rawTextLength : int	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
specifiedWidth : Number	Sí	Sí
textBlock : flash.text.engine:TextBlock	Sí	Sí
textBlockBeginIndex : int	Sí	Sí
textHeight: Number	Sí	Sí
textWidth: Number	Sí	Sí
unjustifiedTextWidth : Number	Sí	Sí
userData : *	Sí	Sí
validity : String	Sí	Sí

TextLineCreationResult

La siguiente tabla muestra los miembros de la clase TextLineCreationResult que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
COMPLETE : String = "complete"	Sí	Sí
EMERGENCY : String = "emergency"	Sí	Sí
INSUFFICIENT_WIDTH : String = "insufficientWidth"	Sí	Sí
SUCCESS : String = "success"	Sí	Sí

TextLineMetrics

La siguiente tabla muestra los miembros de la clase TextLineMetrics que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ascent : Number	Sí	Sí
descent : Number	Sí	Sí
height: Number	Sí	Sí
leading: Number	Sí	Sí
TextLineMetrics(x:Number, width:Number, height:Number, ascent:Number, descent:Number, leading:Number)	Sí	Sí
width: Number	Sí	Sí
x: Number	Sí	Sí

TextLineMirrorRegion

La siguiente tabla muestra los miembros de la clase TextLineMirrorRegion que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
bounds : Rectangle	Sí	Sí
element : ContentElement	Sí	Sí
mirror : EventDispatcher	Sí	Sí
nextRegion : flash.text.engine:TextLineMirrorRegion	Sí	Sí
previousRegion : flash.text.engine:TextLineMirrorRegion	Sí	Sí
textLine : flash.text.engine:TextLine	Sí	Sí

TextLineValidity

La siguiente tabla muestra los miembros de la clase TextLineValidity que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
INVALID : String = "invalid"	Sí	Sí
POSSIBLY_INVALID : String = "possiblyInvalid"	Sí	Sí
STATIC : String = "static"	Sí	Sí
VALID : String = "valid"	Sí	Sí

TextRotation

La siguiente tabla muestra los miembros de la clase TextRotation que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
AUTO : String = "auto"	Sí	Sí
ROTATE_0 : String = "rotate0"	Sí	Sí
ROTATE_180 : String = "rotate180"	Sí	Sí
ROTATE_270 : String = "rotate270"	Sí	Sí
ROTATE_90 : String = "rotate90"	Sí	Sí

TextSnapshot

La siguiente tabla muestra los miembros de la clase TextSnapshot que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
charCount : int	Sí	Sí
findText(beginIndex:int, textToFind:String, caseSensitive:Boolean):int	Sí	Sí
getSelected(beginIndex:int, endIndex:int):Boolean	Sí	Sí
getSelectedText(includeLineEndings:Boolean = false):String	Sí	Sí
getText(beginIndex:int, endIndex:int, includeLineEndings:Boolean = false):String	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
getTextRunInfo(beginIndex:int, endIndex:int):Array	Sí	Sí
hitTestTextNearPos(x:Number, y:Number, maxDistance:Number = 0):Number	Sí	Sí
setSelectColor(hexColor:uint = 0xFFFF00):void	Sí	Sí
setSelected(beginIndex:int, endIndex:int, select:Boolean):void	Sí	Sí

Timer

La siguiente tabla muestra los miembros de la clase Timer que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
currentCount : int	Sí	Sí
delay : Number	Sí	Sí
repeatCount : int	Sí	Sí
reset():void	Sí	Sí
running : Boolean	Sí	Sí
start():void	Sí	Sí
stop():void	Sí	Sí
timer	Sí	Sí
Timer(delay:Number, repeatCount:int = 0)	Sí	Sí
timerComplete	Sí	Sí

TimerEvent

La siguiente tabla muestra los miembros de la clase TimerEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
TIMER : String = "timer"	Sí	Sí
TimerEvent(type:String, bubbles:Boolean = false, cancelable:Boolean = false)	Sí	Sí
TIMER_COMPLETE : String = "timerComplete"	Sí	Sí
toString() : String	Sí	Sí
updateAfterEvent():void	Sí	Sí

TouchEvent

La siguiente tabla muestra los miembros de la clase TouchEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
altKey : Boolean	Sí	No
clone():Event	Sí	Sí
ctrlKey : Boolean	Sí	No
isPrimaryTouchPoint : Boolean	Sí	Sí
isRelatedObjectInaccessible : Boolean	Sí	Sí
localX : Number	Sí	Sí
localY : Number	Sí	Sí
pressure : Number	Sí	Sí
relatedObject : InteractiveObject	Sí	Sí
shiftKey : Boolean	Sí	No
sizeX : Number	Sí	Sí
sizeY : Number	Sí	Sí
stageX : Number	Sí	Sí
stageY : Number	Sí	Sí
toString() : String	Sí	Sí
TouchEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, touchPointID:int = 0, isPrimaryTouchPoint:Boolean = false, localX:Number = NaN, localY:Number = NaN, sizeX:Number = NaN, sizeY:Number = NaN, pressure:Number = NaN, relatedObject:InteractiveObject = null, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false)	Sí	Sí
touchPointID : int	Sí	Sí
TOUCH_BEGIN : String = "touchBegin"	Sí	Sí
TOUCH_END : String = "touchEnd"	Sí	Sí
TOUCH_MOVE : String = "touchMove"	Sí	Sí
TOUCH_OUT : String = "touchOut"	Sí	Sí
TOUCH_OVER : String = "touchOver"	Sí	Sí
TOUCH_ROLL_OUT : String = "touchRollOut"	Sí	Sí
TOUCH_ROLL_OVER : String = "touchRollOver"	Sí	Sí
TOUCH_TAP : String = "touchTap"	Sí	Sí
updateAfterEvent():void	Sí	Sí

Transform

La siguiente tabla muestra los miembros de la clase Transform que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
colorTransform : flash.geom:ColorTransform	Sí	Sí
concatenatedColorTransform : flash.geom:ColorTransform	Sí	Sí
concatenatedMatrix : Matrix	Sí	Sí
getRelativeMatrix3D(relativeTo:DisplayObject):Matrix3D	Sí	No
matrix: Matrix	Sí	Sí
matrix3D : Matrix3D	Sí	No
perspectiveProjection : PerspectiveProjection	Sí	No
pixelBounds:Rectangle	Sí	Sí

TransformGestureEvent

La siguiente tabla muestra los miembros de la clase TransformGestureEvent que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clone():Event	Sí	Sí
GESTURE_PAN : String = "gesturePan"	Sí	Sí
GESTURE_ROTATE : String = "gestureRotate"	Sí	Sí
GESTURE_SWIPE : String = "gestureSwipe"	Sí	Sí
GESTURE_ZOOM : String = "gestureZoom"	Sí	Sí
offsetX : Number	Sí	Sí
offsetY : Number	Sí	Sí
rotation : Number	Sí	Sí
scaleX:Number	Sí	Sí
scaleY:Number	Sí	Sí
TransformGestureEvent(type:String, bubbles:Boolean = true, cancelable:Boolean = false, phase:String = null, localX:Number = 0, localY:Number = 0, scaleX:Number = 1.0, scaleY:Number = 1.0, rotation:Number = 0, offsetX:Number = 0, offsetY:Number = 0, ctrlKey:Boolean = false, altKey:Boolean = false, shiftKey:Boolean = false)	Sí	Sí
toString() : String	Sí	Sí

TypeError

La siguiente tabla muestra los miembros de la clase TypeError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
TypeError(message:String = "")	Sí	Sí

TypographicCase

La siguiente tabla muestra los miembros de la clase `TypographicCase` que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
<code>CAPS : String = "caps"</code>	Sí	Sí
<code>CAPS_AND_SMALL_CAPS : String = "capsAndSmallCaps"</code>	Sí	Sí
<code>DEFAULT : String = "default"</code>	Sí	Sí
<code>LOWERCASE : String = "lowercase"</code>	Sí	Sí
<code>SMALL_CAPS : String = "smallCaps"</code>	Sí	Sí
<code>TITLE : String = "title"</code>	Sí	Sí
<code>UPPERCASE : String = "uppercase"</code>	Sí	Sí

uint

La siguiente tabla muestra los miembros de la clase `uint` que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
<code>MAX_VALUE : uint = 4294967295</code>	Sí	Sí
<code>MIN_VALUE : uint = 0</code>	Sí	Sí
<code>uint(num:Object)</code>	Sí	Sí
<code>toExponential(fractionDigits:uint):String</code>	Sí	Sí
<code>toFixed(fractionDigits:uint):String</code>	Sí	Sí
<code>toPrecision(precision:uint):String</code>	Sí	Sí
<code>toString(radix:uint):String</code>	Sí	Sí
<code>valueOf():uint</code>	Sí	Sí

URIError

La siguiente tabla muestra los miembros de la clase `URIError` que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
<code>URIError(message:String = "")</code>	Sí	Sí

URLLoader

La siguiente tabla muestra los miembros de la clase `URLLoader` que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
<code>bytesLoaded : uint = 0</code>	Sí	Sí
<code>bytesTotal : uint = 0</code>	Sí	Sí
<code>close():void</code>	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
complete	Sí	Sí
data : *	Sí	Sí
dataFormat : String = "text"	Sí	Sí
httpStatus	Sí	Sí
ioError	Sí	Sí
load(request:URLRequest):void	Sí	Sí
open	Sí	Sí
progress	Sí	Sí
securityError	Sí	Sí

URLLoaderDataFormat

La siguiente tabla muestra los miembros de la clase URLLoaderDataFormat que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
BINARY : String = "binary"	Sí	Sí
TEXT : String = "text"	Sí	Sí
VARIABLES : String = "variables"	Sí	Sí

URLRequest

La siguiente tabla muestra los miembros de la clase URLRequest que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
authenticate : Boolean	Sí	No
cacheResponse : Boolean	Sí	No
contentType: String	Sí	Sí
data: Object	Sí	Sí
digest : String	Sí	Sí
manageCookies : Boolean	Sí	No
method:String	Sí	Sí
requestHeaders : Array	Sí	Sí
url: String	Sí	Sí
URLRequest(url:String = null)	Sí	Sí
useCache : Boolean	Sí	No

URLRequestHeader

La siguiente tabla muestra los miembros de la clase URLRequestHeader que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
name: String	Sí	Sí
URLRequestHeader(name:String = "", value:String = "")	Sí	Sí
value: String	Sí	Sí

URLRequestMethod

La siguiente tabla muestra los miembros de la clase URLRequestMethod que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
DELETE : String = "DELETE"	Sí	No
GET : String = "GET"	Sí	Sí
HEAD : String = "HEAD"	Sí	No
OPTIONS : String = "OPTIONS"	Sí	No
POST : String = "POST"	Sí	Sí
PUT : String = "PUT"	Sí	No

URLStream

La siguiente tabla muestra los miembros de la clase URLStream que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
bytesAvailable : uint	Sí	Sí
close():void	Sí	Sí
complete	Sí	Sí
connected : Boolean	Sí	Sí
endian : String	Sí	Sí
httpStatus	Sí	Sí
ioError	Sí	Sí
load(request:URLRequest):void	Sí	Sí
objectEncoding : uint	Sí	Sí
open	Sí	Sí
progress	Sí	Sí
readBoolean():Boolean	Sí	Sí
readByte():int	Sí	Sí
readBytes(bytes:ByteArray, offset:uint = 0, length:uint = 0):void	Sí	Sí
readDouble():Number	Sí	Sí
readFloat():Number	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
readInt():int	Sí	Sí
readMultiByte(length:uint, charSet:String):String	Sí	Sí
readObject():*	Sí	Sí
readShort():int	Sí	Sí
readUnsignedByte():uint	Sí	Sí
readUnsignedInt():uint	Sí	Sí
readUnsignedShort():uint	Sí	Sí
readUTF():String	Sí	Sí
readUTFBytes(length:uint):String	Sí	Sí
securityError	Sí	Sí

URLVariables

La siguiente tabla muestra los miembros de la clase URLVariables que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
decode(source:String):void	Sí	Sí
toString() : String	Sí	Sí
URLVariables(source:String = null)	Sí	Sí

Vector

La siguiente tabla muestra los miembros de la clase Vector que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
concat(... args):Vector.<T>	Sí	Sí
every(callback:Function, thisObject:Object = null):Boolean	Sí	Sí
filter(callback:Function, thisObject:Object = null):Vector.<T>	Sí	Sí
fixed : Boolean	Sí	Sí
forEach(callback:Function, thisObject:Object = null):void	Sí	Sí
indexOf(searchElement:T, fromIndex:int = 0):int	Sí	Sí
join(sep:String = ","):String	Sí	Sí
lastIndexOf(searchElement:T, fromIndex:int = 0x7fffffff):int	Sí	Sí
length : uint	Sí	Sí
map(callback:Function, thisObject:Object = null):Vector.<T>	Sí	Sí
pop():T	Sí	Sí
push(... args):uint	Sí	Sí
reverse():Vector.<T>	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
shift():T	Sí	Sí
slice(startIndex:int = 0, endIndex:int = 16777215):Vector.<T>	Sí	Sí
some(callback:Function, thisObject:Object = null):Boolean	No	Sí
sort(compareFunction:Function):Vector.<T>	Sí	Sí
splice(startIndex:int, deleteCount:uint = 4294967295, ... items):Vector.<T>	Sí	Sí
toLocaleString():String	No	Sí
toString() : String	No	Sí
unshift(... args):uint	No	Sí

VerifyError

La siguiente tabla muestra los miembros de la clase VerifyError que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
VerifyError(message:String = "")	Sí	Sí

Video

La siguiente tabla muestra los miembros de la clase Video que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
attachCamera(camera:Camera):void	Sí	No
attachNetStream(netStream:NetStream):void	Sí	Sí
clear():void	Sí	Sí
constructVideoObject(width:int = 320, height:int = 240):void	Sí	Sí
deblocking : int	Sí	Sí
smoothing: Boolean	Sí	Sí
Video(width:int = 320, height:int = 240)	Sí	Sí
videoHeight : int	Sí	Sí
videoWidth : int	Sí	Sí

XML

La siguiente tabla muestra los miembros de la clase XML que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
addNamespace(ns:Object):XML	Sí	Sí
appendChild(child:Object):XML	Sí	Sí
attribute(attributeName*):XMLList	Sí	Sí
attributes():XMLList	Sí	Sí
child(propertyName:Object):XMLList	Sí	Sí
childIndex():int	Sí	Sí
children():XMLList	Sí	Sí
comments():XMLList	Sí	Sí
contains(value:XML):Boolean	Sí	Sí
copy():XML	Sí	Sí
defaultSettings():Object	Sí	
descendants(name:Object = *):XMLList	Sí	Sí
elements(name:Object = *):XMLList	Sí	Sí
hasComplexContent():Boolean	Sí	Sí
hasOwnProperty(p:String):Boolean	Sí	Sí
hasSimpleContent():Boolean	Sí	Sí
ignoreComments : Boolean	Sí	Sí
ignoreProcessingInstructions : Boolean	Sí	Sí
ignoreWhitespace : Boolean	Sí	Sí
inScopeNamespaces():Array	Sí	Sí
insertChildAfter(child1:Object, child2:Object):*	Sí	Sí
insertChildBefore(child1:Object, child2:Object):*	Sí	Sí
length():int	Sí	Sí
localName():Object	Sí	Sí
name():Object	Sí	Sí
namespace(prefix:String = null):*	Sí	Sí
namespaceDeclarations():Array	Sí	Sí
nodeKind():String	Sí	Sí
normalize():XML	Sí	Sí
parent():*	Sí	Sí
prependChild(value:Object):XML	Sí	Sí
prettyIndent : int	Sí	Sí
prettyPrinting : Boolean	Sí	Sí
processingInstructions(name:String = "*"):XMLList	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
propertyIsEnumerable(p:String):Boolean	Sí	Sí
removeNamespace(ns:Namespace):XML	Sí	Sí
replace(propertyName:Object, value:XML):XML	Sí	Sí
setChildren(value:Object):XML	Sí	Sí
setLocalName(name:String):void	Sí	Sí
setName(name:String):void	Sí	Sí
setNamespace(ns:Namespace):void	Sí	Sí
setSettings(... rest):void	Sí	Sí
settings():Object	Sí	Sí
text():XMLList	Sí	Sí
toString() : String	Sí	Sí
toXMLString():String	Sí	Sí
valueOf():XML	Sí	Sí

XMLDocument

La siguiente tabla muestra los miembros de la clase XMLDocument que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
createElement(name:String) : XMLNode	Sí	Sí
createTextNode(text:String):XMLNode	Sí	Sí
docTypeDecl : Object = null	Sí	Sí
idMap: Object	Sí	Sí
ignoreWhite : Boolean = false	Sí	Sí
parseXML(source:String):void	Sí	Sí
toString() : String	Sí	Sí
xmlDecl : Object = null	Sí	Sí
XMLDocument(source:String = null)	Sí	Sí

XMLList

La siguiente tabla muestra los miembros de la clase XMLList que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
attribute(attributeName:*):XMLList	Sí	Sí
attributes():XMLList	Sí	Sí
child(propertyName:Object):XMLList	Sí	Sí
children():XMLList	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
comments():XMLList	Sí	Sí
comments():XMLList	Sí	Sí
contains(value:XML):Boolean	Sí	Sí
copy():XML	Sí	Sí
descendants(name:Object = *):XMLList	Sí	Sí
elements(name:Object = *):XMLList	Sí	Sí
hasComplexContent():Boolean	Sí	Sí
hasOwnProperty(p:String):Boolean	Sí	Sí
hasSimpleContent():Boolean	Sí	Sí
length():int	Sí	Sí
normalize():XMLList	Sí	Sí
parent():Object	Sí	Sí
processingInstructions(name:String = "*"):XMLList	Sí	Sí
propertyIsEnumerable(p:String):Boolean	Sí	Sí
text():XMLList	Sí	Sí
toString() : String	Sí	Sí
toXMLString():String	Sí	Sí
valueOf():XMLList	Sí	Sí

XMLNode

La siguiente tabla muestra los miembros de la clase XMLNode que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
appendChild(node:XMLNode):void	Sí	Sí
attributes: Object	Sí	Sí
childNodes : Array	Sí	Sí
cloneNode(deep:Boolean) : XMLNode	Sí	Sí
firstChild : XMLNode	Sí	Sí
getNamespaceForPrefix(prefix:String) : String	Sí	Sí
getPrefixForNamespace(ns:String):String	Sí	Sí
hasChildNodes() : Boolean	Sí	Sí
insertBefore(node:XMLNode, before:XMLNode):void	Sí	Sí
lastChild : XMLNode	Sí	Sí
localName : String	Sí	Sí
namespaceURI : String	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
nextSibling : XMLNode	Sí	Sí
nodeName: String	Sí	Sí
nodeType : uint	Sí	Sí
nodeValue: String	Sí	Sí
parentNode : XMLNode	Sí	Sí
prefix : String	Sí	Sí
previousSibling : XMLNode	Sí	Sí
removeNode():void	Sí	Sí
toString() : String	Sí	Sí
XMLNode(type:uint, value:String)	Sí	Sí

XMLNodeType

La siguiente tabla muestra los miembros de la clase XMLNodeType que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
ELEMENT_NODE : uint = 1	Sí	Sí
TEXT_NODE : uint = 3	Sí	Sí

XMLSocket

La siguiente tabla muestra los miembros de la clase XMLSocket que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
close	Sí	Sí
close():void	Sí	Sí
connect	Sí	Sí
connect(host:String, port:int):void	Sí	Sí
connected : Boolean	Sí	Sí
data	Sí	Sí
ioError	Sí	Sí
securityError	Sí	Sí
send(object:*) : void	Sí	Sí
timeout : int	Sí	Sí
XMLSocket(host:String = null, port:int = 0)	Sí	Sí

Funciones globales

flash.net

La siguiente tabla muestra las funciones globales del paquete flash.net que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
getClassByAlias(aliasName:String):Class	Sí	Sí
navigateToURL(request:URLRequest, window:String = null):void	Sí	Sí
registerClassAlias(aliasName:String, classObject:Class):void	Sí	Sí
sendToURL(request:URLRequest):void	Sí	Sí

flash.utils

La siguiente tabla muestra las funciones globales del paquete flash.utils que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
clearInterval(id:uint):void	Sí	Sí
clearTimeout(id:uint):void	Sí	Sí
describeType(value:*):XML	Sí	Sí
escapeMultiByte(value:String):String	Sí	Sí
getAliasName(value:*):String	Sí	Sí
getDefinitionByName(name:String):Object	Sí	Sí
getQualifiedClassName(value:*):String	Sí	Sí
getQualifiedSuperclassName(value:*):String	Sí	Sí
getTimer():int	Sí	Sí
setInterval(closure:Function, delay:Number, ... arguments):uint	Sí	Sí
setTimeout(closure:Function, delay:Number, ... arguments):uint	Sí	Sí
unescapeMultiByte(value:String):String	Sí	Sí

Nivel superior

La siguiente tabla muestra las funciones globales de nivel superior que admiten ActionScript 3.0 y Flash Lite 4.

Nombre de miembro	ActionScript 3.0	Flash Lite 4
Array(... args):Array	Sí	Sí
Boolean(expression:Object) : Boolean	Sí	Sí
decodeURI(uri:String):String	Sí	Sí
decodeURIComponent(uri:String):String	Sí	Sí
encodeURIComponent(uri:String):String	Sí	Sí

Nombre de miembro	ActionScript 3.0	Flash Lite 4
encodeURIComponent(uri:String):String	Sí	Sí
escape(str:String):String	Sí	Sí
int(value:Number):int	Sí	Sí
isFinite(num:Number):Boolean	Sí	Sí
isNaN(num:Number):Boolean	Sí	Sí
isXMLName(str:String):Boolean	Sí	Sí
Number(expression:Object):Number	Sí	Sí
Object(value:Object):Object	Sí	Sí
parseFloat(str:String):Number	Sí	Sí
parseInt(str:String, radix:uint = 0):Number	Sí	Sí
String(expression:Object):String	Sí	Sí
trace(... arguments):void	Sí	Sí
uint(value:Number):uint	Sí	Sí
unescape(str:String):String	Sí	Sí
Vector(sourceArray:Object):Vector.<T>	Sí	Sí
XML(expression:Object):XML	Sí	Sí
XMLList(expression:Object):XMLList	Sí	Sí

Clases de ActionScript admitidas parcialmente: detalles

Existen algunas clases de ActionScript 3.0 admitidas parcialmente o que no son compatibles en Flash Lite 4. Para obtener información sobre estas clases, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

Clase BitmapData

La clase BitmapData permite trabajar con datos (píxeles) de un objeto Bitmap. Los métodos de la clase BitmapData se pueden utilizar para crear imagen de mapa de bits opacas o transparentes de tamaño arbitrario. La clase BitmapData también se puede utilizar para manipular estas imágenes de distintos modos en tiempo de ejecución. Asimismo, es posible acceder a BitmapData para una imagen de mapa de bits que se carga con la clase flash.display.Loader.

Resumen de métodos

La siguiente tabla incluye los métodos de la clase BitMapData que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
applyFilter()	Toma una imagen de origen y un objeto de filtro y genera la imagen filtrada.	No se admite
generateFilterRect()	Determina el rectángulo de destino al que afecta la llamada applyFilter(), dados un objeto BitmapData, un rectángulo de origen y un objeto de filtro.	No se admite
lock()	Bloquea una imagen, por lo que los objetos que hagan referencia al objeto BitmapData como, por ejemplo, Bitmap, no se actualizan cuando este objeto BitmapData cambia.	No se admite
noise()	Rellena una imagen con píxeles que representan ruido aleatorio.	No se admite
perlinNoise()	Genera una imagen de ruido Perlin.	No se admite
pixelDissolve()	Realiza una disolución de píxeles de una imagen de origen a una imagen de destino o utilizando la misma imagen.	No se admite
scroll()	Desplaza una imagen una determinada cantidad de píxeles (x, y).	No se admite
threshold()	Prueba valores de píxel en una imagen con un umbral especificado y define los píxeles que se pasan para probar los valores del nuevo color.	No se admite
unlock()	Desbloquea una imagen, por lo que los objetos que hagan referencia al objeto BitmapData como, por ejemplo, Bitmap, se actualizan cuando este objeto BitmapData cambia.	No se admite

Clase ByteArray

La clase ByteArray proporciona métodos y propiedades para optimizar la lectura, escritura y trabajo con datos binarios.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase ByteArray que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
compress	Comprime el conjunto de bytes utilizando el formato de datos comprimidos zlib.	No se admite
inflate	Descomprime el conjunto de bytes.	No se admite

Clase Capabilities

La clase Capabilities proporciona métodos y propiedades para optimizar la lectura, escritura y trabajo con datos binarios.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase Capabilities que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
supports32BitProcesses	Especifica si el sistema admite procesos de 32 bits en ejecución.	No se admite
supports64BitProcesses	Especifica si el sistema admite procesos de 64 bits en ejecución.	No se admite

Clase DisplayObject

La clase DisplayObject es la clase base para todos los objetos que se pueden situar en la lista de visualización. La lista de visualización administra todos los objetos mostrados en Flash Player. Utilice la clase DisplayObjectContainer para organizar los objetos en la lista de visualización.

Los objetos DisplayObjectContainer pueden tener objetos de visualización secundarios, mientras otros objetos de visualización, como Shape y TextField, son nodos "hoja" que sólo cuentan con nodos principales y relacionados, no secundarios.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase DisplayObject que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
accessibilityProperties	Opciones de accesibilidad actuales para este objeto de visualización.	No se admite
blendMode	Valor de la clase BlendMode que especifica qué modo de mezcla utilizar.	No se admite
blendShader	Establece un sombreado que se utiliza para mezclar el fondo y el primer plano.	No se admite
filters	Conjunto indexado que contiene cada objeto de filtro asociado actualmente con el objeto de visualización.	No se admite
scaleZ	Indica la escala de profundidad (porcentaje) de un objeto aplicada desde el punto de registro del objeto.	No se admite
scrollRect	Límites del rectángulo de desplazamiento del objeto de visualización.	No se admite
z	Indica la posición de coordenada z junto con el eje z de la instancia de DisplayObject relativa al contenedor principal 3D.	No se admite

Clase DisplayObjectContainer

La clase DisplayObjectContainer es la clase base para todos los objetos que pueden servir como contenedores de objetos de visualización en la lista de visualización. La lista de visualización administra todos los objetos mostrados en Flash Player.

Utilice la clase DisplayObjectContainer para organizar los objetos en la lista de visualización. Cada objeto DisplayObjectContainer tiene su propia lista secundaria para organizar el orden z de los objetos. El orden z es el orden de delante a atrás que determina qué objeto se dibuja en la parte de delante, cuál se sitúa detrás, etc.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase DisplayObjectContainer que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
textSnapshot	Devuelve un objeto TextSnapshot para esta instancia de DisplayObjectContainer.	No se admite

Clase Font

La clase Font se utiliza para administrar las fuentes incorporadas en los archivos SWF. Las fuentes incorporadas se representan como subclase de la clase Font. La clase Font resulta útil únicamente para buscar información sobre fuentes incorporadas, de modo que no es posible modificar una fuente utilizando esta clase.

La clase Font no se puede utilizar para cargar fuentes externas ni para crear una instancia de un objeto Font por sí misma. Utilice la clase Font como una clase base abstracta.

Resumen de métodos

La siguiente tabla incluye los métodos de la clase Font que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
enumerateFonts()	Especifica si proporcionar una lista de las fuentes incorporadas disponibles actualmente.	Se admite de forma parcial

Clase Graphics

La clase Graphics contiene un conjunto de métodos que se pueden utilizar para crear una forma vectorial. Los objetos de visualización que admiten el dibujo incluyen objetos Sprite y Shape. Cada una de estas clases incluye una propiedad de gráficos que es un objeto Graphics. Las siguientes funciones se encuentran entre las auxiliares proporcionadas para facilidad de uso: drawRect(), drawRoundRect(), drawCircle() y drawEllipse().

Resumen de métodos

La siguiente tabla incluye los métodos de la clase Graphics que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
beginShaderFill()	Especifica un relleno de sombreado que utiliza Flash Player para posteriores llamadas a otros métodos Graphics (por ejemplo, lineTo() o drawCircle()) para el objeto.	No se admite
drawGraphicsData()	Envía una serie de instancias de IGraphicsData para el dibujo.	No se admite
drawPath()	Envía una serie de comandos para el dibujo.	No se admite
drawTriangles()	Representa un conjunto de triángulos, generalmente para distorsionar mapas de bits y proporcionarles un aspecto tridimensional.	No se admite
lineBitmapStyle()	Especifica un mapa de bits para utilizar en el trazo de línea al dibujar líneas.	No se admite
lineGradientStyle()	Especifica un degradado para utilizar en el trazo al dibujar líneas.	No se admite
lineShaderStyle()	Especifica un sombreado para utilizar en el trazo de línea al dibujar líneas.	No se admite
lineStyle()	Especifica un estilo de línea utilizado para posteriores llamadas a los métodos Graphics como, por ejemplo, el método lineTo() o drawCircle().	Se admite de forma parcial

Clase InteractiveObject

La clase InteractiveObject es la clase base abstracta para todos los objetos de visualización con los que puede interactuar el usuario, utilizando el ratón y el teclado.

No se puede crear una instancia de la clase InteractiveObject directamente. Una llamada al nuevo constructor `InteractiveObject()` genera una excepción `ArgumentError`.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase InteractiveObject que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
contextMenu	Especifica el menú contextual asociado a este objeto.	No se admite
doubleClickEnabled	Especifica si el objeto recibe eventos doubleClick.	No se admite
tabEnabled	Especifica si el objeto está en el orden de tabulación.	No se admite
tabIndex	Especifica el orden de tabulación de los objetos de un archivo SWF.	No se admite

Resumen de eventos

La siguiente tabla incluye los eventos de la clase InteractiveObject que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Evento	Descripción	Compatibilidad
copy	Se distribuye cuando el usuario activa la combinación de teclas de aceleración específica de la plataforma para una operación de copia o selecciona 'Copiar' en el menú contextual de texto.	No se admite
cut	Se distribuye cuando el usuario activa la combinación de teclas de aceleración específica de la plataforma para una operación de corte o selecciona 'Cortar' en el menú contextual de texto.	No se admite
doubleClick	Se distribuye cuando un usuario presiona y suelta el botón principal de un dispositivo de señalización dos veces en sucesión rápida sobre el mismo objeto InteractiveObject cuando el indicador doubleClickEnabled de ese objeto se establece en true.	No se admite
mouseWheel	Se distribuye cuando se hace girar una rueda del ratón sobre una instancia de InteractiveObject en la ventana de Flash Player.	No se admite
paste	Se distribuye cuando el usuario activa la combinación de teclas de aceleración específica de la plataforma para una operación de pegado o selecciona 'Pegar' en el menú contextual de texto.	No se admite
selectAll	Se distribuye cuando el usuario activa la combinación de teclas de aceleración específica de la plataforma para una operación de seleccionar todo o elige 'Seleccionar todo' en el menú contextual de texto.	No se admite

Evento	Descripción	Compatibilidad
tabChildrenChange	Se distribuye cuando cambia el valor del indicador tabChildren del objeto.	No se admite
tabEnabledChange	Se distribuye cuando cambia el indicador tabEnabled del objeto.	No se admite
tabIndexChange	Se distribuye cuando cambia el valor de la propiedad tabIndex del objeto.	No se admite

Clase Keyboard

La clase Keyboard se utiliza para crear una interfaz que puede controlar un usuario con un teclado estándar. Los métodos y las propiedades de la clase Keyboard se pueden utilizar sin usar un constructor. Las propiedades de la clase Keyboard son constantes que representan las teclas más utilizadas para controlar juegos.

Resumen de métodos

La siguiente tabla incluye los métodos de la clase Keyboard que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
isAccessible()	Especifica si a la última tecla presionada se puede acceder mediante otros archivos SWF.	No se admite

Clase KeyboardEvent

Flash Player distribuye objetos KeyboardEvent como respuesta a la entrada del usuario mediante un teclado. Existen dos tipos de eventos de teclado: KeyboardEvent.KEY_DOWN y KeyboardEvent.KEY_UP.

Resumen de métodos

La siguiente tabla incluye los métodos de la clase KeyboardEvent que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
updateAfterEvent()	Indica a Flash Player que realice la representación una vez finalizado el procesamiento de este evento, si la lista de visualización se ha modificado.	No se admite

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase KeyboardEvent que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
altKey	Indica si la tecla Alt se encuentra activa (true) o inactiva (false).	No se admite
ctrlKey	Indica si la tecla Control se encuentra activa (true) o inactiva (false).	No se admite
shiftKey	Indica si la tecla Mayús se encuentra activa (true) o inactiva (false).	No se admite

Clase LocalConnection

La clase LocalConnection permite crear un objeto LocalConnection que puede invocar un método en otro objeto LocalConnection.

Resumen de propiedades

La siguiente tabla incluye los métodos de la clase LocalConnection que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
isPerUser	Indica si el objeto LocalConnection está en el ámbito del usuario actual (true) o es globalmente accesible para todos los usuarios del equipo (false).	No se admite

Clase MouseEvent

Player distribuye objetos MouseEvent en el flujo del evento cada vez que se producen eventos de ratón. Un evento de ratón suele generarse mediante un dispositivo de entrada de usuario como, por ejemplo, un ratón o una bola de seguimiento, que utilice un puntero.

Resumen de métodos

La siguiente tabla incluye los métodos de la clase MouseEvent que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
updateAfterEvent()	Si se ha modificado la lista de visualización, da instrucciones a Flash Player para que muestre la representación tras finalizar el procesamiento del evento.	No se admite

Resumen de eventos

La siguiente tabla incluye el evento de la clase MouseEvent que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Evento	Descripción	Compatibilidad
MOUSE_WHEEL()	Define el valor de la propiedad type de un objeto de evento mouseWheel.	No se admite

Clase MovieClip

La clase MovieClip hereda de las siguientes clases: Sprite, DisplayObjectContainer, InteractiveObject, DisplayObject y EventDispatcher.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase MovieClip que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
trackAsMenu	Indica si otros objetos de visualización que son SimpleButton o MovieClip pueden recibir eventos de liberación de botón.	No se admite

Clase NetConnection

La clase NetConnection crea una conexión bidireccional entre Flash Player y una aplicación de Flash Media o entre Flash Player y un servidor de aplicaciones que ejecute Flash Remoting.

Un objeto NetConnection object es como una canalización entre el cliente y el servidor. Utilice los objetos NetStream para enviar flujos a través de la canalización.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase NetConnection que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
farID	Identificador de la instancia de Flash Media Server a la que se conecta esta instancia de Flash Player o Adobe AIR.	No se admite
farNonce	Valor elegido básicamente por Flash Media Server, exclusivo de esta conexión.	No se admite
maxPeerConnections	Número total de conexiones homólogas entrantes y salientes que permite esta instancia de Flash Player o Adobe AIR.	No se admite
nearID	Identificador de esta instancia de Flash Player o Adobe AIR para esta instancia de NetConnection.	No se admite
nearNonce	Valor elegido básicamente por esta instancia de Flash Player o Adobe AIR, exclusivo de esta conexión.	No se admite
protocol	Protocolo utilizado para establecer la conexión.	No se admite
unconnectedPeerStreams	Objeto que alberga todos los objetos NetStream del suscriptor homólogo que no se asocian a los objetos NetStream de publicación.	No se admite

Clase NetStream

La clase NetStream abre una conexión de corriente unidireccional continua entre Flash Player y Flash Media Server o entre Flash Player y el sistema de archivos local. Un objeto NetStream object es un canal en un objeto NetConnection. Este canal puede publicar un flujo, utilizando NetStream.publish(), o bien, suscribirse a un flujo publicado y recibir datos, utilizando NetStream.play().

Se pueden publicar o reproducir datos en vivo (en tiempo real) y datos grabados con anterioridad. Los objetos NetStream también se pueden utilizar para enviar mensajes de texto a todos los clientes suscritos (consulte el método NetStream.send()).

Resumen de métodos

La siguiente tabla incluye los métodos de la clase NetStream que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
checkPolicyFile	Especifica si Flash Player debe intentar descargar un archivo de política URL desde el servidor del archivo FLV cargado antes de comenzar a cargarlo.	No se admite
publish()	Envía flujos de audio, vídeo y mensajes de texto desde un cliente a Flash Media Server, grabando opcionalmente el flujo durante la transmisión.	No se admite
send()	Envía un mensaje en un flujo publicado a todos los clientes suscritos.	No se admite

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase NetStream que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
checkPolicyFile	Especifica si Flash Player debe intentar descargar un archivo de política URL desde el servidor del archivo FLV cargado antes de comenzar a cargarlo.	No se admite
farID	Identificador del extremo que está conectado a esta instancia de NetStream.	No se admite
farNonce	Valor elegido básicamente por el otro extremo de este flujo, exclusivo de esta conexión.	No se admite
peerStreams	Objeto que alberga todas las instancias suscritas de NetStream que detectan esta instancia de publicación de NetStream.	No se admite
soundTransform	Controla el sonido en este objeto NetStream.	No se admite

Clase SimpleButton

La clase SimpleButton permite controlar todas las instancias de los símbolos de botón de un archivo SWF. Tras crear una instancia de un botón en la herramienta de edición, puede utilizar los métodos y propiedades de la clase SimpleButton para manipular botones con ActionScript.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase SimpleButton que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
soundTransform	Objeto SoundTransform asignado a este botón.	No se admite
trackAsMenu	Indica si otros objetos de visualización que son SimpleButton o MovieClip pueden recibir eventos de liberación de botón.	No se admite
useHandCursor	Valor booleano que, cuando se establece en true, indica si Flash Player muestra el cursor de mano cuando el ratón pasa por encima de un botón.	No se admite

Clase Sound

La clase Sound permite trabajar con sonido en una aplicación. La clase Sound permite crear un objeto Sound, cargar y reproducir un archivo MP3 externo en el objeto, cerrar el flujo de sonido y acceder a datos sobre el sonido como, por ejemplo, información sobre el número de bytes del flujo y los metadatos ID3.

Se puede lograr un mayor control del sonido mediante la fuente de sonido (los objetos SoundChannel o Microphone del sonido) y a través de las propiedades de la clase SoundTransform que controlan la salida del sonido en los altavoces del equipo.

Resumen de eventos

La siguiente tabla incluye los eventos de la clase Sound que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Evento	Descripción	Compatibilidad
sampleData	Se distribuye cuando el reproductor solicita nuevos datos de audio.	No se admite

Resumen de métodos

La siguiente tabla incluye los métodos de la clase Sound que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
extract()	Extrae datos de sonido sin formato de un objeto Sound.	No se admite

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase Sound que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
id3	Proporciona acceso a los metadatos que forman parte de un archivo MP3.	Se admite de forma parcial

Clase SoundMixer

La clase SoundMixer contiene propiedades y métodos estáticos para el control global del sonido en la aplicación. La clase SoundMixer controla los sonidos de flujo e incorporados en la aplicación. No controla los sonidos creados dinámicamente (es decir, los sonidos generados como respuesta a un objeto Sound que distribuye un evento sampleData).

Resumen de métodos

La siguiente tabla incluye los métodos de la clase SoundMixer que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
areSoundsInaccessible()	Determina si no se puede acceder a algún sonido debido a restricciones de seguridad.	No se admite
computeSpectrum()	Toma una instantánea de la onda de sonido actual y la coloca en el objeto ByteArray especificado.	No se admite

Clase Sprite

La clase Sprite es un elemento constituyente básico de la lista de visualización: un nodo de la lista de visualización que puede mostrar gráficos y contener también elementos secundarios.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase Sprite que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
useHandCursor	Valor booleano que indica si debe aparecer la mano que señala (cursor de mano) cuando el ratón pasa por encima de un elemento sprite en el que la propiedad buttonMode se ha establecido en true.	No se admite

Clase Stage

La clase Stage representa el área de dibujo principal. La clase Stage representa toda el área donde se muestra el contenido Flash. No es posible obtener acceso al objeto Stage de forma global. Se debe acceder al mismo mediante la propiedad stage de una instancia de DisplayObject.

Resumen de métodos

La siguiente tabla incluye los métodos de la clase Stage que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
assignFocus()	Establece la selección mediante teclado en el objeto interactivo especificado por objectToFocus, con la dirección de la selección especificada mediante el parámetro de dirección.	No se admite
willTrigger()	Comprueba si hay registrado un detector de eventos con este objeto EventDispatcher o con cualquiera de sus ascendientes para el tipo de evento concreto.	No se admite

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase Stage que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
colorCorrection	Controla la corrección de color de Flash Player para las visualizaciones.	No se admite
colorCorrectionSupport	Especifica si Flash Player se está ejecutando en un sistema operativo que admite la corrección de color y si el perfil de color del monitor principal se puede leer y entender mediante Flash Player.	No se admite
showDefaultContextMenu	Especifica si deben mostrarse u ocultarse los elementos predeterminados del menú contextual de Flash Player.	No se admite
tabChildren	Determina si los elementos secundarios del objeto están habilitados para tabulación.	No se admite
textSnapshot	Devuelve un objeto TextSnapshot para esta instancia de DisplayObjectContainer.	No se admite

Clase System

La clase System contiene propiedades relacionadas con determinadas operaciones que tienen lugar en el equipo del usuario, como las realizadas con objetos compartidos, la configuración local de cámaras y micrófonos y el uso del portapapeles.

Resumen de métodos

La siguiente tabla incluye los métodos de la clase System que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
exit()	Cierra Flash Player.	No se admite
setClipboard()	Reemplaza el contenido del portapapeles por una cadena de texto especificada.	No se admite

Clase TextBlock

La clase TextBlock es una fábrica para la creación de objetos TextLine, que se pueden representar colocándolos en la lista de visualización.

La clase TextBlock incluye un sólo párrafo, ya que los algoritmos de salto de línea y bidireccionales Unicode funcionan en un párrafo al mismo tiempo. Para las aplicaciones que se componen de varios párrafos de texto, utilice un lenguaje de marcado o análisis de texto para dividir el texto en párrafos y crear un objeto TextBlock por párrafo.

Resumen de métodos

La siguiente tabla incluye los métodos de la clase TextBlock que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
dump()	Vuelca el contenido subyacente de TextBlock como cadena XML.	No se admite

Clase TextField

La clase TextField se utiliza para crear objetos de visualización para ver e introducir texto. Todos los campos dinámicos y de entrada de texto de un archivo SWF son instancias de la clase TextField. Puede asignar a un campo de texto un nombre de instancia en el inspector de propiedades y utilizar los métodos y propiedades de la clase TextField para manipularlo con ActionScript. Los nombres de instancias de TextField se muestran en el explorador de películas y en el cuadro de diálogo Insertar ruta de destino del panel Acciones.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase TextField que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
antiAliasType	Tipo de suavizado que se utiliza en este campo de texto.	No se admite
gridFitType	Tipo de ajuste de cuadrícula que se utiliza en este campo de texto.	No se admite
mouseWheelEnabled	Valor booleano que indica si Flash Player debe desplazar automáticamente los campos de texto de varias líneas cuando se hace clic en un campo de texto y se hace girar la rueda del ratón.	No se admite
sharpness	Nitidez de los bordes de glifo en este campo de texto.	No se admite
thickness	Grosor de los bordes de glifo en este campo de texto.	No se admite
useRichTextClipboard	Especifica si se copia y se pega el formato del texto junto con el propio texto.	No se admite

Clase TextFormat

La clase TextFormat representa la información relativa al formato de los caracteres. Utilice la clase TextFormat para crear formato de texto específico para los campos de texto. Puede aplicar formato de texto tanto a campos de texto estáticos como dinámicos. Las propiedades de la clase TextFormat se aplican a las fuentes incorporadas y de dispositivo.

Sin embargo, en el caso de las fuentes incorporadas, el texto en negrita y en cursiva requiere fuentes específicas. Si desea mostrar texto en negrita o en cursiva con una fuente incorporada, deberá incorporar las variaciones en negrita y cursiva de dicha fuente.

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase TextFormat que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
kerning	Valor booleano que indica si el ajuste de caracteres está activado (true) o desactivado (false).	No se admite
letterSpacing	Número que representa la cantidad de espacio distribuido uniformemente entre todos los caracteres.	No se admite

Clase TextLine

La clase TextLine se utiliza para mostrar texto en la lista de visualización.

No se puede crear un objeto TextLine directamente desde el código ActionScript. Si se llama a `new TextLine()`, se genera una excepción. Para crear un objeto TextLine, llame al método `createTextLine()` de un objeto TextBlock.

Resumen de métodos

La siguiente tabla incluye las propiedades de la clase TextLine que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
dump()	Vuelca el contenido subyacente de TextLine como cadena XML.	No se admite

Clase Transform

La clase Transform proporciona acceso a las propiedades de ajuste de color y a dos o tres objetos de transformación tridimensional que se pueden aplicar a un objeto de visualización. Durante la transformación, el color o la orientación y la posición de un objeto de visualización se ajusta (desplazamiento) a partir de las coordenadas o los valores actuales a los nuevos valores o coordenadas.

La clase Transform también recopila datos sobre el color y las transformaciones de matriz bidimensionales que se aplican a un objeto de visualización y a todos sus objetos principales. Se puede acceder a estas transformaciones combinadas mediante las propiedades `concatenatedColorTransform` y `concatenatedMatrix`.

Resumen de métodos

La siguiente tabla incluye las propiedades de la clase Transform que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
<code>getRelativeMatrix3D()</code>	Devuelve un objeto <code>Matrix3D</code> , que puede transformar el espacio de un objeto de visualización especificado en relación con el espacio del objeto de visualización actual.	No se admite

Resumen de propiedades

La siguiente tabla incluye las propiedades de la clase Transform que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Propiedad	Descripción	Compatibilidad
<code>matrix3D</code>	Proporciona acceso al objeto <code>Matrix3D</code> de un objeto de visualización tridimensional.	No se admite
<code>perspectiveProjection</code>	Proporciona acceso al objeto <code>PerspectiveProjection</code> de un objeto de visualización tridimensional.	No se admite

Clase Video

La clase Video muestra vídeo grabado o en directo en una aplicación sin incorporar el vídeo en el archivo SWF. Esta clase crea un objeto Video en una interfaz de Adobe Flash o Adobe Flex que reproduce cualquiera de los siguientes tipos de vídeo:

- Archivos FLV grabados almacenados en un servidor o de forma local.
- Vídeo en directo capturado desde el equipo de un usuario.

Un objeto Video es un objeto de visualización de la lista de visualización de la aplicación y representa el espacio visual en el que se ejecuta el vídeo en una interfaz de usuario.

Resumen de métodos

La siguiente tabla incluye las propiedades de la clase Video que no admite Flash Lite 4 cuando se utiliza ActionScript 3.0.

Método	Descripción	Compatibilidad
<code>attachCamera()</code>	Especifica un flujo de vídeo de una cámara para que se muestre dentro de los límites del objeto Video en la aplicación.	No se admite

Elementos de Actionscript no admitidos: detalles

Algunas clases de ActionScript 3.0 no se admiten en ActionScript 3.0 para Flash Lite 4. Estas clases se separan en función de los paquetes a los que pertenecen.

Para obtener información sobre estas clases, consulte *Adobe ActionScript 3.0 Reference for the Adobe Flash Platform*.

adobe.utils

El paquete `adobe.utils` contiene funciones y clases empleadas por los desarrolladores de la herramienta de edición de Flash.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- CustomActions
- XMLUI

flash.accessibility

El paquete `flash.accessibility` contiene clases que permiten la accesibilidad en el contenido y las aplicaciones de Flash.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- Accessibility
- AccessibilityProperties

flash.desktop

El paquete `flash.desktop` contiene clases utilizadas para las operaciones de copiar y pegar y arrastrar y colocar, así como la clase `Icon`, utilizada para definir iconos del sistema utilizados por un archivo.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- Clipboard
- ClipboardFormats
- ClipboardTransferMode

flash.display

El paquete `flash.display` contiene clases utilizadas para las operaciones de copiar y pegar y arrastrar y colocar, así como la clase `Icon`, utilizada para definir iconos del sistema utilizados por un archivo.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- BlendMode
- CapsStyle
- ColorCorrection
- ColorCorrectionSupport
- FocusDirection
- GraphicsBitmapFill
- GraphicsEndFill

- GraphicsGradientFill
- GraphicsPath
- GraphicsPathCommand
- GraphicsPathWinding
- GraphicsShaderFill
- GraphicsSolidFill
- GraphicsStroke
- GraphicsTrianglePath
- IGraphicsData
- IGraphicsFill
- IGraphicsPath
- IGraphicsStroke
- InterpolationMethod
- JointStyle
- LineScaleMode
- NativeMenu
- NativeMenuItem
- NativeWindow
- NativeWindowDisplayState
- NativeWindowInitOptions
- NativeWindowResize
- NativeWindowSystemChrome
- NativeWindowType
- Pantalla
- Shader
- ShaderData
- ShaderInput
- ShaderJob
- ShaderParameter
- ShaderParameterType
- ShaderPrecision
- SpreadMethod
- TriangleCulling

flash.errors

El paquete flash.error contiene clases de error que forman parte de la interfaz de programación de aplicaciones (API) de Flash Player y no del núcleo del lenguaje ActionScript.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- DRMMangerError
- SQLError
- SQLErrorOperation

flash.events

El paquete flash.events admite el nuevo modelo de evento DOM e incluye la clase base EventDispatcher.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- ActivityEvent
- BrowserInvokeEvent
- DRMAuthenticateEvent
- DRMAuthenticationCompleteEvent
- DRMAuthenticationErrorEvent
- DRMErrorEvent
- DRMStatusEvent
- FileListEvent
- HTMLUncaughtScriptExceptionEvent
- InvokeEvent
- NativeDragEvent
- NativeWindowBoundsEvent
- NativeWindowDisplayStateEvent
- OutputProgressEvent
- SampleDataEvent
- ScreenMouseEvent
- ShaderEvent
- SQLErrorEvent
- SQLEvent
- SQLUpdateEvent

flash.filters

El paquete flash.filters contiene clases para efectos de filtro de mapa de bits. Los filtros permiten aplicar efectos visuales muy diversos, como desenfoque, bisel, iluminación y sombreado, para mostrar objetos.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- BevelFilter
- BitmapFilter
- BitmapFilterQuality
- BitmapFilterType

- BlurFilter
- ColorMatrixFilter
- ConvolutionFilter
- DisplacementMapFilter
- DisplacementMapFilterMode
- DropShadowFilter
- GlowFilter
- GradientBevelFilter
- GradientGlowFilter
- ShaderFilter

flash.geom

El paquete flash.geom contiene clases geométricas, como puntos, rectángulos y matrices de transformación, que permiten utilizar la clase BitmapData y la función de almacenamiento en caché de mapa de bits.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- Matrix3D
- Orientation3D
- PerspectiveProjection
- Utils3D
- Vector3D

flash.media

El paquete flash.media contiene clases para trabajar con activos multimedia, como sonido y vídeo. También contiene las clases de vídeo y audio disponibles en Flash Media Server.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- Camera
- Microphone
- scanHardware
- SoundCodec

flash.net

El paquete flash.net contiene clases para el envió y la recepción de red, como, por ejemplo, la descarga de URL y Flash Remoting.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- FileFilter
- FileReference
- FileReferenceList

flash.printing

El paquete flash.printing contiene clases para la impresión de contenido basado en Flash.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- PrintJob
- PrintJobOptions
- PrintJobOrientation

flash.profiler

El paquete flash.profiler contiene funciones utilizadas para la depuración y la creación de perfiles de código ActionScript.

La clase ShowRedrawRegions no se admiten en ActionScript 3.0 para Flash Lite 4.

flash.sampler

El paquete flash.sampler contiene métodos y clases realizar un seguimiento de las llamadas a procedimiento de modo que se pueda crear un perfil del uso de memoria y optimizar las aplicaciones. Este paquete lo utiliza el agente de creación de perfiles distribuido con Adobe Flex Builder y se proporciona para su uso a la hora de personalizar el agente de creación de perfiles o crear sus propias pruebas de memoria. Una vez detectado el lugar en el que la aplicación utiliza más memoria, los esfuerzos de optimización se pueden centrar para acelerar el rendimiento de la aplicación.

La clase StackFrame no se admite en ActionScript 3.0 para Flash Lite 4.

flash.system

El paquete flash.system contiene clases para el acceso a funcionalidad a nivel del sistema, como, por ejemplo, seguridad, recolección de elementos no utilizados, etc.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- IME
- IMEConversionMode

flash.text

El paquete flash.text contiene clases para trabajar con campos de texto, formato de texto, medidas del texto, hojas de estilo y diseño. El suavizado avanzado está disponible en Flash Player 8 y versiones posteriores a través de las clases flash.text.TextFormat y flash.text.TextRenderer.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- AntiAliasType
- CSMSettings
- FontStyle
- GridFitType
- TextColorType
- TextDisplayMode
- TextExtent

- TextRenderer

flash.ui

El paquete flash.ui contiene clases de interfaz de usuario, como, por ejemplo, clases para interactuar con el ratón y el teclado.

Las siguientes clases no se admiten en ActionScript 3.0 para Flash Lite 4.

- ContextMenu
- ContextMenuBuiltInItems
- ContextMenuClipboardItems
- ContextMenuItem
- KeyLocation
- Mouse
- MouseCursor