

Amig@s:

Desde hace unos días tengo la ocasión de probar la beta de Flash Pro 8, conocida por todos como "8ball", cuya versión definitiva se espera para el mes de septiembre, con lo que la versión española estará para Navidades si queremos ser optimistas, aunque el asunto ha levantado tal expectación entre los desarrolladores que el chiste de moda dice:

¿Sabes cuándo saldrá Flash 8?

- Sí, claro, el ocho de Agosto de 2008 (8/8/08).

Sí, ya sé que es bastante malo, pero no podemos pedir a los developers que, además de listos, sean graciosos, aunque los hay que sí, pero no ejercen por la cuenta que les tiene, je, je.

En el mundillo se insiste en que el retraso tiene que ver con el anuncio de la compra de Macromedia por Adobe, paralizada por los tribunales hasta fin de año, pese a los continuos desmentidos del jefe del equipo de desarrollo Flash, Mike Chambers, que estos días demostraba la poca gracia que le hace el dicho de que el nuevo Flash se llamará Flash CS (por comparación con Photoshop, programa que ya ha sacado el Photoshop CS 2 en inglés, pero como el mundo hispano es tan pirata ya se puede ver en muchas máquinas instalado ¡¡¡En español!!!).

Bueno, mi intención con este mensaje es analizar algunas de las novedades que incorporará Flash 8, permitidme un leve repaso:

Tras una carátula de entrada que bien nos puede recordar el paquete de tabaco de Fortuna, nos encontramos con una de las versiones de interface más bonita y simple de Flash, que sigue la apariencia de Flash MX 2004, pero en la que se han desarrollado y asentado nuevas mejoras que el equipo de desarrolladores clasifican en tres grupos, gráficos, video y dispositivos móviles. Pero como yo no soy de su equipo, haré este repaso al nuevo programa bajo mi criterio, no bajo el suyo

Mi punto de vista es que esta es una versión más enfocada a completar las herramientas de diseño y video que las enfocadas a la programación. Y en este aspecto quiero recordar que en la versión 7 Flash se presentó por vez primera en dos aplicaciones separadas, una para diseñadores y otra, la profesional, para programadores. Pero los diseñadores gráficos también se consideran profesionales, y a pesar de las advertencias, usaban Flash Pro porque además incorporaba más herramientas. La sorpresa era que no podían programar en modo normal, como antes, sino que debían conocer la sintaxis de action script y escribir el código 'a mano'.

Ahora tenemos un Flash Basic 8 y un Flash Profesional 8, ambos con capacidades diferentes que podemos contrastar en <http://www.macromedia.com/es/software/flash/basic/>, pero en ambos aparece el modo normal de programación.

Flash se robustece en su arquitectura interna. Presenta un nuevo reproductor Flash MediaPlayer 8, con capacidad para mostrar metadatos a los motores de búsqueda en internet y especialmente enfocado a ser un buen lector de pantallas accesibles.



Pero quede claro que este no es un documento oficial de Macromedia, ni siquiera es un documento de Macromedia, sino mis opiniones personales. Por ello no comentaré los cambios tontos en la interface ni los cambios menores presentados como auténticos logros del equipo de desarrolladores.

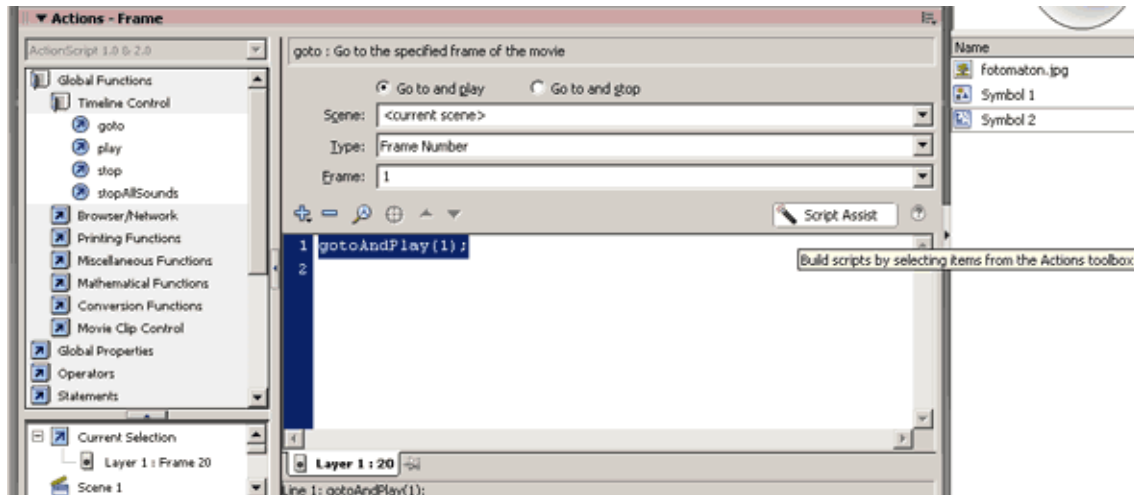
Sobre ella, sobre **la programación**, destaco que:

¡¡¡VUELVE EL MODO 'NORMAL' EN EL PANEL DE ACTION SCRIPT!!!, También se puede escribir el código 'a mano' –modo experto en la versión 7- pero el panel incorpora un botoncito de asistente de script que da paso a 'la vuelta a la normalidad'.

Personalmente pienso que el intervalo de programación forzosa "a mano", así como la programación orientada a objetos y la dura sintaxis de AS2, ha forzado a muchos desarrolladores a estudiar y practicar el código de action script, con lo que han elevado considerablemente su nivel de calidad, pero ciertamente las quejas de los desarrolladores gráficos han sido continuas en este aspecto.

Flash se ha caracterizado en otros tiempos precisamente por dar paso a la programación a un gran número de grafistas gracias a la facilidad del uso en 'modo normal' del panel de acciones desde el que se programa.

Sin duda la vuelta del 'modo normal' de programar es una ventaja añadida de la aplicación.



Pero lo que nos interesa a los programadores son las novedades que incorpora el lenguaje de programación: Son variadas y lo hacen tanto en el aspecto de nuevos elementos del lenguaje como en nuevas herramientas de implementación y edición.

Una de las más novedosas es la nueva ventana de cadenas, que es una herramienta para trabajar con archivos xml y campos de textos. A través de ella podemos ahora implementar las traducciones de nuestra web en Flash con mucha facilidad.

En el panel de acciones, existen pocos cambios, pero interesantes, como la posibilidad de releer antiguos scripts, o de editarlos con texto que incluye caracteres ocultos.

En cuanto a los cambios realizados sobre el lenguaje Action Script, no son drásticos, por lo que conserva aún la denominación Action Script 2, pero sí destacan unas 20 nuevas clases y casi 50 nuevos métodos y propiedades, y también destaca que siguen siendo totalmente usables todas las clases, métodos y propiedades de la versión anterior. Creo que el mismo nombre de las clases y acciones nuevas es significativo, pero, aunque no las he probado, cuento lo que hacen:

Clases:

La clase BevelFilter .- Permite agregar efectos biselados a los objetos

La clase BitmapData .- Permite crear y modificar bitmaps opacos y transparentes.

La clase BitmapFilter .- Es una clase base para la aplicación de efectos.

La clase BlurFilter .- Para aplicar desenfoco sobre los objetos.

La clase ColorMatrixFilter .- Aplica transformaciones sobre colores RGB y valores alfa.

La clase ColorTransform .- Sustituye a la vieja clase color (que puede se puede seguir usando), para dar valores de color a los movieclips.

La clase ConvolutionFilter .- Permite aplicar efectos de filtros a matrices.

La clase DisplacementMapFilter .- Permite utilizar valores de pixels de un objeto desde BitmapData para realizar aplicaciones de filtros en otro objeto.

La clase DropShadowFilter .- Pues éso, sombras difuminadas de objetos.

La clase ExternalInterface .- Permite comunicarsae a través de Action Script con el contenedor de Flash Player (el sistema lleva a cabo la aplicación de Flash como un browser con javascript o como una aplicación de escritorio)

La clase FileReference .- Permite subir y bajar archivos entre el cliente y el servidor.

La clase FileReferenceList .- Permite seleccionar uno o más archivos para subirlos al server.

La clase GlowFilter .- Efecto de resplandor sobre los objetos

La clase GradientBevelFilter .- Agrega biselados con gradientes a los objetos

La clase GradientGlowFilter .- Y ésta biselados con resplandor.

La clase IME .- Permite manipular el IME (input method editor) del sistema operativo desde Flash.

La clase Locale .- Es la que nos da información y nos permite mostrar un texto de un idioma concreto por defecto en nuestras aplicaciones de Flash.

La clase Matrix .- Representa una transformación matricial que determina como trazar puntos desde un espacio con coordenadas a otro.

La clase Point .- Determina la localización en un sistema de coordenadas bidimensional.

La clase Rectangle .- Crea y modifica objetos rectangulares.

La clase TextRenderer .- Proporciona la funcionalidad para el antialias de las fuentes embebidas.

La clase Transform .- Recoge datos sobre transformaciones del color y coordina las manipulaciones que aplicamos sobre una instancia de MovieClip.

Veamos ahora las nuevas funciones, propiedades y métodos aparecidos en la versión 8, que sobre todo afectan a las clases MovieClip, TextField y Video:

La función global **showRedrawRegions** proporciona la capacidad para el depurador de Flash de contornear las regiones de la pantalla que están siendo redibujadas (es decir, las regiones sucias que están siendo actualizadas).

La propiedad **blendMode** en la clase Button, que fija el modo de mezcla para instancias de botón.

La propiedad **cacheAsBitmap** en la clase Button, que permite almacenar el objeto en la memoria como representación de imagen interna de la instancia.

La propiedad **filters** en la clase Button, que es un arsenal puesto en un índice que contiene cada objeto del filtro asociado al botón.

La propiedad **scale9Grid** de la clase Button, que es la región rectangular que define nueve regiones de escalamiento para la instancia.

La propiedad **hasIME** en la clase de System.capabilities, que indica si el sistema tiene un IME instalado.

La propiedad **getUTCYear** de la clase Date, que devuelve el año en curso, según tiempo universal.

El **ALT** constante en la clase dominante.

El método **isAccessible()** en la clase dominante, devuelve un valor booleano que indica si la última llave presionada puede ser accesible desde otros archivos de SWF, dependiendo de restricciones de seguridad.

El evento **onHTTPStatus** de la clase LoadVars devuelve el código de estado enviado de vuelta por el servidor (por ejemplo, el valor 404 para la página no encontrada).

El método **attachBitmap()** de la clase MovieClip, que une (atacha) un bitmap a un clip de película.

El método **beginBitmapFill()** de la clase MovieClip, que llena un clip de la película con una imagen.

El **spreadMethod**, el **interpolationMethod**, y los parámetros del **focalPointRatio** del método del **beginGradientFill()** en la clase MovieClip. Este método llena un área de dibujo de una imagen, y el bitmap se puede repetir o fijar para llenar el área.

La propiedad **blendMode** de la clase de MovieClip, que permite fijar el modo de mezcla para la instancia.

La propiedad **cacheAsBitmap** de la clase de MovieClip, que permite almacenar el objeto en la memoria como representación interna en tipo bitmap de una instancia del movieclip, lo que evita la continua representación vectorial de un movieclip de vectores.

La propiedad **filters** de la clase MovieClip, que es una cadena indexada que contiene cada filtro asociado actualmente a la instancia.

El método **getRect()** de la clase MovieClip, que devuelve las propiedades que son los valores mínimos y máximos de las coordenadas de posición de la instancia especificada.

El método **lineGradientStyle()** de la clase MovieClip, que especifica un estilo del gradiente de la línea que utilizemos al dibujar una trayectoria.

Los parámetros **pixelHinting**, **noScale**, **capsStyle**, **jointStyle**, y **miterLimit** del método **lineStyle()** en la clase MovieClip. Estos parámetros especifican los estilos de líneas que podemos utilizar al dibujar líneas.

La propiedad **opaqueBackground** de la clase MovieClip, que fija el color del fondo (no transparente) opaco del clip de la película al color especificado por el valor hexadecimal del RGB.

La propiedad **scale9Grid** de la clase MovieClip, que es la región rectangular que define nueve regiones de escalamiento para el caso.

La propiedad **scrollRect** de la clase MovieClip, que permite hacer un scroll con el contenido del clip de la película y tener un contenido más grande que el que se ve en la ventana.

La propiedad **transform** de la clase MovieClip, que permite hacer ajustes con respecto a la matriz de un clip de la película, transformaciones de color y límites del píxel.

El parámetro **status** del evento MovieClipLoader.onLoadComplete devuelve el código de estado devuelto por el servidor (por ejemplo, el valor 404 para la página no encontrada).

El evento **onLoadError** de la clase MovieClipLoader se invoca cuando un archivo cargado con MovieClipLoader.loadClip() no se pudo cargar.

El parámetro **secure** del método SharedObject.getLocal() determina si el acceso a este objeto compartido está restringido a los archivos SWF servidos sobre una conexión de HTTPS.

El **ID3** es una constante de la clase Sounds.

El método **parse()** de la clase de TextField.styleSheet.

La propiedad **antiAliasType** de la clase TextField, que fija el tipo de anti-aliasing que utilizamos para la instancia de campo de texto.

La propiedad **filters** de la clase TextField, que es una cadena indexada que contiene cada filtro asociado actualmente a la instancia de campo de texto.

La propiedad **gridFitType** de la clase de TextField, que determina cómo fijamos la instancia de campo de texto a la rejilla.

La propiedad **sharpness** de la clase TextField, que fija la agudeza de los bordes del carácter tipográfico para la instancia de campo de texto.

La propiedad **thickness** de la clase TextField, que fija el grosor de los bordes del carácter tipográfico en la instancia de TextField. Debemos fijar el antiAliasType() a avanzado si utilizamos esta propiedad.

El valor **justify** para la propiedad **aling** de la clase de TextFormat, que nos permite, por fin, justificar un párrafo especificado.

La propiedad **indent** de la clase de TextFormat, que permite usar valores negativos.

La propiedad **kerning** de la clase de TextFormat, que permite kerning on y off para el objeto de TextFormat.

La propiedad **leading** de la clase de TextFormat, que permite valores negativos, así que el espacio entre las líneas es menos que la altura del texto. Esto permite poner líneas de texto juntas en las aplicaciones.

La propiedad **letterSpacing** de la clase de TextFormat, que permite especificar la cantidad de espacio que se distribuye uniformemente entre los caracteres.

La propiedad **_alfa** de la clase video, que es la cantidad especificada de transparencia para el objeto video.

La propiedad **_height** de la clase video, indica la altura de la instancia de video.

La propiedad **_name** de la clase video, indica el nombre de la instancia de video.

La propiedad **_parent** de la clase video, indica la instancia del clip de la película que contiene la instancia del video.

La propiedad **_rotation** de la clase video, que permite fijar la cantidad de rotación de la instancia video en grados.

La propiedad **_visible** de la clase video, permite establecer la visibilidad de una instancia de video.

La propiedad **_width** de la clase video, permite establecer la anchura de la instancia de video.

La propiedad de **_x** de la clase video, que fija la coordenada X de la instancia de video.

La propiedad **_xmouse** de la clase video, que fija la coordenada X de la posición del ratón.

La propiedad **_xscale** de la clase video, que fija el porcentaje horizontal de la escala de la instancia de video.

La propiedad **_y** de la clase video, que fija la coordenada Y de la instancia de video.

La propiedad **_ymouse** de la clase video, que fija la coordenada Y de la posición del ratón.

La propiedad **_yscale** de la clase video, que fija el porcentaje vertical de la escala de la instancia de video.

El evento **onHTTPStatus** de la clase XML devuelve el código de estado devuelto por el servidor (por ejemplo, el valor 404 para la página no encontrada).

La propiedad **localName** de la clase de XMLNode, que devuelve el nombre completo del objeto del nodo de XML (incluyendo ambos: el prefijo y el nombre local).

La propiedad **namespaceURL** de la clase de XMLNode, que lee la URL especificada en el prefijo del nodo del XML.

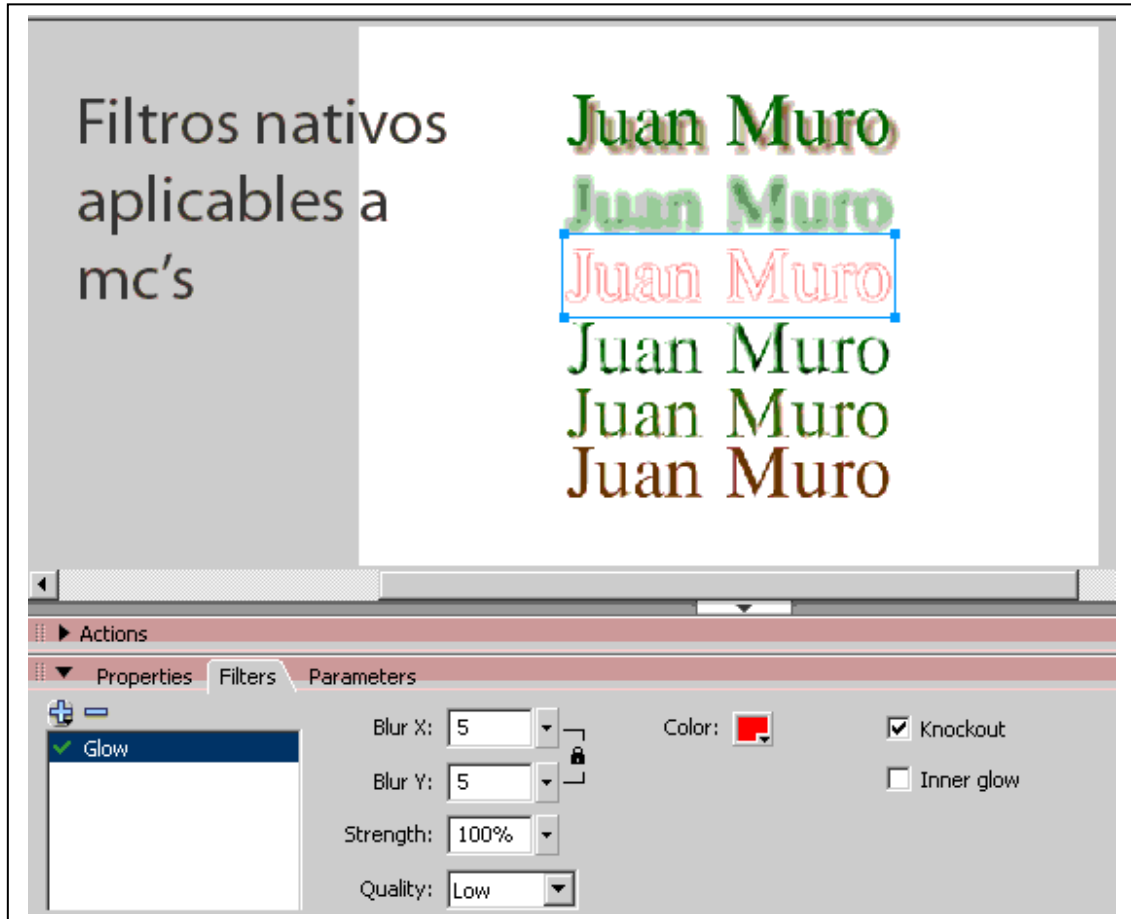
La propiedad **prefix** de la clase XMLNode, que lee el prefijo del nombre del nodo.

El método **getNamespaceForPrefix** de la clase XMLNode, que devuelve el namespace URI asociado al prefijo especificado en el nodo.

El método **getPrefixForNamespace** de la clase XMLNode, que devuelve el prefijo URI asociado a un namespace especificado en el nodo.

En cuanto a gráficos quiero destacar las siguientes mejoras:

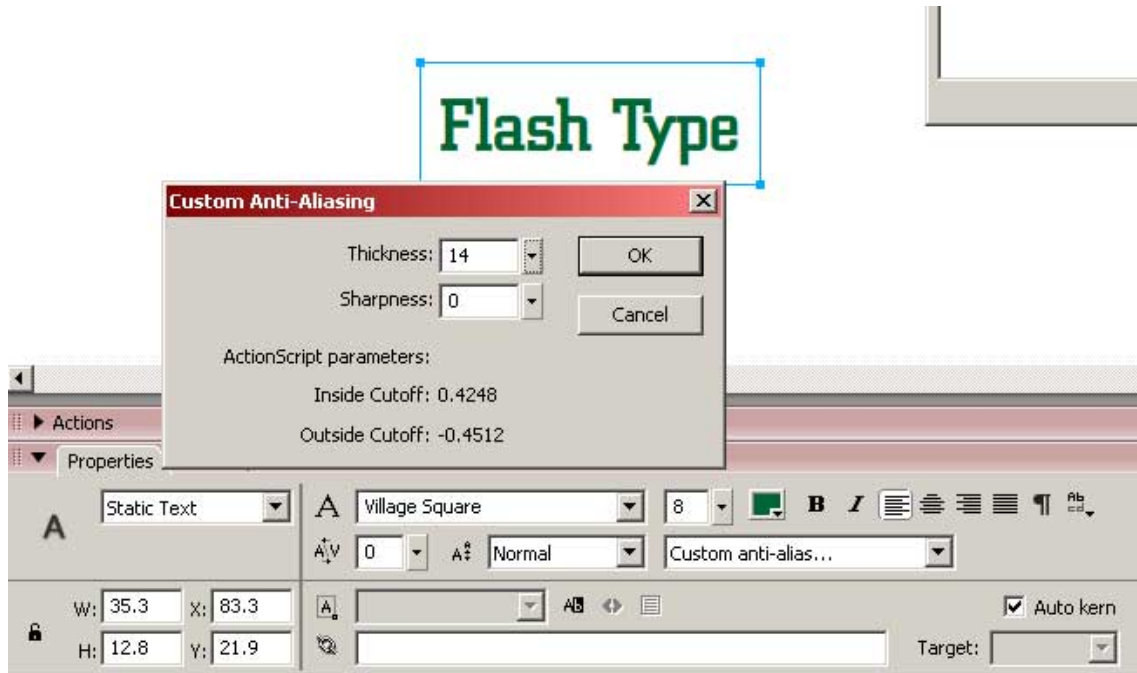
1º.- Por priemra vez Flash incorpora ahora una pestaña en la ventana de propiedades que nos lleva a nueva serie de filtros nativos que se pueden aplicar incluso sobre movieclips. No son muchos, pero sí muy valiosos, pues conceptualmente Flash entra en el mundo de la edición de bitmaps, algo que hasta ahora no hacía.



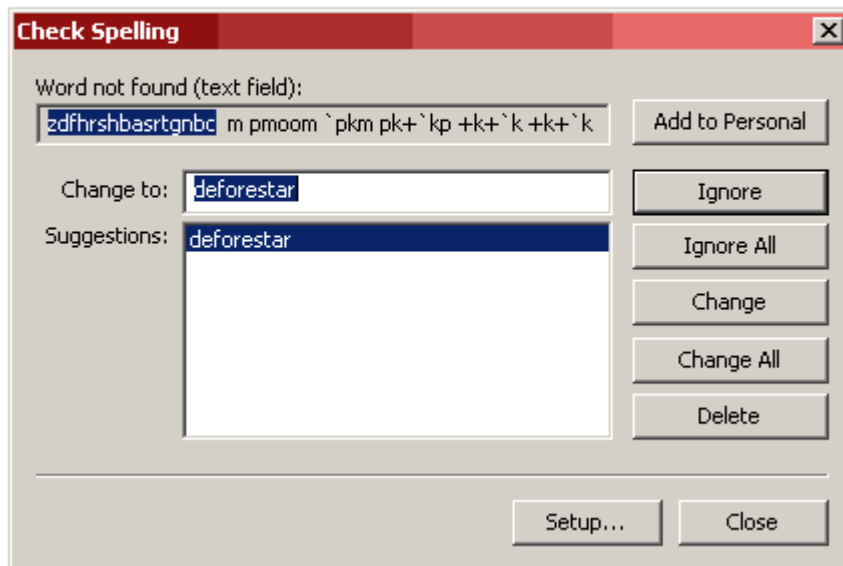
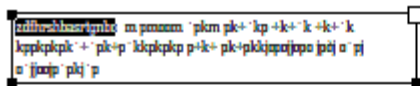
Además los filtros pueden aplicarse unos sobre otros, en el mismo movieclip, por ejemplo, en el siguiente ejemplo he aplicado a la imagen original (izda) un filtro Bevel (bisel), luego un drop sadow (sombra) y por último un Hue (brillo):



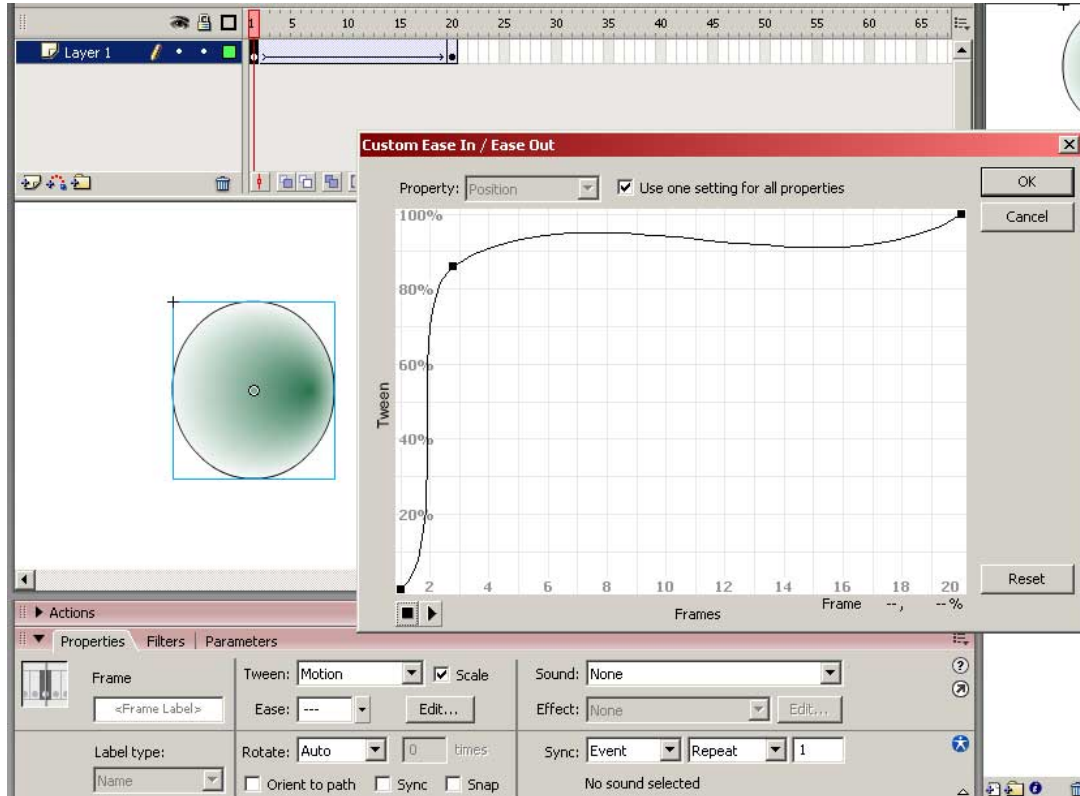
Un nuevo y potente motor de tratamiento de textos, con especial énfasis en resolver el eterno problema del antialias en la publicación. El nuevo motor se llama Flash Type y permite incluso la personalización del suavizado. La legibilidad de las fuentes estáticas pequeñas en campos estáticos está asegurada. ¡Ya era hora!:



Corregido el bug del programa que impedía justificar textos dinámicos (ya lo hemos visto, mediante el valor **justify** para la propiedad **aling** de la clase de TextFormat), Flash aporta ahora un corrector de textos que es otra de las grandes novedades de la versión 8. A él se accede desde el comando de Texto, aunque previamente debemos elegir nuestros diccionarios:

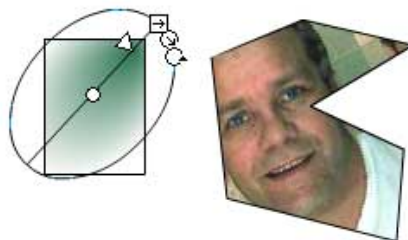


Mayor control de aceleración de animaciones.- Mediante una curva arrastrable muy intuitiva. Esta forma de animar, mucho más propia de programas avanzados de animación, es una herramienta muy poderosa de animadores profesionales, con la que se consiguen efectos elásticos inmediatos en nuestras animaciones, lo que produce que el animador resuelva los principios de anticipación y reacción de la animación clásica con más facilidad.



Es interesante resaltar que estas curvas pueden aplicarse a otros objetos, solamente copiando y pegando con Ctrl+C y Ctrl+V. Por supuesto existen muchas otras mejoras gráficas, pero son, por decirlo así, menores.

Mejor control de gradientes y rellenos:



Mayor control en trazados de bitmaps a vectores, pero también en el proceso inverso (vectores llamados tipo bitmap), lo que se hace a través de la clase **cacheAsBitmap**

Propiedades de trazo muy mejoradas, se permiten los degradados en los trazos, cambiar los tipos de línea y sus terminaciones mediante nuevas funciones de código.

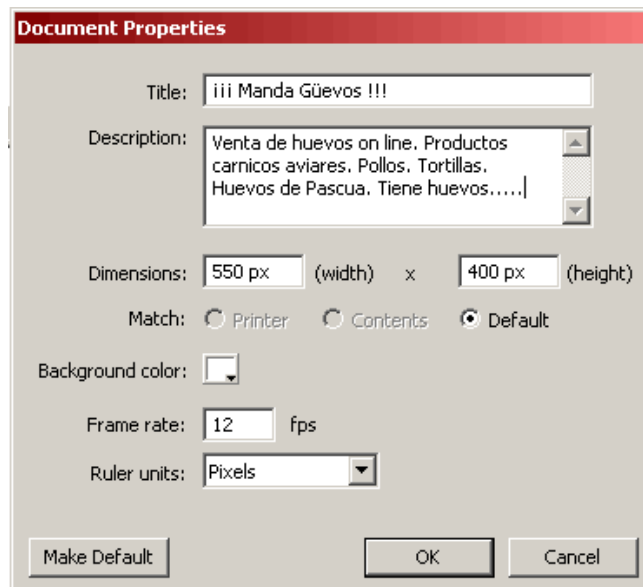
Nuevas modos blend, de muestras de bitmaps, parecidos a los de Photosop y manejables desde action script.

Ya se pueden cargar desde flash archivos externos del tipo jpg progresivos, gifs no animados y png's.

Efectos de linea de tiempo: Muy sencillos de usar y que nos evitan costosas animaciones:



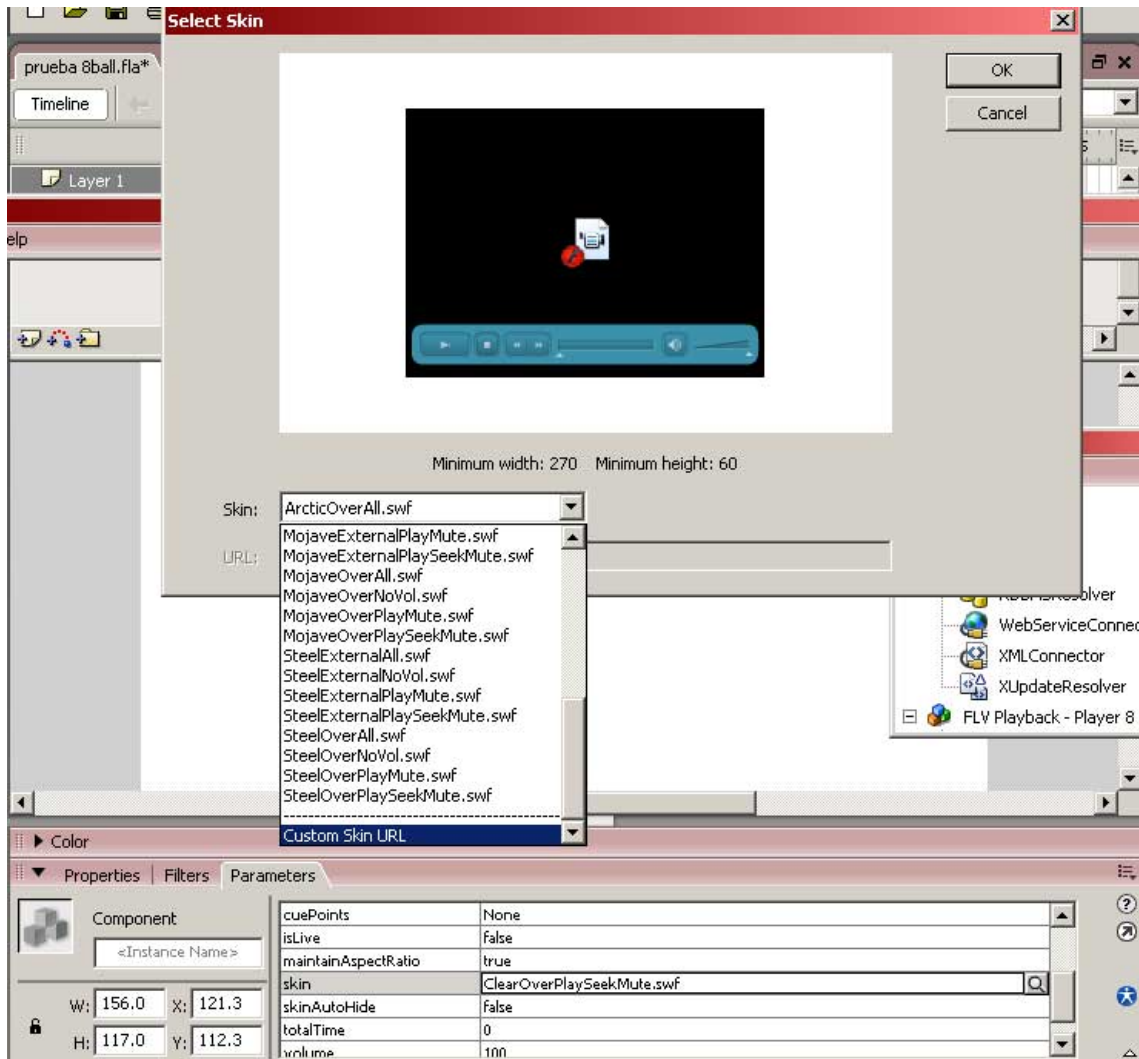
Respecto a la introducción de datos para ser leídos por los buscadores de internet, el asunto es tan simple ahora como meter las keyboards, tags y metatags en el campo de descripción de nuestra película :



Uno de los aspectos más cuidados en esta nueva versión de Flash ha sido el video. Cuando todo hacía pensar que la alianza entre Sorenson y Macromedia sería el único camino del video en Flash, Macromedia nos presenta ahora un nuevo codec propio, que se llama On2 VP6 y que compite muy duramente o el códec Sorenson Spark.

Además se ha desarrollado un plugin para los principales programas editores de video con el que se obtiene salida de video a Flash.

Y no solo se han mejorado los componentes de video con sencillas personalizaciones, sino que se han incluido nuevas funcionalidades, los puntos de inicio, que ayudan a coordinar el flujo de animación con el del video, pero también nuevas posibilidades como el uso del canal alfa en videos, lo que nos permite la superposición de videos y las transparencia y semitransparencias entre ellos.

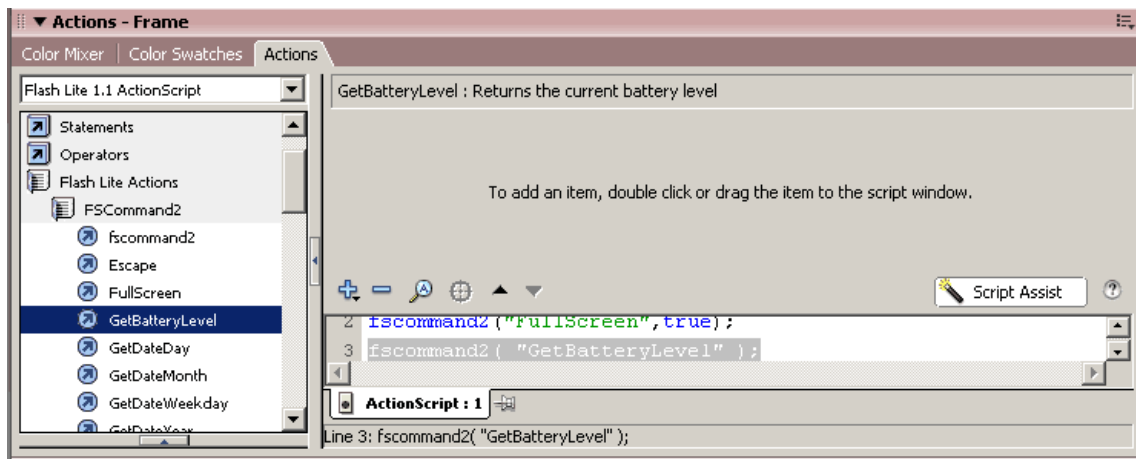


Y por último hay que resaltar el enorme avance de Flash en el terreno del desarrollo para dispositivos móviles. Flash Lite se ha robustecido de forma muy espectacular y admite ahora una nueva gama de acciones del lenguaje action script para Flash Lite, que adopta la versión 1.1 en la que se incluyen nuevas acciones: **LoadVars()**, **_capMMS**, **_capSMS** y **_cap4WayKeyAS** y **\$version**.

Sin embargo el programador avanzado de action script debe aprender a usar la función call para personalizar las funciones, debe saber emular arrays con pseudoarrays, debe saber de prototipos si es que piensa en AS2, también debe saber hacer mc's loopings para sustituir a los eventos y debe saber que Flash Lite 1.1 solamente permite cargar swf's como archivos externos, ni jpg's, ni gifs, ni png's, ni tan siquiera MP3.

A cambio de ello Flash nos da ahora una información muy completa sobre los dispositivos que aceptan las características de nuestra aplicación a través del panel de salida del emulador.

Y también es más que destacable la 'nueva' forma de programar para dispositivos móviles, pues el panel de acciones incorpora las funciones propias de FlashLite y puede programarse también en el modo normal:



Así que ahora ya es mucho más fácil trabajar con Flash para hacer aplicaciones para Teléfonos globales, PDA's y teléfonos japoneses, tal como clasifica Flash las plantillas que nos provee.

