
Kara Effector 3.2:

El inicio de este **Tomo IV** veremos la segunda parte de la Librería “fx” de **Kara Effector**. Estas son las variables que quedaron sin explicar del **Tomo** anterior y a continuación veremos un poco más de cada una de ellas:

Librería “fx”:

Template Type [fx]: Syl

Center in 'X' = syl.center 16

Center in 'Y' = syl.middle 17

Scale in 'X' = 18

Scale in 'Y' = 19

Align [\an] = 5 20

Pos in 'X' = valor x1, valor x2, valor x3, valor x4 21

Pos in 'Y' = valor y1, valor y2, valor y3, valor y4 22

Times Move = tiempo inicial, tiempo Final 23

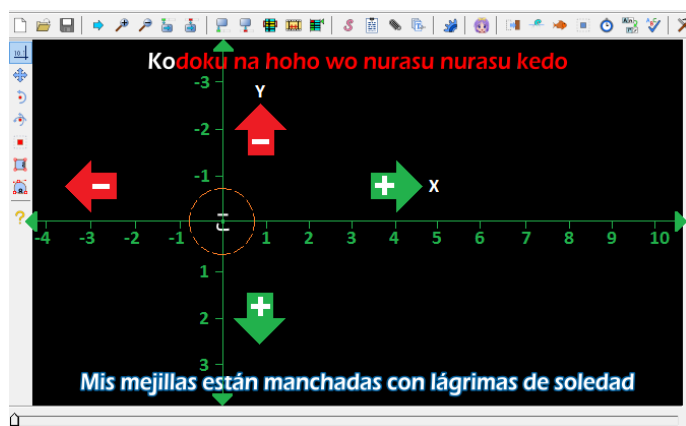
Add Tags: Add Tags Language: Lua

☐ Print Config [fx] Template Folder [fx]: lead-in[fx]

New [fx] Name:

- 16. **fx.center_x**
 - 17. **fx.center_y**
 - 18. **fx.scale_x**
 - 19. **fx.scale_y**
 - 20. **fx.align**
 - 21. **fx.move_x1, fx.move_x2, fx.move_x3, fx.move_x4**
 - 22. **fx.move_y1, fx.move_y2, fx.move_y3, fx.move_y4**
 - 23. **fx.movement_i, fx.movement_f**
-

fx.center_x: es el Centro medido en pixeles con respecto al eje “x” y hace las veces de la Abscisa al Origen de un sistema de coordenadas cartesianas para cada Línea de fx.



En la imagen anterior dibujé las coordenadas cartesianas con el origen en el centro del Hiragana “ko”:



La coordenada en “x” de ese origen sería **syl.center** que como su nombre lo indica, es el centro de la Sílabla. Nótese que en los archivos .ass, con respecto al eje “y”, hacia arriba es negativo y hacia abajo es positivo, es decir que este eje está invertido.

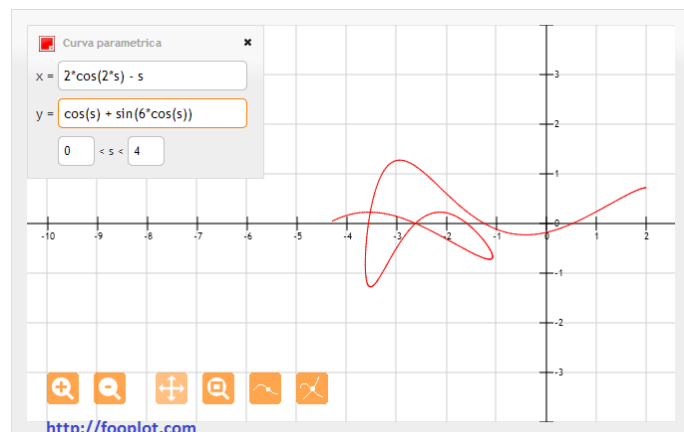
fx.center_y: es el Centro medido en pixeles con respecto al eje “y” y hace las veces de la Ordenada al Origen de un sistema de coordenadas cartesianas para cada Línea de fx.

Como podemos ver en el ejemplo del Hiragana “ko”, sería **syl.middle**, que es el centro de la Sílabla, pero medido verticalmente.

syl.center y **syl.middle** son variables de la Librería “syl” que originalmente ya viene en el Aegisub por default, pero que también explicaré en los próximos tomos, ya que he agregado algunas variables más a esta Librería y es necesario que sepamos en qué consisten dichas variables.

fx.scale_x: junto a **fx.scale_y**, es la Escala con respecto al eje “x” de las gráficas que se generen con ecuaciones paramétricas.

fx.scale_y: es la Escala con respecto al eje “y” de las gráficas que se generen con ecuaciones paramétricas.



Acá hay otro ejemplo de una Gráfica generada por ecuaciones paramétricas, y son este tipo de gráficas las que serán afectadas por las anteriores Escalas. El valor por default de ambas Escalas es 1.

Una vez conocidas las variables **fx.center_x**, **fx.center_y**, **fx.fun_x**, **fx.fun_y**, **fx.scale_x** y **fx.scale_y**; es importante que veamos las siguientes variables:

fx.pos_x: es la coordenada en “x” de la posición final de la Línea fx luego de haber asignado valores a las variables **fx.center_x**, **fx.center_y**, **fx.fun_x**, **fx.fun_y**, **fx.scale_x** y **fx.scale_y**.

$$\text{fx.pos}_x = \text{fx.center}_x + \text{fx.fun}_x * \text{fx.scale}_x$$

fx.pos_y: es la coordenada en “y” de la posición final de la Línea fx luego de haber asignado valores a las variables **fx.center_x**, **fx.center_y**, **fx.fun_x**, **fx.fun_y**, **fx.scale_x** y **fx.scale_y**.

$$\text{fx.pos}_y = \text{fx.center}_y + \text{fx.fun}_y * \text{fx.scale}_y$$

Veamos un ejemplo para tener claro lo de la Posición final:

Center in 'X' =	syl.center
Center in 'Y' =	syl.middle
x(s) =	cos(s)
Scale in 'X' =	2
y(s) =	sin(s)
Scale in 'Y' =	4
Pos in 'X' =	fx.pos_x
Pos in 'Y' =	fx.pos_y

Para este ejemplo obtendríamos los siguientes valores:

$$\text{fx.pos_x} = \text{syl.center} + \cos(s) * 2$$

$$\text{fx.pos_y} = \text{syl.middle} + \sin(s) * 4$$

El valor por default de **fx.fun_x** y **fx.fun_y** es 0, teniendo en cuenta esto, para el siguiente ejemplo tenemos:

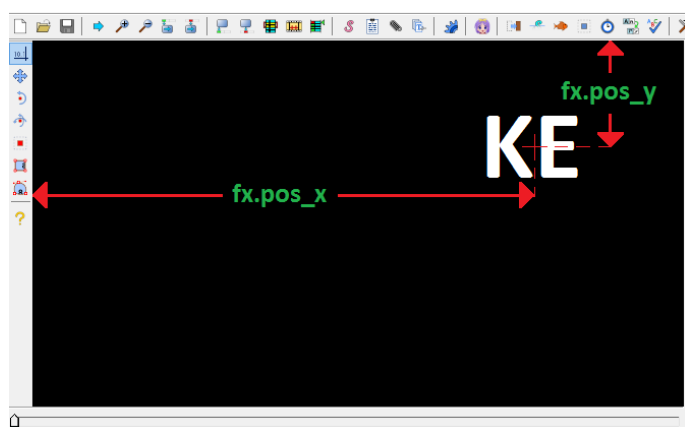
Center in 'X' =	syl.center
Center in 'Y' =	syl.middle
x(s) =	
y(s) =	
Scale in 'X' =	
Scale in 'Y' =	
Pos in 'X' =	fx.pos_x
Pos in 'Y' =	fx.pos_y

$$\text{fx.pos_x} = \text{syl.center} + 0 * 1 = \text{syl.center}$$

$$\text{fx.pos_y} = \text{syl.middle} + 0 * 1 = \text{syl.middle}$$

Los ceros de las anteriores ecuaciones corresponden a los valores por default de **fx.fun_x** y **fx.fun_y**, ya que en la imagen ambas celdas están vacías. Los unos son los valores por default de **fx.scale_x** y **fx.scale_y**, dado que también están vacías sus respectivas celdas de texto.

Entonces concluimos que **fx.pos_x** y **fx.pos_y** son los centros de la posición final de cada una de las Líneas fx:

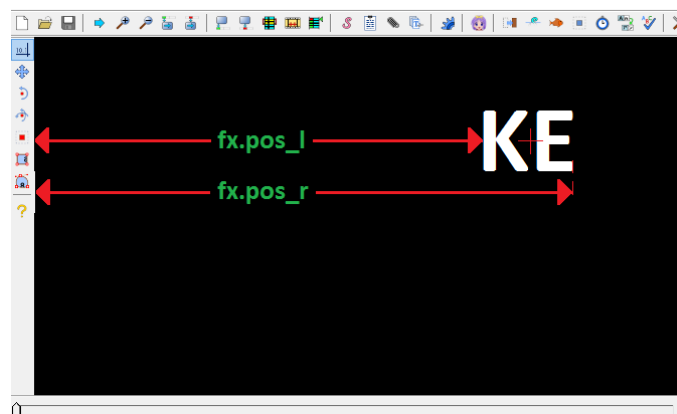


Una vez aclarados los valores de las anteriores dos variables, será más simple entender las cuatro siguientes, ya que dependen directamente de ellas.

fx.pos_l y **fx.pos_r**: (left y right) son la parte izquierda y derecha de **fx.pos_x** respectivamente.

La izquierda y la derecha de **fx.pos_x** dependen del **“Template Type”**, es decir que depende del tipo de plantilla del Efecto. Ejemplo:

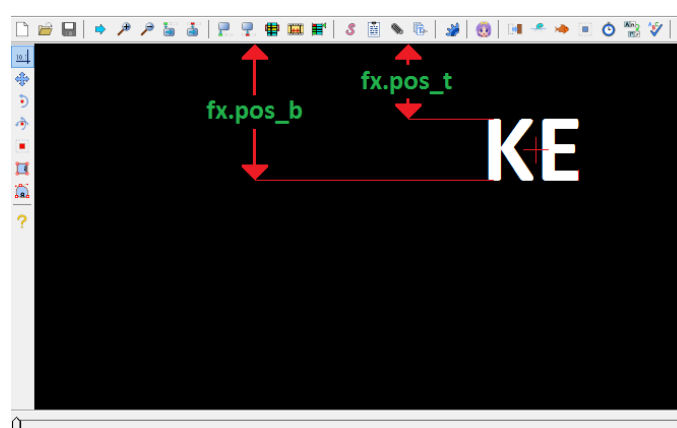
Si la plantilla es tipo **“Syl”** entonces **fx.pos_l** y **fx.pos_r** serán la izquierda y la derecha de la Sílabas, sin importar en dónde esté esa Sílabas:



Para **Template Type “Line”**, entonces **fx.pos_l** y **fx.pos_r** serán la izquierda y la derecha de la Línea de Texto. Lo mismo pasaría para los demás tipos de Plantillas: **“Word”**, **“Furi”**, **“Char”** y las demás.

fx.pos_t y **fx.pos_b**: (top y bottom) son la parte superior e inferior de **fx.pos_y** respectivamente.

Como en el caso de las dos variables anteriores, la parte superior e inferior depende únicamente de la posición final:



fx.align: es la alineación de la Línea fx. Debe ser un valor entero entre 1 y 9, inclusive. El valor por default de esta variable es 5. 5 es la alineación recomendada para aquellos que aún no tienen mucha experiencia a la hora de hacer Efectos Karaoke, pero todas son importantes.

fx.move_x1, fx.move_x2, fx.move_x3, fx.move_x4: son las cuatro posibles coordenadas con respecto al eje “x” de la posición o de la trayectoria del movimiento de la Línea fx. Son las coordenadas en “x” de los tags: \pos, \move, \moves3, \moves4 y \mover.

fx.move_y1, fx.move_y2, fx.move_y3, fx.move_y4: son las cuatro posibles coordenadas con respecto al eje “y” de la posición o de la trayectoria del movimiento de la Línea fx. Son las coordenadas en “y” de los tags: \pos, \move, \moves3, \moves4 y \mover.

Veamos algunos ejemplos de la anteriores ocho variables y de a poco las iremos aclarando:

Pos in 'X' =	<input type="text" value="fx.pos_x - 45"/>	^	v
Pos in 'Y' =	<input type="text" value="fx.pos_y"/>	^	v

fx.move_x1 = fx.pos_x - 45

fx.move_y1 = fx.pos_y

Los valores por default de **fx.move_x1** y **fx.move_y1** son, **fx.pos_x** y **fx.pos_y**, respectivamente. Del ejemplo de la anterior imagen, su resultado en la Línea fx sería el tag \pos, ya que solo hay una coordenada para ambos ejes.

Este ejemplo también retornaría un tag \pos:

Pos in 'X' =	<input type="text"/>	^	v
Pos in 'Y' =	<input type="text" value="fx.pos_y"/>	^	v

fx.move_x1 = fx.pos_x ← por Default

fx.move_y1 = fx.pos_y

Para el caso de que ambas celdas de texto estén vacías, no retornaría ningún tag de posición ni de movimiento:

Pos in 'X' =	<input type="text"/>	^	v
Pos in 'Y' =	<input type="text"/>	^	v

El valor por default de **fx.move_x2**, **fx.move_y2** y todas las superiores, es siempre la variable inmediatamente a cada una de ellas:

Pos in 'X' =	<input type="text"/>	^	v
Pos in 'Y' =	<input type="text" value="fx.pos_y, fx.pos_y + 20"/>	^	v

fx.move_x1 = fx.pos_x ← por Default

fx.move_x2 = fx.move_x1 ← por Default

fx.move_y1 = fx.pos_y

fx.move_y2 = fx.pos_y + 20

El ejemplo anterior daría como resultado un tag \move, ya que habría dos coordenadas para cada eje.

Para el próximo ejemplo, veremos el caso cuando ambos tienen tres coordenadas o al menos una celda de texto tiene tres coordenadas:

Pos in 'X' =	<input type="text" value="fx.pos_x, fx.pos_x - math.random(-30, 70)"/>	^	v
Pos in 'Y' =	<input type="text" value="fx.pos_y, fx.pos_y + 25, fx.pos_y - 25"/>	^	v

fx.move_x1 = fx.pos_x

fx.move_x2 = fx.pos_x - math.random(-30,70)

fx.move_x3 = fx.move_x2 ← por Default

fx.move_y1 = fx.pos_y

fx.move_y2 = fx.pos_y + 25

fx.move_y3 = fx.pos_y - 25

El tag que resultaría de este ejemplo sería un \moves3. Para el caso en el que al menos una de las dos celdas de texto tenga cuatro coordenadas, veamos este ejemplo:

Pos in 'X' =	<input type="text" value="50, 120, -80, syl.center"/>	^	v
Pos in 'Y' =	<input type="text" value="200, 400, 680, syl.middle"/>	^	v

fx.move_x1 = 50

fx.move_x2 = 120

fx.move_x3 = -80

fx.move_x4 = syl.center

fx.move_y1 = 200

fx.move_y2 = 400

fx.move_y3 = 680

fx.move_y4 = syl.middle

Para el caso del ejemplo anterior obtendríamos como tag un `\moves4`, lo que quiere decir que el tag que resulta de la combinación de **Pos in 'X'** y **Pos in 'Y'** depende de cuál de estas dos celdas de texto tenga más coordenadas.

Y para el caso del tag `\mover` hacemos lo siguiente:

Pos in 'X' =	x1, x2, Angle1, Angle2, Radius1, Radius2
Pos in 'Y' =	y1, y2

fx.movet_i, **fx.movet_f**: son los tiempo de inicio y final para los tags de movimiento: `\move`, `\moves3`, `\moves4` y `\mover`:

Times Move =	t1, t2
--------------	--------

“**t1**” representa el tiempo de inicio del movimiento y “**t2**” el tiempo final. Sus valores por default son 0 y **fx.dur**, respectivamente.

Veamos algunos ejemplos:

Pos in 'X' =	x1, x2, x3
Pos in 'Y' =	y1, y2, y3
Times Move =	t1, t2

→ `\moves3(x1, y1, x2, y2, x3, y4, t1, t2)`

El anterior es un ejemplo sencillo de cómo obtener un `\moves3` junto con los parámetros de tiempo incluidos.

Pos in 'X' =	x1
Pos in 'Y' =	y1, y2, y3
Times Move =	

→ `\moves3(x1, y1, x1, y2, x1, y4)`

Este ejemplo ilustra cómo **fx.move_x2** y **fx.move_x3** son **fx.move_x1** por default, y al no haber parámetros de tiempo, entonces no salen en el tag.

Pos in 'X' =	x1, x2
Pos in 'Y' =	y1, y2
Times Move =	t1

→ `\move(x1, y1, x2, y2, t1, fx.dur)`

Y en este otro ejemplo vemos que **fx.movet_f** equivale a **fx.dur** (duración de la Línea fx) por default.

Todo tag de posición o de movimiento que hagamos utilizando estas tres celdas de texto:

Pos in 'X' =	fx.pos_x
Pos in 'Y' =	fx.pos_y
Times Move =	

Se verá reflejado en las Líneas fx:

B I U S fn AB AB AB AB ✓ ⦿ Tiempo ○ Cuadro
[Kara Effector[fx] 3.2: ABC Template \an5\pos(308.34,42)]do

Pero no es la única forma de usar los tags de posición y movimientos, ya que desde **Add Tags** también lo podemos hacer, de hecho, se si hace desde **Add Tags**, entonces cualquier otro tag de posición o de movimiento es anulado:

Pos in 'X' =	fx.pos_x
Pos in 'Y' =	fx.pos_y
Times Move =	
Add Tags:	Add Tags Language: Lua
\move(10,20,30,40)	

El **Kara Effector** detecta de forma automática que hay un tag de movimiento (`\move`) en **Add Tags**, entonces anula el tag `\pos` que se iba a generar por las configuraciones de **Pos in 'X'** y **Pos in 'Y'**:

0	0:00:02.43	0:00:08.16	0:00:05.73	0	0	0
B I U S fn AB AB AB AB ✓ ⦿ Tiempo ○ Cuadro						
[Kara Effector[fx] 3.2: ABC Template \an5\move(10,20,30,40)]Kq						

Si por algún motivo se coloca más de un tag de posición o de movimiento en **Add Tags**, el **Kara Effector** solo tomará en cuenta el último de ellos:

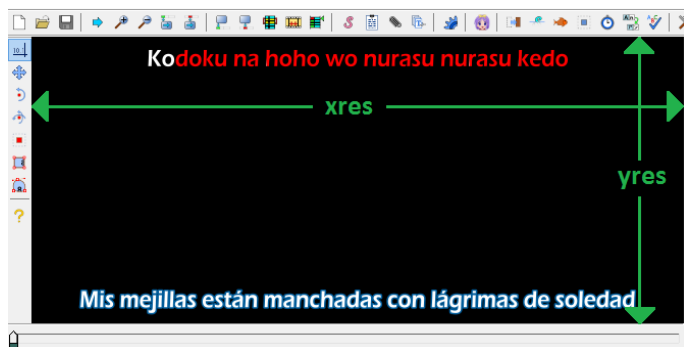
Add Tags:	Add Tags Language: Automation Auto-4
\move(100,200,120,300,0,1000)\pos(\$center,\$middle)	

En este caso solo se toma en cuenta el tag `\pos` ya que es el último y el tag `\move` no saldrá en la Línea de fx.

Con los anteriores ejemplos concluye la explicación de la Librería "fx" pero hay otra serie de variables y valores propios del **Kara Effector** que veremos a continuación:

xres y **yres**: son las dimensiones medidas en pixeles del vídeo que se esté usando al momento de aplicar un Efecto y sus valores por default son 1280 y 720 p.

xres es el ancho del vídeo y **yres**, el alto:



ratio: es la Proporción que usa el **Kara Effector** para que un Efecto funcione exactamente igual sin depender de las dimensiones del vídeo. Al aplicar un Efecto con un vídeo abierto, el valor del **ratio** es: $xres/1280$, para el caso contrario su valor por default es 1.

frame_dur: es la duración medida en milisegundos de cada uno de los cuadros (frames) del vídeo que se esté usando al momento de aplicar un Efecto. Su valor por default es de 40 ms.

line.i: es el Contador numérico de cada una de las Líneas seleccionadas a las que le aplicaremos un Efecto. Es similar a la variable **syl.i**, salvo que esta última es el contador numérico de las Sílabas que contiene una Línea.

line.n: es la cantidad total de Líneas seleccionadas para aplicar un Efecto. Es similar a la variable **syl.n**, salvo que esta última es la cantidad total de Sílabas que contiene una Línea.

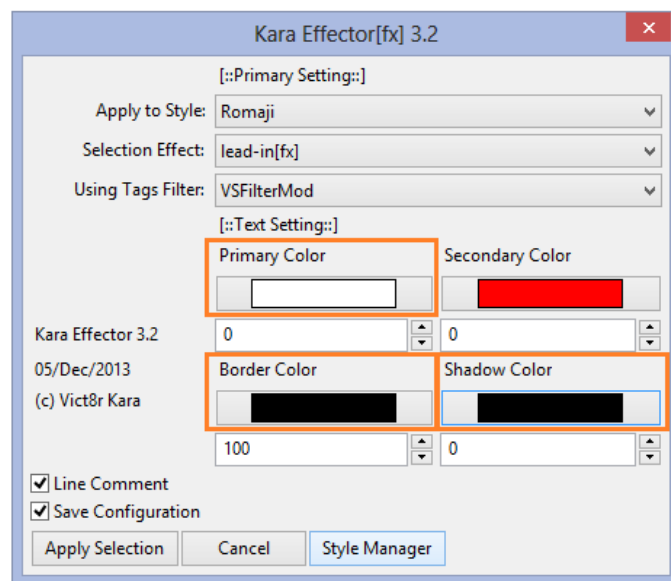
line.index: es el contador de todas las Líneas de Dialogo de un archivo .ass y también puede ser llamado como: **ii**

text.color: son los tres tag de color de la ventana de inicio del Kara Effector. En Lenguaje **Automation Auto-4** sería:

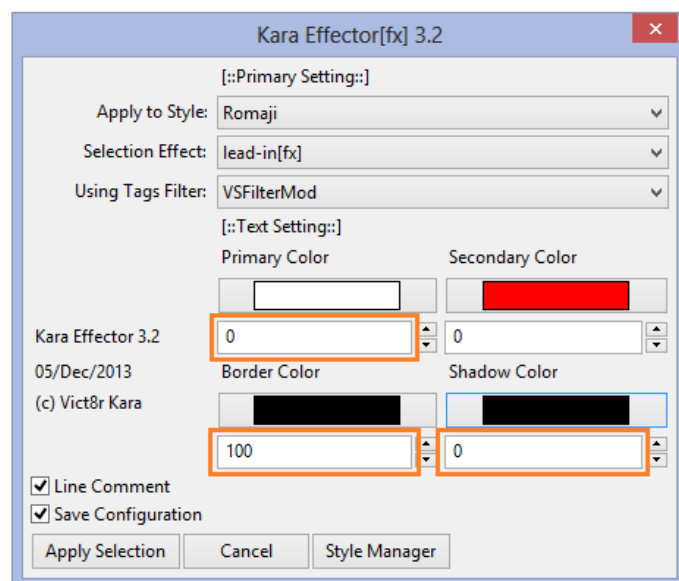
`\1c!text.color1!\3c!text.color3!\4c!text.color4!`

El formato en que se ven depende del filtro seleccionado.

Es decir que **text.color** equivale a estos tres colores:



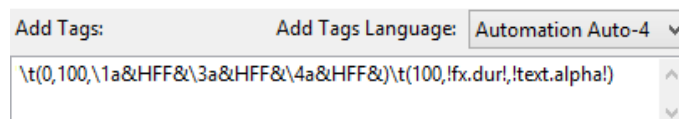
text.alpha: es similar a **text.color**, pero hace referencia a las transparencias de los tres colores anteriormente mencionados:



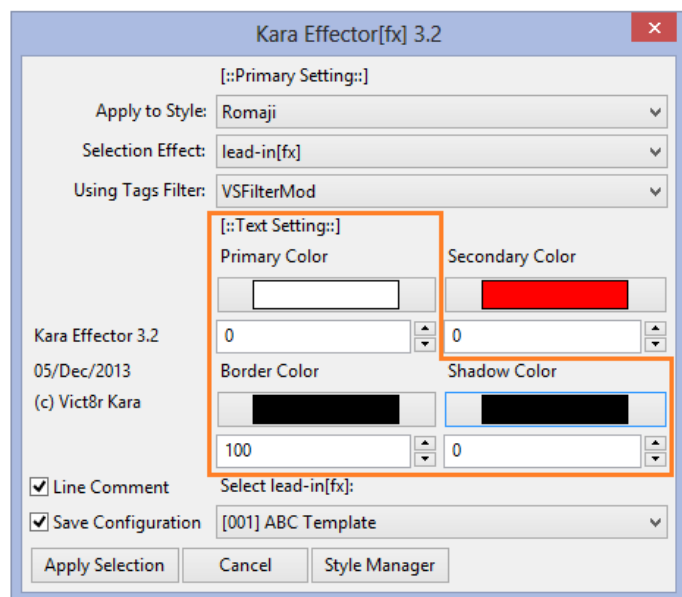
En Lenguaje **Automation Auto-4** sería:

`\1a!text.alpha1!\3a!text.alpha3!\4a!text.alpha4!`

Tanto **text.color** y **text.alpha** pueden servir para volver a las configuraciones de la ventana de inicio luego de alguna transformación de colores y/o de transparencias, ejemplo:

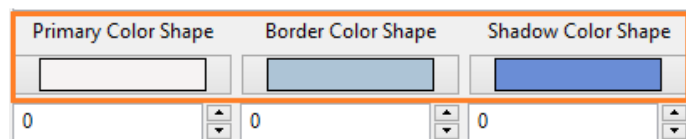


text.style: es la unión de **text.color** y **text.alpha**, o sea que hace referencia a estos seis valores de la ventana de inicio del **Kara Effector**:



text.alpha0: equivale a \alpha&HFF&.

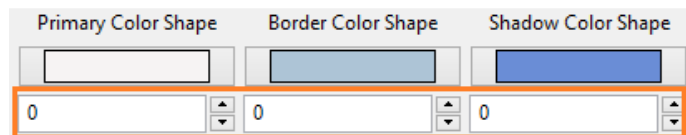
shape.color: son los tres tag de color de las Shapes:



En Lenguaje **Automation Auto-4** sería:

\1c!shape.color1!\3c!shape.color3!\4c!shape.color4!

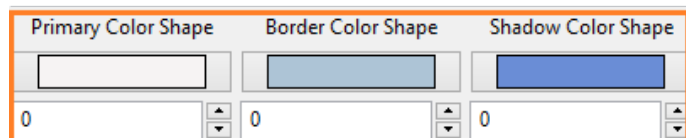
shape.alpha: son los tres tag de transparencia de las Shapes:



En Lenguaje **Automation Auto-4** sería:

\1a!shape.alpha1!\3a!shape.alpha3!\4a!shape.alpha4!

shape.style: es la unión de **shape.text** y **shape.alpha**:



shape.alpha0: equivale a \alpha&HFF&.

module: es la interpolación equidistante de los valores numéricos entre 0 y 1 con respecto al loop de un Efecto.

$$\text{module} = (j - 1) / (\text{maxj} - 1)$$

module1: es la interpolación equidistante de los valores numéricos entre 0 y 1 con respecto a la cantidad de sílabas de las Líneas seleccionadas para un Efecto.

$$\text{module1} = (\text{syl.i} + \text{module} - 1) / \text{syl.n}$$

module2: es la interpolación equidistante de los valores numéricos entre 0 y 1 con respecto a la cantidad de Líneas seleccionadas para un Efecto.

$$\text{module2} = (\text{line.i} + \text{module1} - 1) / \text{line.n}$$

Y hemos llegado al final de las variables de la Librería "fx" sin antes mencionarles que aún hay algunas funciones de la misma que he decidido explicar en futuros Tomos para una mejor comprensión. De a poco vamos revelando los secretos del **Kara Effector** y espero que no se pierdan las próximas entregas.

A medida que avanzamos en los **Tomos**, profundizamos cada vez más en el mundo del **Kara Effector** y contamos con más herramientas para hacer nuestros propios Efectos. No olviden visitarnos en el **Blog Oficial** lo mismo que en los canales de **YouTube** para descargar los nuevos Efectos o si quieren comentar, exponer alguna duda o hacer alguna sugerencia.