Kara Effector 3.2:
Effector Book
Vol. II [Tomo XXXII]

Kara Effector 3.2:

En este **Tomo XXXII** continuaremos viendo más de los Recursos disponibles en el **Kara Effector**, que espero que con la ayuda de esta documentación, le puedan sacar el máximo provecho a la hora de llevar a cabo sus proyectos, no solo karaokes, sino también para la edición de las líneas de subtítulos.

Recursos [KE]:

Recursos [KE]: → Actualización

Función retime - Nuevos Modos [fx]

Esta función modifica los tiempos de inicio y final de las líneas fx generadas por un efecto. La función **retime** es una de las *funciones más usadas del* ***Automation Auto-4*** del **Aegisub**, y en su versión original consta de 10 modos:

1. **preline**
2. **line**
3. **postline**
4. **presyl**
5. **start2syl**
6. **syl**
7. **syl2end**
8. **postsyl**
9. **sylpct**
10. **set or abs**

En el **KE**, al haber más opciones para aplicar un efecto, los modos aumentaron en 29:

Función Retime - 29 Modos iniciales				
LINE	WORD	SYL	FURI	CHAR
preline	preword	presyl	prefuri	prechar
	start2word	start2syl	start2furi	start2char
line	word	syl	furi	char
	word2end	syl2end	furi2end	char2end
postline	postword	postsyl	postfuri	postchar
linepct	wordpct	sylpct	furipct	charpct
set or abs				

El **Kara Effector**, a partir de la versión **3.2.8** está provisto con 4 nuevos modos más:

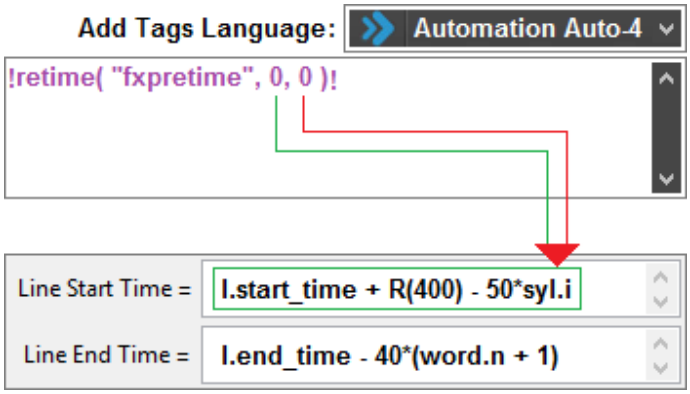
1. **fxpretime**
2. **fxtime**
3. **fxposttime**
4. **fxpct**

Modo: fxpretime

Retimea los tiempos de las líneas fx generadas por un efecto, con referencia al tiempo de inicio registrado en la celda "**Line Start Time**" de la Ventana de Modificación del **Kara Efector**.

Ambos ceros en la función equivalen al tiempo de inicio del efecto, en la celda "**Line Start Time**".

Ejemplo:



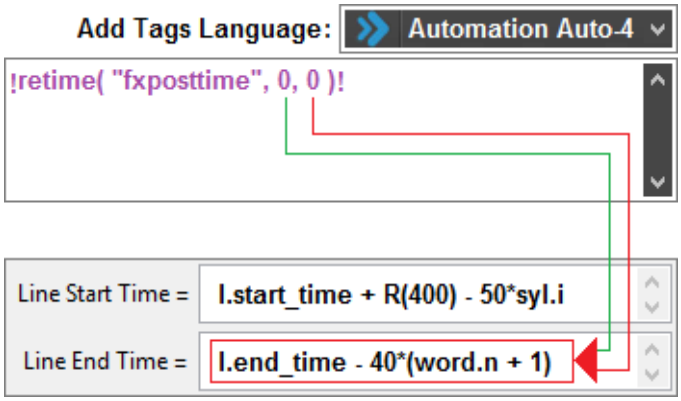
Este modo de la función es similar al modo "**preline**", pero con la diferencia de que no toma como referencia al tiempo de inicio de la línea karaoke, sino el de la línea fx.

Modo: fxposttime

Retimea los tiempos de las líneas fx generadas por un efecto, con referencia al tiempo de inicio registrado en la celda "**Line Start Time**" de la Ventana de Modificación del **Kara Efector**.

Ambos ceros en la función equivalen al tiempo final del efecto, en la celda "**Line End Time**".

Ejemplo:



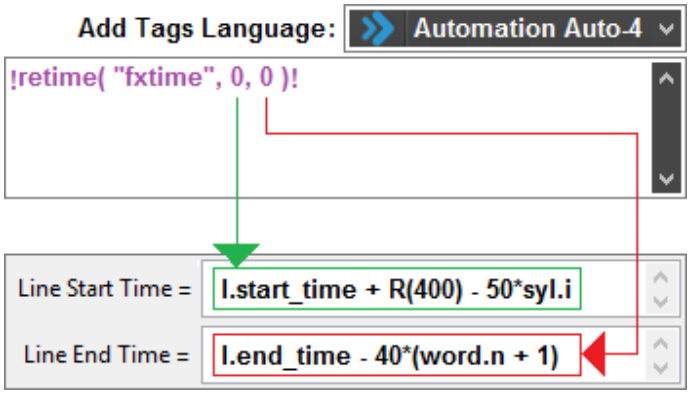
Este modo de la función es similar al modo "**postline**", pero con la diferencia de que no toma como referencia al tiempo final de la línea karaoke, sino el de la línea fx.

Modo: fxtime

Retimea los tiempos de las líneas fx generadas por un efecto, con referencia al tiempo de inicio registrado en la celda "**Line Start Time**" de la Ventana de Modificación del **Kara Efector**.

El primer cero en la función equivale al tiempo de inicio del efecto, en la celda "**Line Start Time**" y el segundo, al tiempo final del efecto, en la celda "**Line End Time**".

Ejemplo:



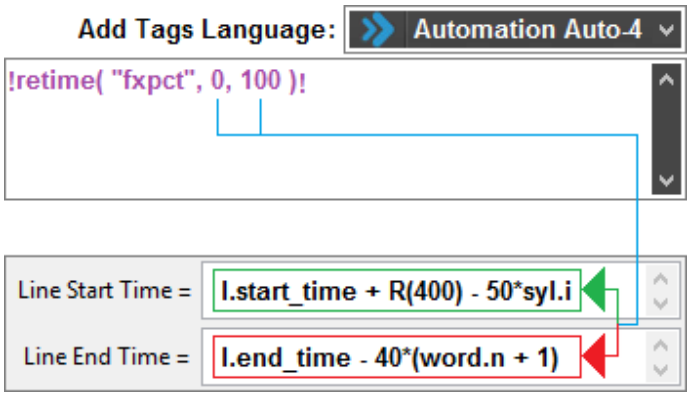
Este modo de la función es similar al modo "**line**", pero con la diferencia de que no toma como referencia a los tiempos de la línea karaoke, sino los tiempos de la línea fx.

Modo: fxpct

Retimea los tiempos de las líneas fx generadas por un efecto, con referencia al tiempo de inicio registrado en la celda "**Line Start Time**" de la Ventana de Modificación del **Kara Efector**.

El primer cero en la función equivale al 0% de la duración total de la línea fx (**fx.dur**). El 100 equivale al 100% de la duración total de la línea fx.

Ejemplo:



Este modo de la función es similar al modo "**linepct**", pero con la diferencia de que no toma como referencia a los tiempos de la línea karaoke, sino los tiempos de la línea fx.

Los cuatro ejemplos en los anteriores cuatro modos de la función **retime** están en lenguaje **Automation Auto-4**, lo que no quiere decir que esta función solo se puede usar en dicho lenguaje, ya que también la podemos usar en **LUA**.

Vistos estos cuatro nuevos modos de la función **retime**, el total de los modos es de 33, lo que hace que la tabla final de ellos quede de la siguiente manera:

Función Retime [KE] - 33 Modos totales				
LINE	WORD	SYL	FURI	CHAR
preline	preword	presyl	prefuri	prechar
	start2word	start2syl	start2furi	start2char
line	word	syl	furi	char
	word2end	syl2end	furi2end	char2end
postline	postword	postsyl	postfuri	postchar
linepct	wordpct	sylpct	furipct	charpct
set or abs				
FX	fxposttime	fxpct	fxtime	fxpretime

Recursos [KE]: → Actualización

text.bezier(Shape, mode, Accel, Offset_time)

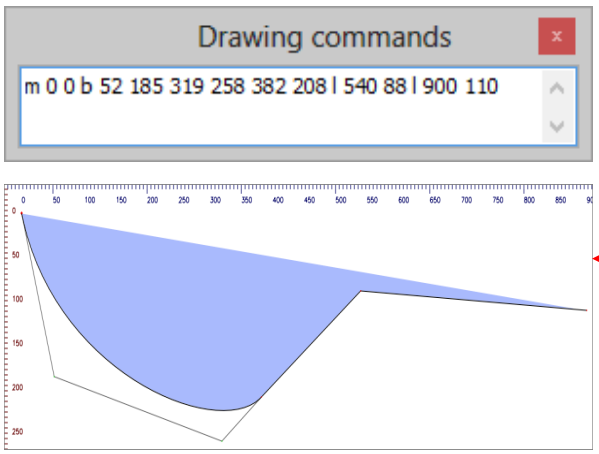
Esta función hace que el texto adopte la forma de una **curva Bézier**, de una **shape** ingresada en el primer parámetro o de un clip dibujado en las líneas del script.

Esta función hace parte de la librería **text** y está disponible para la versión **3.2.9.4** o superior del **Kara Effector**, razón por la cual está en la sesión de recursos de actualización del **KE**.

El parámetro **Shape** tiene tres diferentes opciones, y cada una de ella con diversas características que a continuación veremos:

1. el código convencional de una **shape**.

Ejemplo:

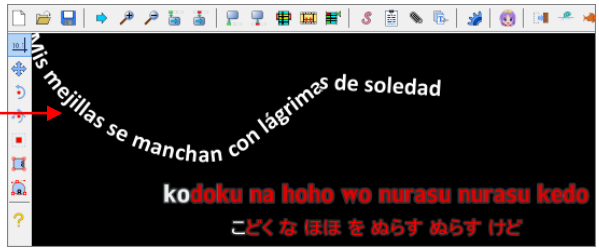


Template Type [fx]: Char

Add Tags Language: Lua

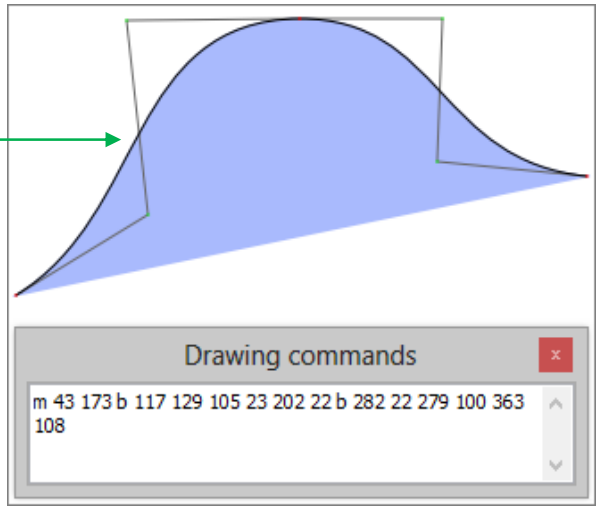
```
text.bezier( "m 0 0 b 52 185 319 258 382 208
l 540 88 l 900 110 " )
```

Y al aplicar:

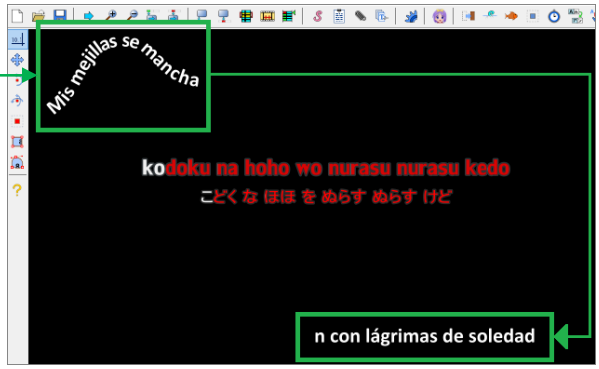


No importa la opción que elijamos en el parámetro **Shape**, debemos cerciorarnos que ésta mida igual o más que la longitud de la línea del script, o sea que su longitud sea mayor o igual que **line.width**.

Ejemplo:



Si ingresamos una shape que mida menos que la longitud de la línea a la que le apliquemos un efecto, el texto quedará en su posición por default como ni no hubiésemos usado la función:



Ejemplo:

Podemos usar alguna de las Shapes que traen por default el **KE** o una modificación de las mismas:

Variables:

```
mi_shape = shape.centerpos( shape.size(
    shape.circle, 340 ), 640, 360 )
```

una shape por default de KE

Y en **Add Tags** ponemos:

Template Type [fx]: Char

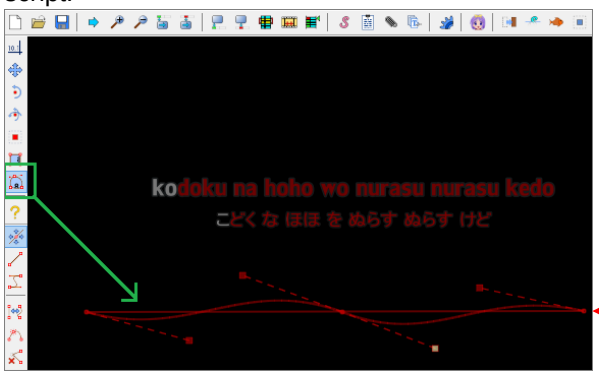
Add Tags Language: Lua

```
text.bezier( mi_shape )
```

2. el parámetro **Shape** puede ser, también, un clip dibujado en una o más líneas del script. Las líneas que no tengan un clip dentro de ellas, no se verán afectadas por la función.

Ejemplo:

Dibujamos un clip en una o más de las líneas del script:

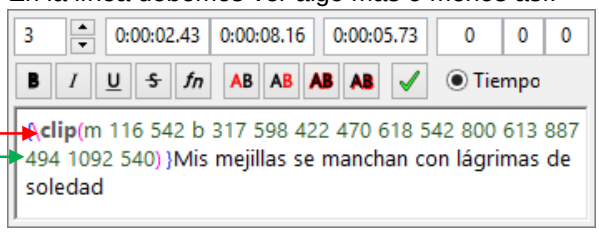


Y en **Add Tags** ponemos a alguna de estas dos opciones:

A. text.bezier(nil)
B. text.bezier( )

O sea que: **text.bezier(nil) = text.bezier( )**

En la línea debemos ver algo más o menos así:



Y al aplicar:

Y aquellas líneas en las que no dibujamos un clip, no se verá afectadas por la función.
La tercera opción del parámetro **Shape** es cuando es una **tabla** con mínimo 2 Shapes y máximo 4, y la veremos más adelante con el fin de ver primero el resto de los parámetros de la función.

El parámetro **mode** es un número entero entre 1 y 6, lo que nos da seis opciones diferentes de usar la función. Cada uno de ellos es independiente del parámetro **Shape**, es decir que no importa la manera en que le ingresemos la **shape** a la función:

mode = 1

Justifica el texto en el centro de la longitud de la shape:

Ejemplo:

Usamos como ejemplo el clip dibujado del ejemplo anterior, pero con cualquier **shape** ingresada, el **mode =1** hará que el texto quede justificado en el centro de la longitud de la misma:

Template Type [fx]: Char

Add Tags Language: Lua

```
text.bezier( nil, 1 )
```

Al aplicar, el texto (en este caso, los caracteres de la línea) adopta la forma del clip dibujado previamente y como lo había mencionado antes, éste queda justificado respecto al centro de la longitud total de la **shape**:

mode = 2

Es el modo por default de la función, y justifica el texto a la izquierda de la **shape**, o dicho de otra manera, coloca al texto desde el inicio de la **shape**:

Ejemplo:

Template Type [fx]: Char

Add Tags Language: Lua

```
text.bezier( nil, 2 )
```

mode = 3

Justifica el texto a la derecha de la **shape**, o dicho de otra manera, coloca al texto desde el final de la **shape**:

Ejemplo:

Template Type [fx]: Char

Add Tags Language: Lua

```
text.bezier( nil, 3 )
```

mode = 4

Hace que el texto se extienda a lo largo de la longitud total de la **shape**:

Ejemplo:

Template Type [fx]: Char

Add Tags Language: Lua

```
text.bezier( nil, 4 )
```

Los tres primeros valores del parámetro **mode** (1, 2 y 3) tienen una ventaja más que podemos usarla con la celda "**Actor**" del **Aegisub**, es un número que indica la cantidad de pixeles que necesitemos desplazar el texto una vez que éste adopta la forma de la **shape**.

No importa que en la celda "**Actor**" de una o más líneas del script ya hayamos escrito algo, lo único que debemos hacer es poner el valor que necesitemos al lado de lo que ahí ponga:

- Si la celda "**Actor**" está vacía solo ponemos el valor que necesitemos en la línea:

Ejemplo:

Estilo	Actor	Efecto	Texto
Romaji			✱ko✱do✱ku ✱na ✱ho✱ho ✱wo ✱
Romaji			✱yo✱a✱ke ✱no ✱ke✱ha✱i ✱ga ✱
Romaji			✱wa✱ta✱shi ✱wo ✱so✱ra ✱e ✱r
Romaji			✱ki✱bo✱u ✱ga ✱ka✱na✱ta ✱de
Romaji			✱ma✱yo✱i ✱na✱ga✱ra ✱mo ✱ki

Estilo	Actor	Efecto	Texto
Romaji			✱ko✱do✱ku ✱na ✱ho✱ho ✱wo ✱
Romaji	50		✱yo✱a✱ke ✱no ✱ke✱ha✱i ✱ga ✱
Romaji			✱wa✱ta✱shi ✱wo ✱so✱ra ✱e ✱r
Romaji			✱ki✱bo✱u ✱ga ✱ka✱na✱ta ✱de
Romaji			✱ma✱yo✱i ✱na✱ga✱ra ✱mo ✱ki

- Si en la celda "**Actor**" ya pone algo, solo debemos agregar un espacio seguido del valor que vamos a desplazar el texto sobre el perímetro de la shape:

Ejemplo:

Estilo	Actor	Efecto	Texto
Romaji			✱ko✱do✱ku ✱na ✱ho✱ho ✱wo ✱
Romaji	50		✱yo✱a✱ke ✱no ✱ke✱ha✱i ✱ga ✱
Romaji	demo		✱wa✱ta✱shi ✱wo ✱so✱ra ✱e ✱r
Romaji	intro		✱ki✱bo✱u ✱ga ✱ka✱na✱ta ✱de
Romaji			✱ma✱yo✱i ✱na✱ga✱ra ✱mo ✱ki

Estilo	Actor	Efecto	Texto
Romaji			✱ko✱do✱ku ✱na ✱ho✱ho ✱
Romaji	50		✱yo✱a✱ke ✱no ✱ke✱ha✱i ✱
Romaji	demo -86		✱wa✱ta✱shi ✱wo ✱so✱ra ✱
Romaji	intro		✱ki✱bo✱u ✱ga ✱ka✱na✱ta ✱
Romaji			✱ma✱yo✱i ✱na✱ga✱ra ✱mo

De cualquiera de las dos formas, la función reconoce el valor numérico que haya en la celda "**Actor**" y lo tomará como la distancia medida en pixeles para desplazar el texto según queramos.

En el **mode = 1**, dado que el texto queda justificado en el centro de la longitud total del perímetro de la shape, si el valor puesto en la celda "**Actor**" es positivo, lo desplazará hacia la derecha, en caso contrario, hacia la izquierda.

Ejemplo:

Si no hay un valor numérico en la celda "**Actor**", el valor del desplazamiento se asume como cero (0).

Hiragana			✱も✱と✱め ✱あ✱う ✱こ✱こ✱ろ ✱そ✱れ
English			✱Mis mejillas se manchan con lágrimas d
English			Pero puedo sentir la llegada del amanece

Es decir, que para **mode = 1**, el texto no se desplazará ni a la izquierda ni a la derecha.

Si ponemos un valor positivo, se desplazará a la derecha:

Ejemplo:

Add Tags Language: Lua

```
text.bezier( nil, 1 )
```

Hiragana			✱も✱と✱め ✱あ✱う ✱こ✱こ✱ろ ✱
English	80		✱Mis mejillas se manchan con lágr
English			Pero puedo sentir la llegada del ar

Y para valores negativos, se desplazará a la izquierda:

Ejemplo:

Add Tags Language: Lua

```
text.bezier( nil, 1 )
```

Hiragana			✱も✱と✱め ✱あ✱う ✱こ✱こ✱ろ ✱
English	-50		✱Mis mejillas se manchan con lágr
English			Pero puedo sentir la llegada del ar

Entonces, con **mode = 1**, sí se tiene en cuenta el signo de la cantidad numérica que pongamos en la celda "**Actor**", todo depende de hacia dónde queramos desplazar al texto.

Para **mode = 2** y **mode = 3**, el valor numérico en la celda "**Actor**" debe positivo.

Ejemplo:

En **mode = 2**, el texto se desplazará tomando como punto de partida el inicio de la **shape**:

Add Tags Language: Lua

```
text.bezier( nil, 2 )
```

Hiragana			✱も✱と✱め ✱あ✱う ✱こ✱こ✱ろ ✱
English	100		✱Mis mejillas se manchan con lágr
English			Pero puedo sentir la llegada del ar

Ejemplo:

En **mode = 3**, el texto se desplazará tomando como punto de partida el final de la **shape**:

Add Tags Language: Lua

```
text.bezier( nil, 3 )
```

Hiragana			✱も✱と✱め ✱あ✱う ✱こ✱こ✱ろ ✱
English	125		✱Mis mejillas se manchan con lágr
English			Pero puedo sentir la llegada del ar

Continuaremos en el **Tomo XXXIII** con todo el resto de la documentación de la función **text.bezier**

Es todo por ahora para el **Tomo XXXII**. Intenten poner en práctica todos los ejemplos vistos y no olviden descargar la última actualización disponible del **Kara Effector 3.2** y visitarnos en el **Blog Oficial**, lo mismo que en los canales de **YouTube** para descargar los nuevos Efectos o dejar algún comentario. Pueden visitarnos y dejar su comentario en nuestra página de **Facebook**:

- www.karaeffector.blogspot.com
- www.facebook.com/karaeffector
- www.youtube.com/user/victor8607
- www.youtube.com/user/NatsuoKE
- www.youtube.com/user/karalaura2012