

Recursos [KE]

He denominado al próximo tema como **Recursos** del **KE**, y son una serie de funciones, variables y múltiples valores y herramientas extras, con las que también podemos contar para ayudarnos en nuestros proyectos.

set_temp(ref, val) :

Esta función es una creación del maestro [Pyointa](#), cuya finalidad principal era el poder almacenar el valor de una variable definida con valor aleatorio, para hacer efectos con templates en **Automation Auto-4**. La función almacena a una variable en una **tabla** específica, para ser usada posteriormente. La **tabla** en donde son almacenadas las variables se llama:

temp = { }

El parámetro “**ref**” es el nombre de la variable que vamos a crear y el parámetro “**val**” es el valor que le daremos a dicha variable. El argumento “**ref**” debe ir entre comillas, ya sean dobles o sencillas.

Ejemplo:

```
set_temp( "sizexy", R(80,120) )  
→ temp.sizexy = R(80,120)  
→ temp = { sizexy = R(80,120) }
```

Ejemplo:

Add Tags Language: Automation Auto-4

```
\fscx!set_temp( "xy", R(80,120) )!\fscy!temp.xy!
```

Ejemplo:

Add Tags Language: Lua

```
format("\fscx%s\fscy%s", set_temp("xy", R(80,120)),  
temp.xy)
```

Esta función es equivalente a declarar una función en la celda de texto “**Variable**” y luego llamarla la cantidad de veces que sea necesaria. Este hecho hace que la función **set_temp** sea prácticamente obsoleta en el **Kara Effector**, pero aun así está incluida dentro de él, porque hay muchos que la usan y ya están familiarizados con ella.

remember(ref, val) :

Es la función equivalente a la función **set_temp** que **Aegisub** estrenó en su versión **2.1.9** y la finalidad de esta es exactamente la misma, con la ligera diferencia de que ésta almacena las variables en una **tabla** llamada:

```
recall = { }
```

Ejemplo:

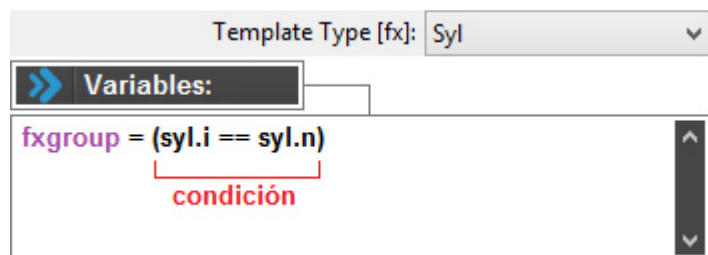
```
remember( "angle", R(360) )  
→ recall.angle = R(360)  
→ recall = { angle = R(360)}
```

fxgroup = false or true :

Es una variable booleana a la que solo le podemos asignar los valores de falso o verdadero (false o true), usada en la celda de texto “**Variables**”.

Esta variable es una función original del **Aegisub** que ha sido adaptada al **KE**, que restringe por medio de una o más condiciones dadas, los objetos a los cuales se les aplicará un efecto seleccionado.

Ejemplo:



La condición con la que definimos a la variable **fxgroup** es de igualdad, que es una condición de comparación, y solo es verdadera cuando **syl.i** es igual a la última sílaba de cada línea seleccionada para aplicar el efecto.

Entonces, cualquiera que haya sido el efecto aplicado, éste solo se aplicará a la última sílaba y no tendrá en cuenta al resto de ellas.

Recordemos las condiciones de comparación:

- == → igual a
- ~= → diferente a
- > → mayor que
- < → menor que
- >= → mayor o igual que
- <= → menor o igual que

También tenemos a la disyunción (**or**) y la conjunción (**and**), que también arrojan un valor booleano entre falso (false) o verdadero (true), al comparar el valor de dos condiciones:

Disyunción (**or**):

- **true or true** → **true**
- **true or false** → **true**
- **false or true** → **true**
- **false or false** → **false**

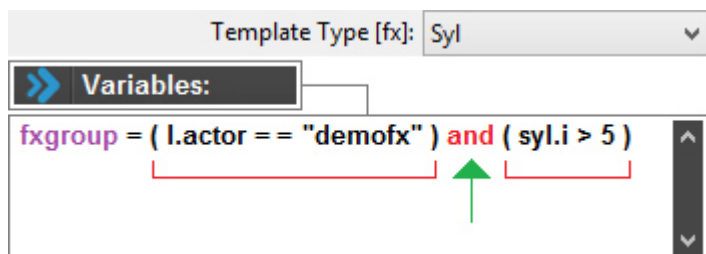
Conjunción (**and**):

- **true and true** → **true**
- **true and false** → **false**
- **false and true** → **false**
- **false and false** → **false**

Ejemplo:

Estilo	Actor	Texto
Romaji		*ko*do*ku *na *ho*ho *wo *nu*ra*su *nu*
Romaji	demofx	*yo*a*ke *no *ke*ha*i *ga *shi*zu*ka *ni
Romaji		*wa*ta*shi *wo *so*ra *e *ma*ne*ku *yo
Romaji		*ki*bo*ku *ga *ka*na*ta *de *ma*te*te*ru
Romaji		*ma*yo*i *na*ga*ra *mo *ki*mi *wo *sa*
Romaji		*su*re*chi*ga*ku *i*shi*ki *te *ga *fu*re*

Para este ejemplo, ponemos un nombre cualquiera a una o más líneas del script (no todas para que se note el fxgroup) en la celda “actor” del **Aegisub**, como se nota en la imagen anterior.



La primera condición es verdadera (true) solo para aquellas líneas que en la celda “actor” ponga: demofx

La segunda condición es verdadera (true) solo cuando las sílabas de las líneas son mayores que 5.

Y como las dos condiciones están relacionadas con una Conjunción (**and**), la variable **fxgroup** será verdadera solo cuando las dos condiciones sean verdadera. Es decir que al aplicar un efecto, éste solo tomará en cuenta a las sílabas mayores que 5 que pertenezcan a las líneas que tengan como actor a “demofx”:



El rectángulo naranja resalta el hecho que el efecto aplicado no tuvo en cuenta a las sílabas “yo“, “a“, “ke“, “no” y “ke“, porque a pesar de que éstas pertenecen a una línea con actor “demofx”, no cumplen con la segunda condición de que el número de su posición sea mayor que 5.

Variables Dólar [Automation Auto – 4] :

Son una serie de variables inspiradas en las variables Dólar del **Automation Auto-4** del **Aegisub**, y sirven para acceder a valores concernientes a las líneas y a las sílabas de las líneas de texto, pero acá en el **KE** he querido ampliar ese listado de variables para que abarquen a todos los modos de efectos dependiendo del **Template Type**.

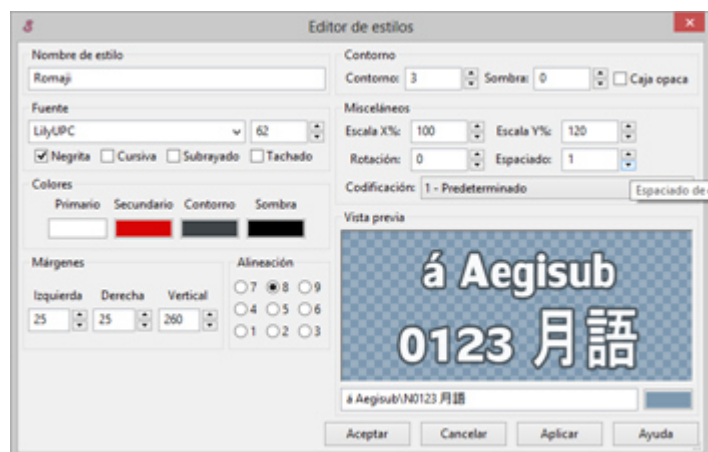
La forma en la que el **KE** reconoce a dichas variables es cambiando el lenguaje de **Add Tags**. Ejemplo:



El primer grupo de variables dólar son las referentes a los estilos de cada una de las líneas del script:

variables Dólar de Estilos			
\$layer	\$margin_b	\$fontsize	\$color4
\$style	\$scale_x	\$fontname	\$alpha1
\$actor	\$scale_y	\$spacing	\$alpha2
\$margin_l	\$angle	\$color1	\$alpha3
\$margin_r	\$outline	\$color2	\$alpha4
\$margin_v	\$shadow	\$color3	\$align
\$margin_t			

\$layer, **\$style** y **\$actor** son variables de las líneas del script y el resto son las variables de todos los valores de los estilos de las líneas:



Luego tenemos las variables de tiempo:

variables Dólar de Tiempo			
\$lstart	\$wstart	\$sstart	\$fstart
\$cstart	\$fxstart	\$start	
\$lend	\$wend	\$send	\$fend
\$cend	\$fxend	\$end	
\$lmid	\$wmid	\$smid	\$fmid
\$cmid	\$fxmid	\$mid	
\$lkdur	\$wkdur	\$skdur	\$fkdur
\$ckdur	\$fxkdur	\$kdur	
\$ldur	\$wdur	\$sdur	\$fdur
\$cdur	\$fxdur	\$dur	

- **\$lstart:** tiempo de inicio de la línea medido en ms, relativo al cero absoluto del vídeo.
- **\$wstar:** tiempo de inicio de cada palabra medido en ms, relativo al tiempo de inicio de la línea a la que pertenece.
- **\$sstart:** tiempo de inicio de cada sílaba medido en ms, relativo al tiempo de inicio de la línea a la que pertenece.
- **\$fstart:** tiempo de inicio de cada **furigana** medido en ms, relativo al tiempo de inicio de la línea a la que pertenece.
- **\$cstart:** tiempo de inicio de cada carácter medido en ms, relativo al tiempo de inicio de la línea a la que pertenece.
- **\$fxstart:** es el tiempo de inicio del objeto karaoke por default dependiendo del **Template Type**. O sea que, si aplicamos un karaoke con un modo Word, la variable **\$fxstart** tomará el valor de **\$wstart**.
- **\$start:** esta variable tiene el mismo valor de **\$fxstart**, es decir que ambas variables son la forma general de obtener el tiempo de inicio del objeto karaoke, dependiendo del **Template Type**.

Estas dos últimas variables las vemos en un recuadro rojo en la tabla de las variables, este recuadro indica las variables generales de cada categoría de variables.

En esta misma categoría de variables dólar de tiempo están las de tiempo final y tiempo medio y de duración total de los objetos karaokes:

- **\$lend:** tiempo final de la línea. Es equivalente a **Lend_time**
- **\$wmid:** tiempo medio de la palabra. Es equivalente a **word.mid_time**
- **\$skdur:** duración de la sílaba en centésimas de segundos. Es equivalente a **syl.dur / 10**
- **\$cdur:** duración del caracter medido en ms. Es equivalente a **char.dur**

Ejemplo:

Template Type [fx]: Word

Add Tags Language: Automation Auto-4

\lt(0,\$wdur,\blur4)

variables Dólar de Contadores			
\$li	\$wi	\$si	\$fi
\$ci	\$fxi	\$i	
\$linen	\$wordn	\$slyn	\$furin
\$charn	\$fxn	\$n	

Las variables de contadores nos dan dos valores, el índice de la posición y la cantidad de objetos karaoke.

- **\$li:** es el contador de las líneas seleccionadas para aplicarle un efecto, equivalente a **line.i**
- **\$wi:** es el contador de las palabras por línea, y es equivalente a **word.i**
- **\$si:** es el contador de las sílabas por línea, y es equivalente a **syl.i**
- **\$fi:** es el contador de los furiganas por línea, y es equivalente a **furi.i**
- **\$ci:** es el contador de los caracteres por línea, y es equivalente a **char.i**
- **\$fxi:** es el contador de los objetos karaokes en un efecto, dependiendo del **Template Type**.
- **\$i:** es lo mismo que **\$fxi**

El resto de las variables de esa categoría indican la cantidad de objetos karaokes, en el caso de **\$linen**, ésta indica la cantidad total de líneas seleccionadas para aplicar un efecto, y las demás indican la cantidad de dicho objeto karaoke en cada línea seleccionada. Ejemplo:

- **\$slyn:** indica la cantidad total de sílabas en cada línea, de las seleccionadas para un efecto.

variables Dólar de Distancias Notables			
\$lleft	\$wleft	\$sleft	\$fleft
\$clef	\$fxleft	\$left	
\$lright	\$wright	\$sright	\$fright
\$cright	\$fxright	\$right	
\$ltop	\$wtop	\$stop	\$ftop
\$ctop	\$fxtop	\$top	
\$lbottom	\$wbottom	\$sbottom	\$fbottom
\$cbottom	\$fxbottom	\$bottom	
\$lcenter	\$wcenter	\$scenter	\$fcenter
\$ccenter	\$fxcenter	\$center	
\$lx	\$wx	\$sx	\$fx
\$cx	\$fxx	\$x	
\$lmiddle	\$wmiddle	\$smiddle	\$fmiddle
\$cmiddle	\$fxmiddle	\$middle	
\$ly	\$wy	\$sy	\$fy
\$cy	\$fxy	\$y	

Esta categoría es la de las variables de distancias notables de los objetos karaokes. Son seis diferentes:

- **left (izquierda)**
- **center (centro)**
- **right (derecha)**
- **top (superior)**
- **middle (medio)**
- **bottom (inferior)**

También tenemos dos equivalencias respectivas para las variables de centro y medio, que son \$x y \$y para cada uno de los objetos karaokes. Ejemplo

- **\$wcenter = \$wx**

variables Dólar de Medidas Notables			
\$lwidth	\$wwidth	\$swidth	\$fwidth
\$cwidth	\$fxwidth	\$width	
\$lheight	\$wheight	\$sheight	\$fheight
\$cheight	\$fxheight	\$height	

Esta última categoría de variables dólar, es de las medidas notables de los objetos karaoke. Dichos valores vienen medidos en pixeles y equivalen al ancho y al alto.

- **\$lwidth = l.width**
- **\$wwidth = word.width**
- **\$swidth = syl.width**
- **\$fwidth = furi.width**
- **\$cwidth = char.width**
- **\$fxwidth = \$width**
- **\$lheight = l. height**
- **\$wheight = word. height**
- **\$sheight = syl. height**
- **\$fheight = furi. Height**
- **\$cheight = char. height**
- **\$fxheight = \$height**

Podemos concluir que no hay una variable dólar de la cual no hayamos conocido su valor con anterioridad, pero como algunos de los usuarios del **KE** antes hacían efectos con los templates del **Aegisub**, entonces están más familiarizados con este tipo de notación de las variables y se les facilita seguir creando efectos en el **Kara Effector** con el lenguaje **Automation Auto-4**

Estas variables son algunas de las herramientas habilitadas para servirnos de apoyo a la hora de hacer efectos en lenguaje **Automation Auto-4**, pero no son las únicas.

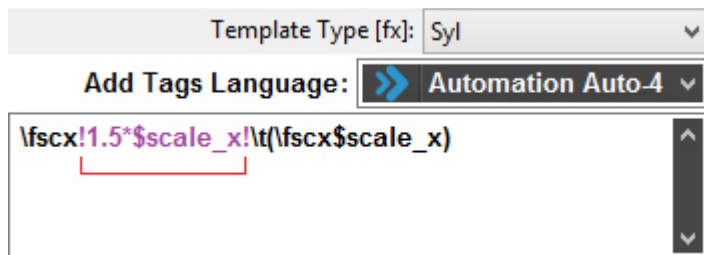
Signos de Admiración [!!] :

El uso de los signos de admiración es heredado de los efectos hechos en el **Aegisub** con **Automation Auto-4** y lo que hacen es posibilitar el uso de cualquier operación matemática y el poder aplicar una o más funciones dentro de ellos.

Entonces, para empezar a usar los signos de admiración como una herramienta más del **KE**, debemos cambiar el lenguaje de **Add Tags** a **Automation Auto-4**:

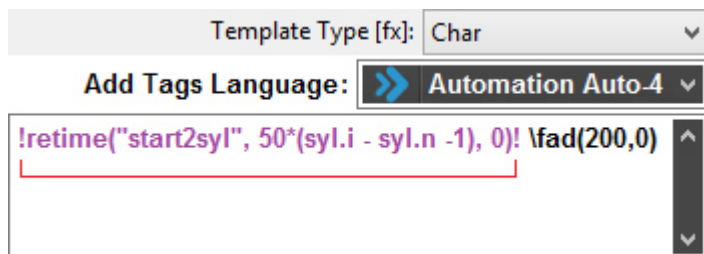
Ejemplo:

Los signos de admiración hacen posible la multiplicación llevada a cabo entre 1.5 y **\$scale_x**:



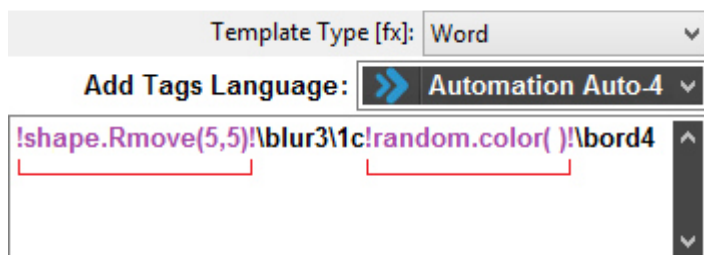
Ejemplo:

Los signos de admiración hacen posible, para este caso, llamar a la función **retime** y modificar los tiempos de inicio y final del efecto aplicado:



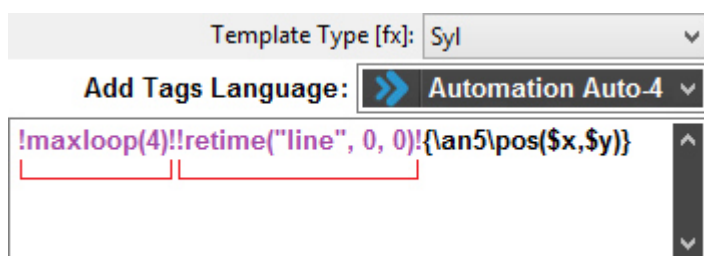
Ejemplo:

Llamar dos o más funciones independientes:



Ejemplo:

Llamar funciones consecutivas:

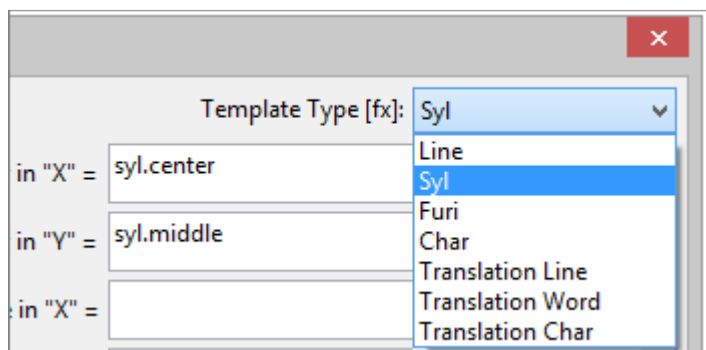


Podemos afirmar, en pocas palabras, que los signos de admiración para el **Automation Auto-4** son equivalentes a la función **string.format** para el lenguaje **LUA**, ya que ambos nos dan la opción de hacer operaciones y agregar funciones a los tags de nuestros efectos.

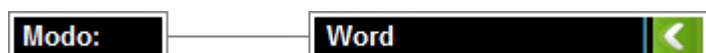
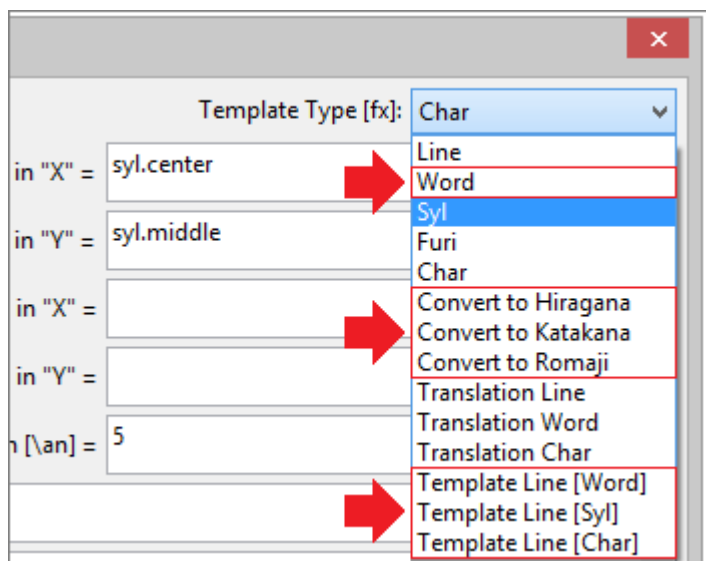
Nuevas Opciones de Template Type [Modos fx] :

Desde la primera versión del **KE**, las opciones de los modos de los efectos (**Template Type**) han aumentado, lo que ha hecho que las posibilidades para nuestros efectos también lo hagan.

En un principio, el **Kara Effector** constaba de solo siete opciones de efectos en el **Template Type**, 4 dedicados a las líneas de karaoke y 3 para las de traducción, estas opciones eran: **Line**, **Syl**, **Furi**, **Char**, **Translation Line**, **Translation Word** y **Translation Char**.



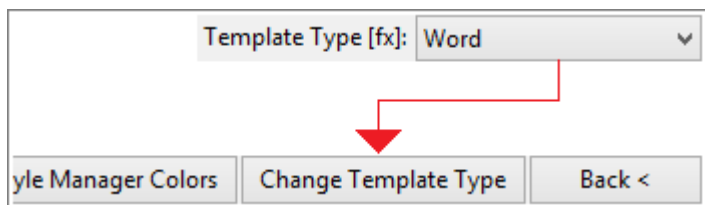
Y en la actualidad (**KE versión 3.2.9.4 en adelante**) tenemos 7 nuevos modos más, lo que nos da un total de 14, 8 para las líneas de karaoke y otras 6 para las líneas de traducción:



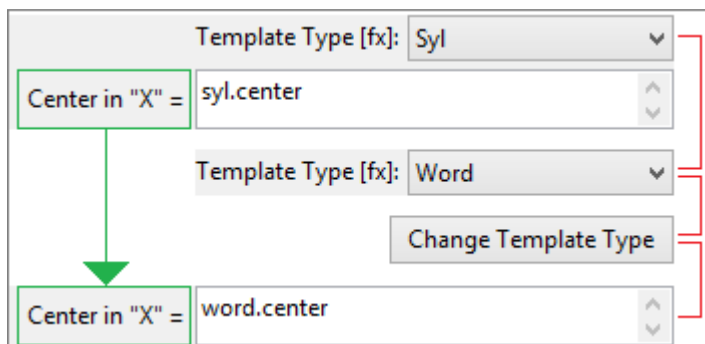
El modo **Word** permite aplicar efectos a las líneas de karaoke, palabra por palabra, es decir que retorna una línea de fx por cada palabra que contenga cada una de las líneas de karaoke seleccionadas para aplicar un efecto.

Es modo es similar al modo “**Translation Word**“, pero con la diferencia de que este último calcula la duración de cada palabra (**word.dur**) dependiendo de la cantidad de letras (caracteres) que cada una de ellas tenga; y el modo “**Word**” calcula la duración de las palabras dependiendo la duración individual de cada una de la sílabas que la componen.

Y como ya es sabido, luego de seleccionar el modo con el que aplicaremos nuestros efectos, debemos pulsar el botón “**Change Template Type**“:



Ejemplo:



Del ejemplo anterior notamos cómo la variable en “**Center in X**” = **syl.center**, luego de seleccionar el **Template Type: Word** y pulsar el botón “**Change Template Type**”, cambió a “**Center in X**” = **word.center**



El modo **Convert to Hiragana** convierte todo el texto de las líneas de karaoke (**Romaji** o **Katakana**) a **Hiraganas**. Este modo no afecta a las líneas de traducción, solo afecta a aquellas líneas separadas por sílabas que estén en **Romaji** o en **Katakana**.

Si el texto en una línea, a pesar de estar separado por sílabas, no está en **Romaji** o en **Katakana**, no se verá afectado por el modo y quedará tal cual.

Ejemplo:



El modo **Convert to Katakana** convierte todo el texto de las líneas de karaoke (**Romaji** o **Hiragana**) a **Katakana**. Este modo no afecta a las líneas de traducción, solo afecta a aquellas líneas separadas por sílabas que estén en **Romaji** o en **Hiragana**.

Si el texto en una línea, a pesar de estar separado por sílabas, no está en **Romaji** o en **Hiragana**, no se verá afectado por el modo y quedará tal cual.

Ejemplo:



El modo **Convert to Romaji** convierte todo el texto de las líneas de karaoke (**Hiragana** o **Katakana**) a Romaji. Este modo no afecta a las líneas de traducción, solo afecta a aquellas líneas separadas por sílabas que estén en **Hiragana** o en **Katakana**.

Si el texto en una línea, a pesar de estar separado por sílabas, no está en **Hiragana** o en **Katakana**, no se verá afectado por el modo y quedará tal cual.

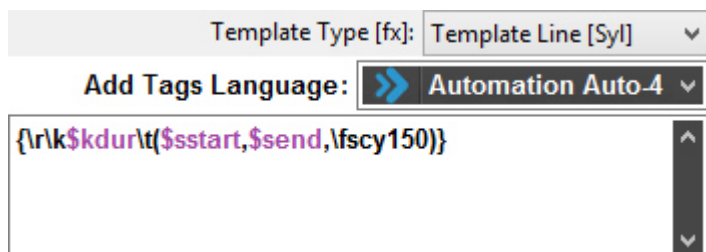
Ejemplo:



El modo **Template Line [Syl]** es el conocido y poco usado “**template line**” del **Aegisub**, que aplicaba efectos a cada una de las sílabas de las líneas de karaoke, pero todo dentro de una única línea de fx retornada por cada línea seleccionada para aplicar un efecto.

Ejemplo:

Los tres diferentes modos de **Template Line** del **KE** tienen la particularidad de que en los dos tipos de lenguaje de **Add Tags** (**Lua** y **Automation Auto-4**) se pueden escribir los tags en lenguaje **Automation Auto-4**, e incluso incluir las llaves (“{ }”) en donde escribir a los mismos:

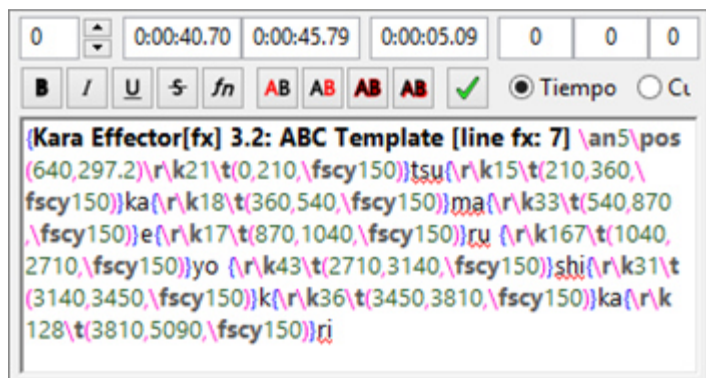


El tag `\r` cancela a todos los tags de las sílabas anteriores con el fin de no afectar a las sílabas siguientes. Lo que significa es `\r = reset`

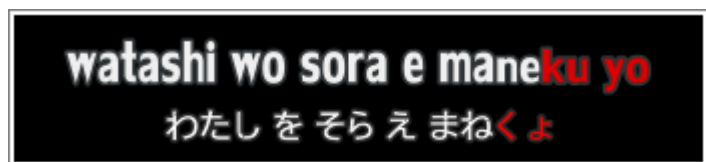
Al aplicar, se genera una línea fx por cada línea karaoke:

Estilo	Actor	Efecto	Texto
English			*Dos corazones que se buscan conf
Romaji	lead-in	Effector [Fx]	*ko*do*ku *na *ho*ho *wo *ni
Romaji	lead-in	Effector [Fx]	*yo*a*ke *no *ke*ha*i *ga *s
Romaji	lead-in	Effector [Fx]	*wa*ta*shi *wo *so*ra *e *ma
Romaji	lead-in	Effector [Fx]	*ki*bo*u *ga *ka*na*ta *de *
Romaji	lead-in	Effector [Fx]	*ma*yo*i *na*ga*ra *mo *ki*
Romaji	lead-in	Effector [Fx]	*su*re*chi*ga*u *i*shi*ki *te
Romaji	lead-in	Effector [Fx]	*tsu*ka*ma*e*ru *yo *shi*ki*
Romaji	lead-in	Effector [Fx]	*mo*to*me *a*u *ko*ko*ro *s

Al ver en detalle a una de las líneas generadas, notamos que el efecto se aplica a cada una de las sílabas, pero todo dentro de una única línea fx por cada línea de karaoke a la que le aplicamos el efecto:

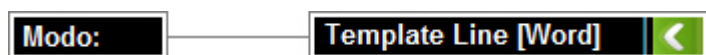


Y un ejemplo del anterior efecto en el vídeo sería:



Para agregar tags en este modo, no es obligatorio es uso de las llaves (“{ }”), pero igual el **KE** da la opción de ponerlos ya que hay muchos que aprendieron a hacer efectos de esta forma y ya están acostumbrados a ponerlas.

En este punto, el modo “**Template Line [Syl]**” es lo mismo que el “**template line**” del **Aegisub**, pero más adelante veremos que hasta acá llegan las similitudes entre los dos modos. En la web aún es posible hallar algunos **templates lines** del **Aegisub**, los cuales podríamos copiar y pegar tal cual en la celda de texto **Add Tags**, y el **KE** los reconocería y aplicaría sin ningún problema.



El modo **Template Line [Word]** es similar al “**Template Line [Syl]**” con la obvia diferencia de que aplica los efectos palabra por palabra y todo dentro de una única línea fx por cada línea de karaoke a la que le apliquemos un efecto.

Ejemplo:

Template Type [fx]: Template Line [Word] ▾

Add Tags Language: >> Automation Auto-4 ▾

`\r\l($wstart,$wend,\fscx!$scale_x + 50!\blur3)`

Los tres modos de “**Template Line**” del **KE (Word, Syl y Char)**, pueden ser aplicados tanto a líneas de karaoke como a líneas de traducción, con la particularidad que si aplicamos un efecto en “**Template Line [Syl]**” a una línea de traducción, éste se aplicará palabra por palabra, como si fuera un “**Template Type [Word]**”.

*Mis *mejillas *se *manchan *con *lágrimas *de *soledad
*Pero *puedo *sentir *la *llegada *del *amanecer
*Que *me *atrae *con *rumbo *hacia *los *cielos
*La *esperanza *espera *al *otro *lado, *por *eso *volaré
*Me *pierdo *en *el *camino *cada *vez *que *te *busco
*Siento *los *pensamientos *que *dejaste *atrás
*Me *aferraré *a *ti *y *nunca *te *soltaré
*Dos *corazones *que *se *buscan *conforman *este *sueño

Modo: Template Line [Char] <

El modo **Template Line [Char]** es similar al “**Template Line [Syl]**” con la obvia diferencia de que aplica los efectos letra por letra y todo dentro de una única línea fx por cada línea de karaoke a la que le apliquemos un efecto.

Ejemplo:

Template Type [fx]: Template Line [Char] ▾

Add Tags Language: >> Automation Auto-4 ▾

`\r\bord4\blur3\1c!color.interpolate("&H00FFFF&", "&H0000FF", $ci/$charn)!`

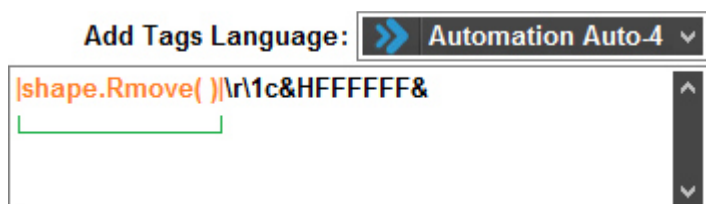
Barras Paralelas en Templates Line [||] :

Estas barras hacen posible agregar tags una única vez al inicio de las líneas fx generadas en los siguientes modos del **Template Type**:

- **Temaplate Line [Word]**
- **Temaplate Line [Syl]**
- **Temaplate Line [Char]**

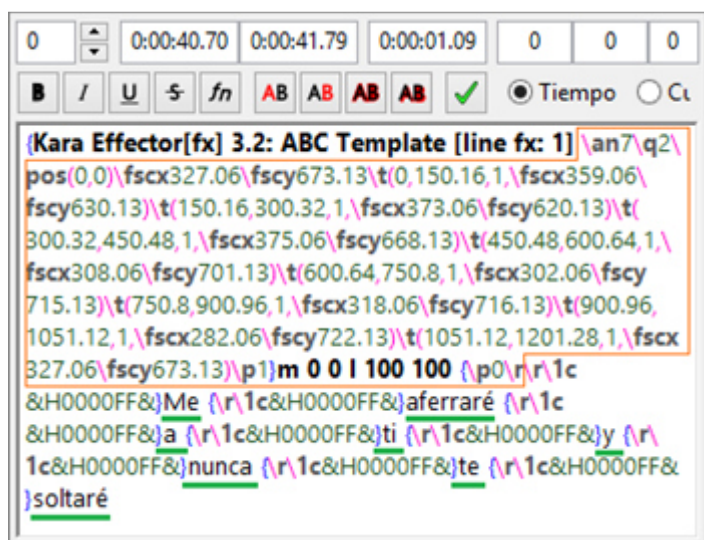
Esta herramienta es exclusiva del **Kara Effector**, y nos da la posibilidad de genera un efecto independiente a los objetos karaoke, al inicio de las líneas fx generadas por los tres anteriores modos enunciados.

Ejemplo:



De los tags agregados en el ejemplo anterior, lo que está en las barras paralelas (resaltado en color naranja) se aplicará una sola vez al inicio de la línea fx, y el resto de los tags (el texto en negro de la imagen) se aplicará en cada uno de los objetos karaoke dependiendo de cuál modo hayamos elegido entre los tres de **Template Line**. Para el ejemplo anterior se usó un **Template Line [Word]**.

Es una de las líneas generadas veremos algo como esto:

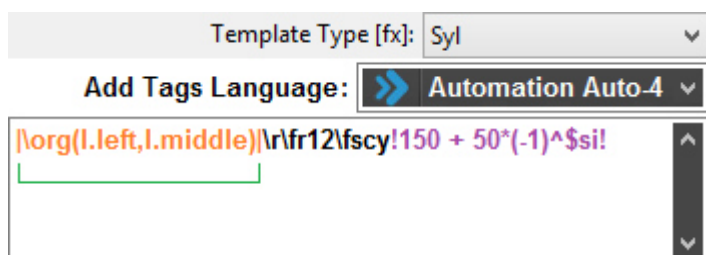


Los tags resaltados en naranja son los generados por la función **shape.Remove** y están solo al inicio de la línea de fx, el resto de los tags se aplicaron palabra por palabra.

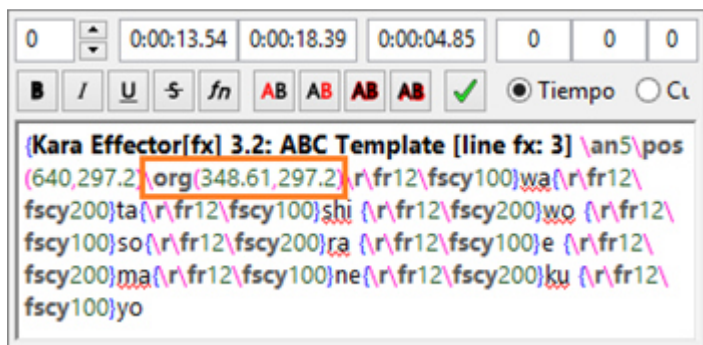
La barra (|) viene como uno de los caracteres de la mayoría de los teclados, pero para el caso de que no lo tengan o no lo encuentren en los suyos, la opción para sacarla es:

Alt + 124

Ejemplo:



El tag en las barras saldrá una sola vez al inicio de la línea:



Y al aplicar el efecto notamos los resultados:



En conclusión, el uso de las barras paralelas en **Add Tags** en los **Template Type**: “**Template Line [Word]**“, “**Template Line [Syl]**” y “**Template Line [Char]**“, es como aplicar un efecto **Templet Type: Line** dentro de ellos, ya que dichas barras aplican los tags una sola vez por línea fx generada.