# Aegisub

*It's time for subtitle!!*

# 一、概述

SSA 全称 SubStation Alpha，是由 CS Low(又称 Kotus)创建的一种字幕格式，用以实现比传统字幕诸如 srt 等格式更为复杂的功能. SSA 目前的版本为 v4.00. SSA 同时也是一款软件的名称，专用于创建和编辑 SSA 格式的字幕. ASS 是一种比 SSA 更为高级的字幕格式，全称 Advanced SubStation Alpha，实质是 SSA v4.00+ 版本. 它拥有比 SSA 更多的功能.

1. SSA v4.00与之前的 SSA 版本格式不同. v4可以阅读和加载以前版本的格式，但之前版本不支持 v4以上的编写格式. 换句话说, SSA 从 v4版本开始可以阅读它认识的命令而忽略不认识的命令，从而可以向下兼容，也可以向上兼容.
2. 文件为普通的 DOS 文本格式. 也就是说它可以用记事本打开编辑，同时需要注意编写出错时有可能导致无法预料的结果.
3. 文件编写时所划分的各部分，从形式上来说类似于 ini 文件，但它并非真正的 ini 文件
4. 各部分中的绝大多行都以一说明性的文字加上冒号来开头，指明该行包含哪一些信息.
5. 每一行中的信息都以逗号分隔. 因此风格名和人物角色名中要求不能出现逗号.
6. 事件部分([Events])里的各行可以不分先后. 也就是说人物对白行可以不按时间顺序排列
7. 不正确的行会被忽略. 同时会给出警告指出被忽略的行数
8. 一行里包含了完整的信息，必须在一行内写完，不能分成多行
9. 当文件中引用了一个未知的风格名(style)时，加载时会用默认的风格来替代(Default)
10. 当一个风格(Style)中引用了系统中没有安装的字体，则会用 Arial 字体来代替.

# 二、文件各部分解析

### [Script Info]

这一部分包含了文件内容的标题和总体信息. [Script Info]这一行必须是 v4版本文件的第一行

### [v4 Styles]

字幕正文使用的风格都在这一部分做出相关定义. 注: ASS 使用的是[v4+ Styles]

### [Events]

这部分包含所有的事件, 有字幕, 评论, 图片, 声音, 影片和命令. 基本上屏幕中出现的所有内容都集中在这一部分.

### [Fonts]

如果想把字体内嵌入字幕文件, 那么字体文件须采用数字编码后放在这一部分. 只有 truetyle 字体才能内嵌入 SSA/ASS 文件

每一个内嵌字体文件以一行开头, 格式如下:

**fontname: <文件名>**

开头的"fontname"必须全部用小写, 如果大写会让 ASS 文件视其为文件编码的一部分.

<文件名>是 SSA 文件保存字体时使用的文件名, 命名规则如下:

truetype 字体原来的字体名称

加一条下划线

如果是粗体则加一个"B"

如果是斜休则加一个"I"

加一个数字表明字体编码(字符集)

最后加上".ttf"

例如:

fontname: comic_B0.ttf

在这一行之后是一些可打印的字符组成的行, 代表组成这个字体的二进制字符, 除了最后一行可能短些, 其余每行有 80个字符.

从二进制转换到字符用的是 UUE-encoding 的编码方式

### [Graphics]

如果选择内嵌图片, 那这一部分就包含了所有用到的数字编码格式的图片文件. 开头一行的格式如下:

**filename: <文件名>**

开头的”filename”必须为小写，如果大写会被认为是文件编码的一部分.

<文件名>是 SSA 文件保存图片时使用的文件名，它与[Events]事件部分中提及的图片名称一致.

SSA 会把文件中找到的任何文件保存到 SSA 的程序目录中的”Pictures”子目录中. 例如: c:\program files\Sub Station Alpha v4.00\Pictures. SSA 会先从文件本身中寻找这些编码好的文件，但当没有找到时会去”Pictures”这个子目录里去找.

注：现在的 SSA 文件已经很少包含”[Pictures]” 或者 “[Fonts]” 这两个部分. 因为这些功能只被 Sub Station Alpha 这一个程序所支持. 而其它的 filter (Vobsub/Vsfilter/Avery Lee Subtitler filter) 都不支持.

# 三、各种类型的行

在这一节里简要地说明在每个部分中出现的所有行的类型和大致功能，各自具体说明参见后面的章节.

| | |
|---|---|
| ; | 只在编写中请说明作用的行，加载字幕时不可见. |
| Title: | 标题，是对字幕的描述 |
| Original Script: | 最初创建字幕的人 |
| Original Translation: | (可选) 最初翻译对话的人 |
| Original Editing: | (可选) 最初的编辑者，一般是所有参与翻译和校对等工作的人 |
| Original Timing: | (可选) 最初的时间轴人员 |
| Synch Point: | (可选) 指出从哪一个时间点开始进行字幕加载播放 |
| Script Updated By: | (可选) 对原字幕对话进行编辑更新的人 |
| Update Details: | 进行了哪些更新等具体信息 |
| ScriptType: | 对 SSA/ASS 文件的版本做说明，例如"v4.00". ASS 的版本为"v4.00+" |
| Collisions: | 当两条字幕重叠时，如何进行相对移动 |
| PlayResY: | 文件所使用的视频高度参考标准 |
| PlayResX: | 文件所使用的视频宽度参考标准 |
| PlayDepth: | 加载字幕时所使用的颜色深度 |
| Timer: | 对字幕加载的速度调整，数值为百分数. 例如"100.0000"代表100%. |
| ScaledBorderAndShadow: | 边框宽度与阴影深度是否随着视频分辨率同等比例缩放. |
| Style: | 定义每条字幕所使用的风格 |
| Dialogue: | 指明为对话事件，即屏幕上出现的字幕 |
| Comment: | 指明此行是评论/解释事件，它与 Dialogue, Picture, Sound, Movie 或者 Command 事件包含相同的信息，以此来进行解释说明，但加载字幕时不会出现在屏幕 |
| Picture: | 指明为图片事件，即显示.bmp, .jpg, .gif, .ico 或者 .wmf 格式的图片(注意不支持 png,且 filter 不支持加载图片) |
| Sound: | 指明为声音事件，即播放.wav 格式的声音(filter 不支持) |
| Movie: | 指明为电影事件，即加载 avi 视频(filter 不支持) |
| Command: | 指明为命令事件，即可在后台打开某个程序 |

# 四、[Script Info]部分的标题行

| | |
|---|---|
| ; | 分号，后面可以跟任何内容．这一行是说明性文字，加载字幕时不显示<br>注意此类型行必须要把分号放最前．老版本不是用分号而是用!: |
| Title: | 标题，如果没有提供，则自动使用<untitled> |
| Original Script: | 剧本的最初作者，若没有提供则自动使用<unknown> |
| Original Translation: | (可选)原剧本的翻译者，若没有提供则该行不显示 |
| Original Editing: | (可选)原剧本的编者和校对，若没有提供则该行不显示 |
| Original Timing: | (可选)原剧本的时间轴人员，若没有提供则该行不显示 |
| Synch Point: | (可选)从哪个时间点开始加载字幕，若没有提供则该行不显示 |
| Script Updated By: | (可选)对原剧本的修改/更新人员，若没有提供则该行不显示 |
| Update Details: | 更新的具体信息，若没有提供则该行不显示 |
| Script Type: | SSA 的版本信息，ASS 的版本为"v4.00+" |
| Collisions: | 当字幕时间重叠时，前后字幕的堆叠方式．<br>值为"Normal"时，后一条字幕出现在前一条字幕的上方．<br>如果值为"Reverse"时，前一条字幕往上移动给后一条字幕让位． |
| PlayResY: | 文件所使用的视频高度参考标准，如果使用 Directdraw 回放 SSA v4会自动选择最相近的启用的设置 |
| PlayResX: | 文件所使用的视频宽度参考标准，如果使用 Directdraw 回放 SSA v4会自动选择最相近的启用的设置．<br>如果只提供了 PlayResX, PlayResY 其中一种，那另一种会按**实际视频的像素值**为准．<br>提供的分辨率数值影响以下参数：<br>1) 所有给出的坐标(到边缘的距离, \pos, \move, 矢量绘图等)都以此分辨率作为参照．<br>2) 所有的文字字号均按照此分辨率等比例放大缩小<br>3) 当 ScaledBorderAndShadow 被启用时，所有边框宽度和阴影深度都按照此分辨率与实际分辨率的比例等比例缩放<br>4) 这个分辨率不影响最终显示文字的宽高比，但影响矢量绘画图形的宽高比． |
| PlayDepth: | 加载字幕时使用的色深(颜色的数目)，如果使用 Directdraw 回放 SSA v4会自动选择最相近的启用的设置 |
| Timer: | 字幕加载的速度调整，数值为百分数．例如"100.0000"代表100%．其数值有4位小数点.它相当于对 ASS 字幕的时间速度进行乘法运算.当速度大于100%时，总时间会缩短，而相应的字幕会越来越靠前.当速度小于100%时，总时间会延长，而相应的字幕 |

会越来越靠后.

WrapStyle: 定义默认的换行方式,

0: 智能换行, 行分得较平均, 上面的行较长

1: 一行结束后从行尾的词分行

2: 不换行. 此模式下只有\n, \N 才换行

3: 与模式0相同, 但下面的行分得比较长

ScaledBorderAndShadow: 指定边框宽度与阴影深度是否随着视频分辨率等比例缩放. 可为 Yes, No. 默认为 No.

当取值为 No 时, 边框宽度与阴影深度完全按照指定的像素数显示.

当取值为 Yes 时, 边框宽度与阴影深度随着实际视频的分辨率同等比例缩放.

# 五、[v4+ Styles]部分的风格行 Style

Style 定义了字幕的样式和位置．所有的 Style 都在单独的 Style 里进行定义．

除了阴影/边框的类型和深度，其余所有的风格设置都可以由字幕文本中的覆写代码所替代．

在定义风格之前先要有一行"Format:"来定义风格中每一个字段所代表的含义，这些字段名称必须拼写准确，顺序可以打乱，字段名表示如下：

Name, Fontname, Fontsize, PrimaryColour, SecondaryColour, TertiaryColour, BackColour, Bold, Italic, Underline, StrikeOut, ScaleX, ScaleY, Spacing, Angle,BorderStyle, Outline, Shadow, Alignment, MarginL, MarginR, MarginV, AlphaLevel, Encoding

字段1: Name．风格(Style)的名称．区分大小写．不能包含逗号．

字段2: Fontname．使用的字体名称，区分大小写．

字段3: Fontsize．字体的字号

字段4: PrimaryColour．设置主要颜色，为蓝-绿-红三色的十六进制代码相排列，BBGGRR．为字幕填充颜色

字段5: SecondaryColour．设置次要颜色，为蓝-绿-红三色的十六进制代码相排列，BBGGRR．在卡拉 OK 效果中由次要颜色变为主要颜色．

字段6: TertiaryColour(ASS 中的名称为 OutlineColor)，设置轮廓颜色，为蓝-绿-红三色的十六进制代码相排列，BBGGRR.

字段7: BackColour，设置阴影颜色，为蓝-绿-红三色的十六进制代码相排列，BBGGRR.

字段4-7: ASS 的这些字段还包含了 alpha 通道信息．(AABBGGRR)，注 ASS 的颜色代码要在前面加上&H

字段8: Bold．-1为粗体，0为常规

字段9: Italic．-1为斜体，0为常规

字段9.1: Underline．[-1 或者 0] 下划线

字段9.2: Strikeout．[-1 或者 0] 中划线/删除线

字段9.3: ScaleX．修改文字的宽度．为百分数

字段9.4: ScaleY．修改文字的高度．为百分数

字段9.5: Spacing．文字间的额外间隙．为像素数

字段9.6: Angle．按 Z 轴进行旋转的度数，原点由 alignment 进行了定义．可以为小数

字段10: BorderStyle．1=边框+阴影，3=纯色背景．当值为3时，文字下方为轮廓颜色的背景，最下方为阴影颜色背景．

字段11: Outline．当 BorderStyle 为1时，该值定义文字轮廓宽度，为像素数，常见有0, 1, 2, 3, 4.

字段12: Shadow．当 BorderStyle 为1时，该值定义阴影的深度，为像素数，常见有0, 1, 2, 3, 4.

字段13: Alignment. 定义字幕的位置. 字幕在下方时, 1=左对齐, 2=居中, 3=右对齐. 1, 2, 3加上4后字幕出现在屏幕上方. 1, 2, 3加上8后字幕出现在屏幕中间. 例: 11=屏幕中间右对齐.

字段13: Alignment. 对于 ASS 字幕而言, 字幕的位置与小键盘数字对应的位置相同.

字段14: MarginL. 字幕可出现区域与左边缘的距离, 为像素数

字段15: MarginR. 字幕可出现区域与右边缘的距离, 为像素数

字段16: MarginV. 垂直距离

对于处在下方的字幕, 此值是字幕到**底端**的像素数

对于处在屏幕上方的字幕, 此值为字幕到**顶端**的像素数.

对于处在屏幕中间的字幕, 此值被忽略.

字段17: AlphaLevel. SSA 字幕用来定义透明度

字段17: AAS 没有该字段.

字段18: Encoding. 指明字体的字符集或编码方式. 如0为英文, 134为简体中文, 136为繁体中文. 当文件为非 UNICODE 类型编码时, 该值对字幕的显示起作用.

# 六、[Events]部分的对话行 Dialogue

Dialogue 类型的行(对话行)包括字幕对白，时间轴信息，以及对白的显示方式.

在对话行出现前必须有一条格式行 Format: 来对逗号分隔的每个字段进行定义，该格式行中的每一个字段必须拼写准确，内容如下:

Marked, Start, End, Style, Name, MarginL, MarginR, MarginV, Effect, Text

最后的一个字段永远都是对白文字的字段，因此可以包含逗号. 前面的字段顺序可以改变.

字段1: Marked. 值为0表示该行为"未标识"行，值为1表示该行为"标识"行

字段1: Layer. ASS 的这个字段名是 Layer(图层)，可以为任意的整数. 图层不同的两条位置/时间有重叠的字幕不被视为有冲突，图层号码大的字幕显示在图层号小的上方.

字段2: Start. 事件开始的时间，格式为0:00:00:00(时:分:秒:百分数)，注意小时只有一位数

字段3: End. 事件结束的时间. 格式为0:00:00:00(时:分:秒:百分数)，注意小时只有一位数

字段4: Style. 该条字幕所使用的风格. 风格的具体信息在[V4+ Style]这一部分中进行定义

字段5: Name. 角色名，指出对白是由影片中哪位演员所说的. 字幕加载时不显示，只为了编写时理解方便.

字段6: MarginL. 使用新的与左边缘的距离，为4位数字代表的像素值. 0000代表使用当前 Style 定义的值.

字段7: MarginR. 使用新的与右边缘的距离，为4位数字代表的像素值. 0000代表使用当前 Style 定义的值.

字段8: MarginV. 使用新的垂直距离，为4位数字代表的像素值. 0000代表使用当前 Style 定义的值. 具体说明参见上面[v4+ Style]里的说明

字段9: Effect. 过渡效果. 可以为空值，或者为三种过渡效果之一.

效果名称区分大小写，必须拼写准确. 且不加任何引号:

"Karaoke" 是卡拉 OK 效果，每个字依次高亮显示. 注: 在 ASS 中该效果已经废弃不用.

"Scroll up;y1;y2;delay[;fadeawayheight]" 滚动效果，指文字/图片向上滚动. 各参数以分号分隔.

y1与 y2是屏幕垂直区域的像素值，位置可以互换. 当这两个值都为0时则全屏幕内滚动

delay 可取值1-100，代表滚动速度的降低值. 当其为0时滚动速度最快.

"Banner;delay"横幅效果. 所有文字被合并到单行，并从右至左横向移动.

delay 的值可由1到100，代表横幅移动速度的降低值. 当其为0时移动速度最快.

"Scroll down;y1;y2;delay[;fadeawayheight]" 向下滚动

"Banner;delay[;lefttoright;fadeawaywidth]"

注意红字为 ASS 新增功能.

lefttoright 可取值0或1，为可选，默认为0(右出左进)

当 delay 值大于0时，移动速度为(1000/delay)像素每秒

(注意: Avery Lee 的字幕插件阅读滚动效果的顺序为: delay;y1;y2)

fadeawayheight 以及 fadeawaywidth 可令效果边缘的像素呈现透明.

字段10: Text. 为对白字幕区域, 是最终出现在屏幕上的字幕. 任何位于第9个逗号后的内容均被看作是对白字幕, 所以本身可以包含逗号.在这一个字段中可以包含\n, \N, \h 这三种分行/空格代码, 以及其它在大括号{ }内的风格覆写控制代码.

# 七、[Events]事件部分的注解行 Comment

在[Events]这一部分内，以 Comment: 开头的行．它可以与其它类型的事件行包含一样的信息，但不会被作为字幕加载到屏幕上．它起评论/说明的作用．

# 八、[Events]事件部分的图片行 Picture

在[Events]这一部分内，以 Picture：开头的行．它与 Dialogue 行包含一样的控制信息，但是在字段10的位置指定要显示的图片完整路径与图片名称．在前面字段指定的风格被忽略，滚动效果可以运用到图片事件上．MarginL 和 MarginV 被用来指定图片与左边缘与下边缘的像素距离．当 MarginL 为0000时图片水平居中显示．当 MarginV 为0000时，图片垂直居中．

支持的图片格式有.bmp, .jpg, .gif, .ico 以及.wmf 格式的图片(不支持 png) 注意，只有 SSA 软件能支持加载图片事件，filter 则不支持.

# 九、[Events]事件部分的声音行 Sound

在[Events]这一部分内，以Sound：开头的行．它与Dialogue行包含一样的控制信息，但是在字段10的位置指定要加载的声音文件完整路径与名称，格式为.wav．风格与距离等值被忽略，而且结束的时间值也被忽略．该声音会播放到它结束，或者播放到新的声音行加入为止．

注：各 filter 不支持加载声音行

# 十、[Events]事件部分的影片行 Movie

在[Events]这一部分内，以 Movie: 开头的行．它与 Dialogue 行包含一样的控制信息，但是在字段10的位置指定要加载的视频文件完整路径与名称，格式为.avi．风格与效果等值被忽略

结束的时间值(End)指出影片画面消失的时间，但如果 avi 文件仍然没有结束，则其声音仍然会持续播放．

MarginL 和 MarginV 被用来指定影片与左边缘，上边缘的像素距离(与图片行不同)．当 MarginL 为0000时视频水平居中显示．当 MarginV 为0000时，视频垂直居中．

注: Filter 不支持加载影片行

当 Movie 与 Sound 行有时间重叠时，实际听到的声音以先开始的内容为准．

# 十一、[Events]事件部分的命令行 Command

在[Events]这一部分内，以Command：开头的行．它与Dialogue行包含一样的控制信息，但是在字段10的位置指定要运行的程序完整路径与名称．

风格，距离，效果，结束时间都被忽略．程序会运行到它结束为止，或者运行到手动关掉为止． 注：Filter 不支持加载命令行．

SSA 软件内置的一些命令可以出现在 SSA 文件内．因完全用不到故将英文原文引用如下：

There are also internal SSA commands which can appear in SSA scripts – the "SSA:Pause", "SSA:Wait for trigger" command events, and genlock control commands. These all begin with "SSA:"

The SSA:Pause command has the same effect as pressing "P" during script playback. It is useful as a second "synch point" to resume subtitling after switching sides of a laserdisk.

The "SSA:Wait for audio trigger" command has the same effect as pressing "P" during script playback, but pausing is automatically cancelled if the audio input to the computer exceeds a specified "trigger" level. It is useful as a second "synch point" to resume subtitling after switching sides of a laserdisk. The audio triggering can be overridden to resume playback – by pressing "P".

Audio triggering "times out" after 10 minutes – If no audio peak of sufficient magnitude is received, and "P" is not pressed within 10 minutes – then playback will resume anyway.

# 十二、风格覆盖代码(Tags, Codes)

风格覆写控制代码专用于[Events]这一部分的最后一个字段中，对于文字/图片风格的重新定义.

除了\n, \N, \h 这三个代码之外，其余所有代码都必须放在大括号{ }之内

所有覆写代码都以反斜线开头\

多个代码可以放在一个{ }内

所有覆写代码作用于其后的所有文字. 如果只想作用于选定的文字，则需要在选定文字的后面加一个"取消"作用的代码. 但也有少数代码会自动应用于整行文字，如设定位置的代码.

下面将所有代码分成一般，绘图两组来详细解释:

**(一) 一般的代码**

**\n** 软性分行(回车)，只在分行模式(WrapStyle)为2时有效. 在其它分行模式下相当于一个空格

**\N** 硬性分行(回车)，在任何分行模式下都有效

**\h** 硬性空格. 它保证显示字幕时不会在它的这个空格上分行(保证左右两个词在同一行)

**\b<0 或 1>** \b1令文字变为粗体. \b0强制文字不是粗体(bold)

当参数大于1时，会被作为字体的重量值. (注: 大多字体只有量化到2级或3级的粗度，所以很少用到这个重量值)

字体重量值为100的倍数，如100为最细, 400为普通, 700为粗体, 900为最粗

**\i<0 或 1>** \i1令文字变为斜体. \i0强制文字不是斜体(italic)

**\u<0 或 1>** 下划线(underline)开关

**\s<0 或 1>** 中划线(删除线)开关(strikeout)

**\bord<宽度>** 指定边框宽度(border)，像素数. 可以为小数

**\shad<深度>** 指定阴影深度(shadow)，不能为负数

**\be<0 或 1>** 模糊边缘 blur edges

**\fn<字体名称>** 指定使用系统中已安装的字体，区分大小写. 如果使用的字体没有安装，则会用 Arial 来替代(font name).

注字体名与 fn 间不能有空格，也没有其它的括号等

**\fs<字号>** 指定文字的大小，<字号>是一个指代高度的像素值，只能用整数. (font size)

**\fsc<x 或 y><百分数>** 缩放文字大小. <x 或 y>指定文字是横向还是纵向缩放. <百分数>指定调整到百分之多少. 同样适用于矢量图形.

**\fsp<像素值>** 文字间增加额外的间隔(font spacing)，默认为0

**\fr[<x/y/z>]<度数>** 文字旋转一定的度数(font rotation)

<x/y/z>表明文字沿着三维空间中哪个坐标轴旋转.

原点由\org 代码来指定，否则由默认的定位点来决定(参见\pos 的说明)

\fr 默认代表\frz. <度数>可以为负值，可以大于360. 此命令同样适用于矢量图形.

**\fe<字符集>** 指定文字的编码(font encoding). 例如0为英文, 134为简体中文, 136为繁体中文. 1为系统默认.

注：当 ASS/SSA 文件本身的编码为非 Unicode 编码时，这个数值总影响到按哪种代码页来显示字幕.

推荐保存 ASS/SSA 文件时选择 Unicode 方式的编码，例如 UTF-8. 这样可以忽略这个值的设置

**\c&H<bbggrr>&** 指定文字的颜色(color). <bbggrr>是一个十六进制的 RGB 数值，但颜色顺序相反(蓝-绿-红).

<bbggrr>排在最前的00可以忽略不写，例如{\c&HFF&}={\c&H0000FF&}，为纯红色

\1c&Hbbggrr&, \2c&Hbbggrr&, \3c&Hbbggrr&, \4c&Hbbggrr&分别设定主要，次要，边框，阴影颜色

\c 相当于\1c，为字体本身填充颜色

**\alpha&H<aa>&** 设定文字的透明度(alpha). <aa>是一个十六进制数值. 00为全见, FF 为全透明

\1a&Haa&, \2a&Haa&, \3a&Haa&, \4a&Haa&分别设定主要，次要，边框，阴影颜色的透明度

\alpha 一次性调整文字所有元素的透明度

**\a<位置>** <位置>是一个数字代码，用来代表字幕出现在屏幕中的位置(alignment)

1, 2, 3代表出现在画面底端的字幕，分别为左对齐，居中，右对齐

5, 6, 7代表出现在画面顶端的字幕，分别为左对齐，居中，右对齐(在1,2,3基础上加了4)

9, 10, 11代表出现在画面中间的字幕，分别为左对齐，居中，右对齐(在1,2,3基础上加了8)

当一行出现多个\a 代码时，只有最前面的一个有效

**\an<位置>** <位置>为一数字代码，代表字幕的位置，从1到9，与小键盘的数字键代表的位置一致

当一行出现多个\an 代码时，只有最前面的一个有效

注：一般情况下\a, \an 设置字幕位置，但当有\pos, \move 等代码时，\a 和\an 设置的是文字的定位点.

**\k<时间长度>** 卡拉 OK 效果，高亮之前文字使用次要颜色，高亮后使用主要颜色

<时间长度>代表在下一小段出现高亮效果之前的当前段高亮停留时间，值为百分之一秒的倍数

\k<时间长度>按照每一分隔好的小段来进行高亮显示

\kf 或者\K<时间长度>是从左至右的流畅填充高亮

\ko<时间长度>，与\k 相似，但在高亮之前文字边框也被去掉，高亮后才显示边框

**\q<方式>** 定义分行方式，即 WrapStyle

值为0：智能分行，大致平均分行，不能完全平均时上面的行较长.

值为1：行尾分行，尽管排满一整行后，再分到下一行

值为2：不分行，超出长度的行会排到屏幕以外．这种方式下，\n 和\N 都可以强制分行

值为3：智能分行，与方式0相似，但下面的行比较长

**\r[<风格>]** 取消一行中之前的所有覆写代码效果，包括动态特效．(restore/reset)

<风格>令其恢复到指定的风格，如果没有指定<风格>则恢复到这一行的默认风格

**\t([<时值1>, <时值2>, ] [<加速度>,] <风格代码>)**

提供从一种风格转到另一种风格的逐渐变化的动态效果．只有部分风格代码可以用\t 进行动态变换：

| | |
|---|---|
| 字体 | \fs \fsp \c \1c \2c \3c \4c \alpha \1a \2a \3a \4a |
| 几何形 | \fscx \fscy \frx\fry \frz \fr |
| 其他 | \bord \shad \clip \pos |

注：对于 \clip，只有矩形可以呈动态效果，矢量绘画图形无法呈动态

<时值1>与<时值2>是从该行开始显示后计算的毫秒数，两时值间的时间间隔就是动态效果的运行时间(两时值无先后之分).

这两个时候没有指定时相当于<时值1> = <时值2> = 0．这时动态效果在整行的时间内运行

<加速度>没有指定时相当于1，此时匀速变化．<加速度>在0和1之间时速度由快变慢，大于1时由慢变快.

在<时值1>之前，显示内容是{\t}代码之前的风格，<时值2>以后是<风格代码>所指定的风格.

**\pos(<x>, <y>)**设置该行显示的位置，x,y 为定位点的坐标值，屏幕可见区的坐标为正值．原点(0,0)在左上角落

对于不同屏幕排列的字幕，定位点的相对位置不同.

\an1的定位点在字幕左下角，\an2的定位点在字幕正中下方，\an3的定位点在字幕右下角

\an4的定位点在字幕正中左端，\an5的定位点在字幕正中，\an6的定位点在字幕正中最右端

\an7的定位点在字幕左上角，\an8的定位点在字幕正中上方，\an9的定位点在字幕右上角

一行中有多个\pos 时以最前面的\pos 值为准

**\move(<x1>, <y1>, <x2>, <y2>[, <时值1>, <时值2>])**

提供从一个位置到另一个位置的移动动态效果.

x1, y1是定位点起始的坐标(定位点的说明参见\pos 代码), x2, y2是定位点结束的坐标

1)未达到<时值1>时, 文字位置的定位点在(x1, y1)

2)在<时值1>和<时值2>之间, 定位点从(x1, y1)移动到(x2, y2)

3)超过<时值2>后, 文字的定位点固定在(x2, y2)点.

当时值1, 时值2没有指明时相当于两个都是0, 此时在该行的整个时间段内匀速移动(时值的具体说明参见\t 代码)

时值1, 时值2可以大于该行的总时间段, 令没有达到终点时结束移动.

\move 只能够匀速移动, 不能够加速移动

一行中有多个\move 代码时只有排在最前面的\move 效果有效

一行中不能同时使用{\pos}和{\move}代码, 如果同时出现则只有排在最前面的效果有效

**\org(<x>, <y>)** 设置旋转的原点坐标(origin), 它影响一行中所有的旋转

当有旋转效果的一行中没有\org 代码, 则用于旋转的原点坐标就是默认的定位点.

原点坐标可以放置在画面可见区域以外, 足够远时通过旋转一个小角度可让字幕产生"穿过屏幕"的效果

一行中有多个\org 代码时只有排在最前面的\org 有效

注意: \t, \move 和 \pos 会忽略位置重叠的检测

**\fad(<淡入时间>,<淡出时间>)**

提供简单的淡入淡出效果. <淡入时间>与<淡出时间>之和不能超过该行的时间长度.

**\fade(<a1>,<a2>,<a3>,<时值1>,<时值2>,<时值3>,<时值4>)**

提供复杂的透明度变化效果. <a1>,<a2>,<a3>为三个不同的透明度值(alpha), 取值从0到255. 0全见,255全透明

这个代码里的7个参数要求全部写齐, 作用方式如下:

1) 在<时值1>之前, 透明度为<a1>

2) 在<时值1>与<时值2>之间, 透明度从<a1>变化到<a2>

3) 在<时值2>与<时值3>之间, 透明度为固定的<a2>

4) 在<时值3>与<时值4>之间, 透明度从<a2>变化到<a3>

5) 在<时值4>之后, 透明度为<a3>

**\clip(<x1>, <y1>, <x2>, <y2>)**

定义一个矩形框, 只有在这个框里的字幕才为可见

<x1>, <y1>, <x2>, <y2>为构成矩形的两个对角点

\clip([<等级>,] <绘图命令>)

定义一个绘画图形，令这个图形内的字幕可见

<绘图命令>参见第二组代码.

<等级>是指定图形的缩放等级．为2的(等级-1)次方．如/clip4，2的(4-1)次方为8，即将后面的图形缩至1/8

**(二)绘图代码**

**\p<等级>** 进入绘图模式并指定坐标的放大等级.

<等级>为坐标的缩放等级，按2的(等级-1)次方计算．如/clip4，2的(4-1)次方为8，即将后面的坐标缩至1/8

当<等级>=0时，关闭绘图模式.

**\pbo<y>** 定义所绘图形的基线偏移值.(baseline offset)

当 y>0时，图形的所有坐标沿 y 轴向下移指定的像素值

当 y<0时，图形的所有坐标沿 y 轴向上移指定的像素值

**绘图命令:**

**m <x> <y>** 将鼠标移至坐标(x, y)，同时将现有的图形封闭(即开始画新的图形)，所有绘画都以这个命令开始.

**n <x> <y>** 将鼠标移至坐标(x, y)，同时不封闭原有的图形

**l <x> <y>** 从鼠标原来的坐标位置画一条直线到(x, y)，并从这个点继续绘画

**b <x1> <y1> <x2> <y2> <x3> <y3>**

画一条三度贝塞尔曲线至(x3, y3)，以(x1, y1), (x2, y2)作为控制点

**s <x1> <y1> <x2> <y2> <x3> <y3> .. <xN> <yN>**

从现有坐标画一条"三次均匀 B 样条"(cubic uniform b-spline)到点(xN, yN)

该命令至少要含有三个坐标点(三个坐标时等同于贝塞尔曲线)

这个命令实质上是把几条贝塞尔曲线连结到一起.

**p <x> <y>** 沿长 B 样条(b-spline)到点(x, y)，作用相当于在 s 命令后多加一个坐标点(x, y)

**c** 结束 B 样条(b-spline)

**绘图代码的注意事项:**

1) 除了在\clip(..)命令中，绘图命令必须以{\p1+}开头，以{\p0}结尾． ("1+"指大于等于1的整数)

2) 所有绘图都必须以 m 命令开头

3) 所有图形都必须最终闭合

4) 所有没有闭合的图形会在起始点和终点之间连上一条直线来闭合

5) 在同一行中，若图形有重叠，则重叠部分执行异或逻辑运算(即正正得负，负负得正)

6) 如果相同的绘图命令在一起，则只需保留最前面命令代码，后面的坐标可以连着写

7) 绘图中使用的是相对坐标系．坐标原点由当前的基线位置(\pbo)，当前行的排列方式决定(参见\pos 的定位点说明)

命令 p c 只能用在 B 样条命令 s 的后面

**ASS 之后又有更新的版本，如 ASS2, ASS3等．这些在开发中没有整合出最终版本，因此没有官方的相关具体说明文件，也没有得到广泛使用．而它新增的一些代码和改动，可以为 VSFilter 2.39以上的版本识别和使用．虽不推荐在字幕发布中使用，但可用于压制中．**

边框宽度　　\xbord<字号>

\ybord<字号>

沿 x 轴和 y 轴分别对文字边框宽度进行调整．注：如果在一行中用了\xbord, \ybord 后又使用\bord，则会被\bord 覆写.

阴影深度　　\xshad<depth>

\yshad<depth>

沿 x 轴和 y 轴分别对文字阴影深度进行调整，可以使用负值

边缘模糊　　\be<强度>

按一般模糊的倍数来模糊文字边缘，<强度>必须为整数．注当强度过大时会导致文字"消失"

\blur<强度>

与\be 相似，但它用的是更加高级的高斯模糊，<强度>可以为非整数．注意过高的值可能占用过多系统 CPU

文字倾斜　　\fax<因数>

\fay<因数>

沿 x 轴和 y 轴分别对文字行进行倾斜调整．<因数>可以为负数．如\fax-0.5等同于斜体．(注：一般情况下因数的绝对值不要超过2)

卡拉 OK　　\kt<时间长度>

传统的卡拉 OK 模式是从左至右，依次填充音节．而\kt<时间长度>重新定义了该音节开始填充的时间

<时间长度>是指该音节从整行开始过了多长时间后开始填充．

例如{\k10}一{\kt30\k10}二{\kt10\k10}三

先填充"一"，过了0.1秒后"一"填充完毕，开始填充"三"，又过了0.1秒"三"填充完毕，再过了0.1秒(一行开始后0.3秒)开始填充"二"

动态风格　　\t([<时值1>, <时值2>, ] [<加速度>,] <风格代码>)

除了原有的风格，新增了以下的代码也可以使用{\t}动态转变效果

\fax \fay \be \blur \xbord \ybord \xshad \yshad \iclip

(注：对于\clip 和\iclip，只有矩形框才能使用动态效果，绘画图形不能使用动态效果)

图形蒙板　　\iclip(<x1>,<y1>,<x2>,<y2>)

\iclip([<等级>,] <绘图命令>)

\iclip 与\clip 相似，所有参数与命令都一致，但效果相反，即在图形区域内的文字不可见，以外的文字可见.

注：当\iclip 和\clip 出现在同一行时，所有的图形效果都会当成"\iclip"处理.

**另注:**

所有的坐标值不再局限使用整数，可以使用小数.

一行的开始，结束时间，以及卡拉 OK 的时间可以使用更加精准的时间(原 SSA, ASS 都只精确到0.01秒)<未验证>

***内嵌的字体/图片编码***

SSA 所使用的内嵌字体/图片编码是 UUE-encoding 的一种形式. (因极少用到，以下仅引用原文)

It takes a binary file, three bytes at a time, and converts the 24bits of those bytes into four 6-bit numbers. 33 is added to each of these four numbers, and the corresponding ascii character for each number is written into the script file.

The offset of 33 means that lower-case characters cannot appear in the encoded output, and this is why the "filename" lines are always lower case.

Each line of an encoded file is 80 characters long, except the last one, which may be shorter.

If the length of the file being encoded is not an exact multiple of 3, then for odd-number filelengths, the last byte is multiplied by hexadecimal 100, and the most significant 12 bits are converted to two characters as above. For even-number filelengths, the last two bytes are multiplied by hexadecimal 10000, and the most significant 18 bits are converted to three characters as above.

There is no terminating code for the embedded files. If a new [section] starts in the script, or if another filename line is found, or the end of the script file is reached then the file is considered complete.

# 十三、自动化特效实现方式

用 ass 特效实现方法有很多。

**手打流**　就是直接用手调节，把所有复杂特效分解为最基本的代码，逐字，甚至一个字的不同时期，都用手写代码步步落实，全过程 Aegisub 实现

**套用 lua 流**　就是通过现有的 lua 直接套用，来实现一些 lua 里边包含的绚丽特效。（也支持自己用 lua 代码编写自己的特效，这个其实可以归到下一类的。）

**auto4 流**　就是我们平时所说的 AEG 自带自动化 apply karaoke template，全过程 Aegisub 实现。

**TCAX 流**　用一个叫 TCAX 的软件自动添加特效，支持用 python 语言自己编写特效，该方法对粒子效果支持较好。

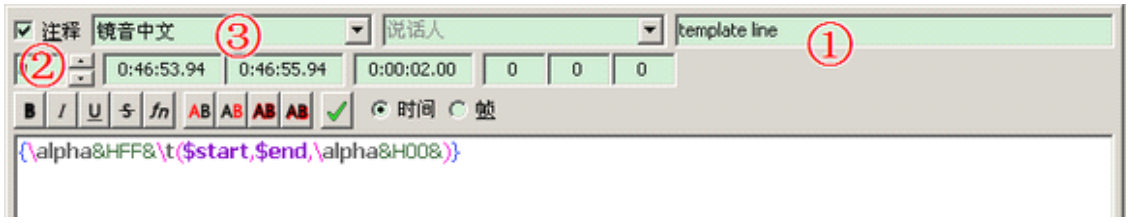
引言：你还在为逐字呈现而烦恼吗？在发现了字幕特效的呈现规律，并且掌握基本特效代码后，你依然为繁琐的手动机械添加而头疼？那么，auto4 流是一个很好的选择。

auto4 流能够在你发现字幕特效的呈现规律、异或对特效呈现有一个大致的想法后，帮助你快速添加特效。简单来说，就是利用了特效的共性。
既然有共性，那么必然是用一个模子做出来的。
那个模版（即 template），就是我们需要编写的内容。

预备知识：先讲讲特效的添加吧。

以逐字呈现（\alpha 递变）为例：

1、将欲添加特效的字幕打好 K 值
2、新建一个字幕行（即 右键→插入）
3、新字幕行文本处加入模板，在本例中模板为：
{\alpha&HFF&\t($start,$end,\alpha&H00&)}
4、①将该行字幕特效注明为"template line"（输入双引号内的内容，输入完成后记得回车！）,②把字幕行修改为“注释”属性。③将这行字幕的样式修改，改为与希望添加这个特效的字幕行的特效一样。如图：




（希望所有样式名为“镜音中文”的字幕行都用上逐字呈现这个特效）
5、点击 自动化→apply karaoke template

添加完成！

接下来是解释刚刚做的东西：

**1、Aegisub 在刚刚做了些什么？**
在 apply karaoke template 之后，aegisub 将待添加特效的字幕行中的所有{\k}都替换为你的模板了。你可以看看刚刚生成了什么？是不是很神奇？

**2、模板的含义：**
{\alpha&HFF&\t($start,$end,\alpha&H00&)}

\alpha&HFF& \t \alpha&H00& 相信大家看得懂，没问题。
但是$start $end 是什么？我们把他们称为内联变量。
在使用在 apply karaoke template 之后，其实 aegisub 还做了一件事
就是把$start $end 换成了一些实际的数值，不信看看刚刚生成的字幕行？

那么换成的数值到底是什么？

aegisub.cellosoft.com/docs/Karaoke_Templater_Reference:_Inline_variables

告诉我们，Aegisub 把$start 换成了每个对应的字的开始时间（单位毫秒，下同）
把$end 换成了每个对应的字的结束时间
不信看看刚刚生成的字幕行（第三次重复了 2333）

那么，是否只有$start 和 $end 这两兄弟呢？
其实还有很多呢~
一些常用的：
$ldur
对应行的持续时间（你可以看到，同一行的全部$ldur 返回的都是同一个值）
$x $y
对应字的 x 坐标 y 坐标
$si
对应字是该行中的第几个字。
懂英文的同学可以仔细研读这个网址最下面的表格，看看内联变量的家庭中还有那些成员。

aegisub.cellosoft.com/docs/Karaoke_Templater_Reference:_Inline_variables

总结起来， {\alpha&HFF&\t($start,$end,\alpha&H00&)} 的含义是：
对于该行的每个字（在已经通过打{\k}将每个字的时间均分的情况下），从每个字的开始时间到结束时间，透明度由全透变为无。

**关于每个字时间均分，我指的打 K 打成类似于这样子
{\K60}那{\K60}就{\K59}是{\K59}名{\K59}为{\K59}「{\K59}心{\K59}」{\K59}的{\K59}程{\K59}式 （开始时间 0:00:00.00 结束时间 0:00:06.50）

**3、template line 是什么？**
template 就是告诉 Aegisub 这一行是模板。
line 就是告诉 Aegisub 生成后的成品还是以行的形式表现出来。

这称为字幕行的修饰语 ，相当于表明了字幕行的身份，以及他的意图。

# 十四、Template 修饰语

不管是写 template 还是 code，都会带一些修饰语，像 template syl，code once，code line 等等，后面的 syl，line，once 就是修饰语，这个很重要，因为关系作用的范围。。。虽然有些可以省略，如果省略，像 template，就会默认为 syl，而 code 则会默认为 once。下面，讲几个比较常用的吧。

## 一、once

这个修饰语比较简单，不过只能用于 code 行，没见过有人写 template once 的。。。一般只有 code once，顾名思义就是这行代码在卡拉 ok 模板执行过程中只运行一次，并且这种代码一般都会写在其他 code 行或者 template 行的前面，并按照所声明的顺序运行。这行一般写的内容无非是自定义的函数，以便在后面的 template 里面直接调用。

例子：

Comment: 0,0:00:00.00,0:00:05.00,Default,,0000,0000,0000,code once,shape="m 0 0 b -4 -2 -11 -14 -1 -12 b -1 -12 0 -10 1 -12 b 10 -15 5 -2 0 0

"大家喜闻乐见的图形。。。。这样只会可以直接用 shape 了。。。。不用再不断的复制那串数字。

## 二、line

这个修饰语在code和template行都可以用，而且比较常见。不过提醒大家的是，如果line和template一块用的话，后面可以加上名字的，如果不加的话只是单纯用template line，则没有匹配的效果，只是对相应的样式有效，嗯。。。我语言表达不好，举个例子吧，比如一个没有名字的template line：

Comment: 0,0:00:00.00,0:00:05.00,Default,,0000,0000,0000,template line,{\r\t($start,$end,\bord0)}

这行很普通，就是单纯的改变音节的描边厚度而已。。。对所有该样式都会有效。。。但是如果是下面一种情况：

Comment:0,0:00:00.00,0:00:05.00,Default,,0000,0000,0000,

template linejumper,{\r\t($start,$mid,\frz-0.1)\t($mid,$end,\frz0}

这个是加了名字的template line ，它只会对被命名为jumper的才会有效。

如果line和code一块用，则没有命名的情况。。。。比如：

Comment: 0,0:00:00.00,0:00:05.00,Default,,0000,0000,0000,code line,fxgroup.funky = line.actor == "funky"

这个主要声明了说话人是funky，下面才会有效。。

## 三、pre-line

这个修饰语和 line 有点像，不过它只能和 template 一块用，也是可以加名字和不加名字的，完全和 line 一样，但是它不能和 code 一块用。

## 四、Syl

估计这个最常见了，没人不知道吧，这个 code 和 template 都能用，不可加名字。例子：

Comment: 0,0:00:00.00,0:00:05.00,Default,,0000,0000,0000,template syl,{\pos($x,$y)}

单纯指示位置。。。= =

## 五、Furi

furi 大概就是 furigana 的缩写吧，假名的意思，如果你打 k 值的内容是假名，也可以用这个。

## 六、All

顾名思义，所有都作用。。。。对整个字幕文件的每个音节。。。= =

## 七、Char

这个是 character 的缩写，用这个会对每个字产生作用，不一定是音节。。。

## 八、fx name

## 九、

这个修饰语我经常用，哈哈，奶大的东西可以控制每句话，每个字的效果的话，实际 aegisub 本身也能做到。。。例子

```
Comment:0,0:00:00.00,0:00:05.00,Default,,0000,0000,0000,
template syl fx drop,{\move($x,$y,$x,!$y+30!,$start,$end)}
```

上面的效果名字叫做 drop，要想实现这个效果，只要在{\k21\-drop}hi{\k10}gu{\k23}ra{\k22}shi {\k38}ga 里面加上\-drop,这个效果会作用整个一句话，如果写在中间则会对后面的产生效果，前面的不会。。。

## 十、keeptags

这个东西也很实用，主要是打 k 值的时候，有时候自己在原句中加了一些 tags 来控制颜色，字体，但是一旦应用了 template，这些东西就消失了，这时候加上 keeptags 则不会。。。貌似不能和 char 或 multi 一块用。。。例子

```
template line keeptags: {\r\t($start,!$start+1!,\frx40)\t(!$start+1!,$end,\frx0)}
karaoke: {\k21}hi{\k10}gu{\k23}ra{\k22}shi {\k38}ga {\k37\1c&H0000FF&}na{\k37}ku
```

## 十一、multi

这个东西估计也有人用过，这个主要是日语里面的汉字发音往往不是占一个音节的情况，甚至用多个假名注音，这时候会这么做，老实说这个算比较高级的用法了吧，占了多个音节，高亮就会有多个。。。我一直想试试，但是因为麻烦一直没用，可以自动实现汉字头顶注假名的效果哦~~（2001貌似讲过。。。）例子

```
template syl multi: {\an5\pos($scenter,$smiddle)\1a&HFF&\t($start,$end,\bord5\3a&HFF&)}
karaoke: {\k33}風{\k36}#{\k89}の{\k46}花{\k28}#{\k57}よ
```

风读作 kaze，两个音，多的一个用#来占位。。。。实际可以写成#|XX（写假名），应用之后假名会自动到汉字头上。。。一个音用一个#。。。。风这样的两个就应该用两个，不过这个例子没要实现那种效果。。。所以只是单纯用#占位而已。。。

## 十二、noblank

这个简单说一下，就是对空格无效。。。(特别提醒一下，不要小看空格)

## 十三、notext

这个经常在做图形，曲线啥的效果时候用，和 template 一块用。。。例子：

```
code once: sword_shape = "m 0 0 l 5 -5 l 5 -30 l 10 -30 l 10 -32 l 2 -32 l 2 -40 l -2 -40 l -2 -32 l -10 -32 l -10 -30 l -5 -30 l -5 -5 "
template syl notext noblank:
{\an5\move($scenter,!$smiddle-30!,$scenter,$smiddle,!$start-20!,$start)\p2}!sword_shape!
```

## 十四、repeat n，loop n

这两个都是指定效果的重复次数的。。。。repeat 还真没用过，我一直用 loop。。。。(貌似还用过 maxloop（）的函数，例子

template syl loop 4:
{\move($x,$y,!$x+math.random(-30,30)!,!$y+math.random(-30,30)!,$start,$end)\alpha&Hc0&\t($start,$end,\alpha&HFF&)}
这个效果就是从原来位置向上下左右大约30像素位置随机运动。。。。然后重复4次。。。。

# 十五、Template 内置变量

什么是内置变量？举个例子吧 Template syl

{\pos($x,$y)\t($start,$end,\blur20)}
{\k36}そ{\k20}の{\k14}手{\k24}伸{\k34}ば{\k28}し{\k26}て{\k137}も{\k150}{\k32}今{\k121}は {\k29}今{\k113}は{\k34}届{\k28}か{\k40}な{\k46}い{\k71}よ

自动化后{\pos($x,$y)}代表每个字自动会对应相应的 pos(位置坐标)，template syl
提示按音节拆分句子，\t($start,$end)会自动对应这一行开始时间和结束时间。

所以变量就是这个意思，下面来说说 Aegisub 2.1.9里有哪些变量以及代表的含义
注意，所有的变量使用要加上$字符，如：{\pos($center,$middle)}

## 1、行变量

[Events]
Format: Layer, Start, End, Style, Name, MarginL, MarginR, MarginV, Effect, Text

这是代表原句子每一行中的变量
layer：每一行句子的层数（layer）
style：每一行句子的样式（style）
actor：每一行句子的说话人（actor），注意，popsub 等老版打轴软件打轴 actor 会变成 Name，如上图，所以要手动改回来，并且 tcax 也不支持 Name 这一表达
margin_l, margin_r：每一行的对应的左边距离，右边距离
margin_v, margin_t, margin_b：每一行的对应的垂直，顶部，底部距离，特别的 margin_v 与 margin_t 相同
lstart, lend, ldur, lmid：每一行句子的开始时间，结束时间，持续时间，中间点时间，单位是毫秒，特别的 $ldur=!line.duration!
syln：每一行的音节总数，通俗的说就是这一行有多少个 k
lleft, lcenter, lright：这一行里最左侧的第一个字到左边的距离，中间到最左边距离（通常是 x 分辨率的一半），最右侧的最后一个字到左边的距离
ltop, lmiddle, lbottom：跟上面相似，每一行字的上端到视频顶端的距离，中间到视频顶端的距离，下端到视频顶端的距离
lx, ly：该行的 pos 坐标，受{\an}代码影响，通常用于{\pos}代码里，如{\pos($lx,$ly)}
lwidth, lheight:,这一行句子的像素宽度和高度，通常 lright= lleft+ lwidth

## 2、音节变量

首先，音节是什么？音节就是 k 所划分出的那一个或几个字，例如{\k24}夜の{\k37}空{\k52}辉{\k28}く{\k19}星，夜の是一个音节，空是一个音节等等
sstart, send, smid：这一行中每个音节的开始时间，结束时间，中间时间，单位是毫秒。音节由 k 划分，k 后面的时间决定变量时间。
sdur, skdur：k 的持续时间，sdur 单位是毫秒，skdur 单位是毫秒
si：这一行中的 k 的数目， 它代表一个计数器，从1开始，通常用于拆分时间，造成时间差，如!-1000+$si*100!,会造成一个100毫秒的时间差
sleft, scenter, sright：跟行变量里 lleft, lcenter, lright 类似只不过每一行改变成了每一个音节，即每一个音

节所指定位置的到左边的像素距离
sbottom, smiddle, stop：类似，就不说了
sx, sy：每个音节的坐标
swidth, sheight：每个音节的像素宽度和高度

## 3、全局自动变量

最常用的就在这里了，适用于行里和音节里
start, end, mid：音节或行的开始，结束，中间点时间
dur, kdur：行或音节的持续时间，前一个单位是毫秒，后一个是厘喵
i：行或音节里的 k 数目，也是一个计数器
left, center, right：行或音节所指定位置的到左边的像素距离
top, middle, bottom：行或音节所指定位置的到顶端的像素距离
x, y：行或音节的坐标
width, height：行或音节像素宽度和高度

# 十六、retime 函数

例子：

Comment:0,0:00:00.00,0:00:00.00,op-jp,,0000,0000,0000,templatesyl noblank,
!retime("start2syl",-200+syl.i*30,0)!{\an5\fs1\pos($center,$middle)\t(0,250,\fs34)}

Comment:0,0:00:00.00,0:00:00.00,op-jp,,0000,0000,0000,templatenoblank,
!retime("syl",0,0)!{\an5\pos($center,$middle)\t(0,!$dur*0.2!,\rnd20)\t(!$dur*0.2!,!$dur*0.4!,\rnd0)\t(!$dur*0.4!,!$dur*0.6!,\rnd10)\t(!$dur*0.6!,!$dur*0.8!,\rnd20)\t(!$dur*0.8!,0,\rnd1)}

Comment:0,0:00:00.00,0:00:11.64,op-jp,,0000,0000,0000,templatesyl noblank,
!retime("syl2end",0,200+syl.i*30)!{\an5\pos($center,$middle)\t(!line.duration-200!,!line.duration!,\fscx200\fscy200\bord0\blur20\alpha&HFF&)}

Comment:0,0:00:07.01,0:00:07.98,op-jp,,0000,0000,0000,templatechar,
!retime("line",math.random(-300,500),math.random(1000,1500))!{\moves4(!$scenter+math.random(-50,50)!,!$smiddle+math.random(-50,50)!,!$scenter+math.random(-60,60)!,!$smiddle+math.random(-60,60)!,!$scenter+math.random(-60,60)!,!$smiddle+math.random(-60,60)!,!$scenter+math.random(-50,50)!,!$smiddle+math.random(-50,50)!)\an5\fsc!math.random(5,20)!\blur6\fad(300,500)\p1}m21 16 b 17 16 17 23 21 23 b 25 23 26 16 21 16 m 24 19 l 43 20 l 24 20 m 18 19 10 20 l 18 20 m 20 16 l 21 0 l 22 16 m 20 23 l 21 40 l 22 23

Comment:0,0:00:00.00,0:00:00.00,op-jp,,0000,0000,0000,templatesyl noblank notext loop5,
!retime("line",line.duration,math.random(4000,8000))!{\moves4($center,$middle,!$center+math.random(-70,100)!,!$middle+math.random(-80,-50)!,!$center+math.random(-80,50)!,!$middle+math.random(-40,80)!,!$center+math.random(-70,50)!,!$middle+math.random(-60,60)!)\\an5\fsc!math.random(5,20)!\blur6\fad(0,!math.random(100,700)!)\p1}m21 16 b 17 16 17 23 21 23 b 25 23 26 16 21 16 m 24 19 l 43 20 l 24 20 m 18 19 10 20 l 18 20 m 20 16 l 21 0 l 22 16 m 20 23 l 21 40 l 22 23

Comment:0,0:01:30.56,0:01:37.33,op-jp,NTP,0000,0000,0000,karaoke,{\k17}目 {\k17}の {\k42}前 {\k131}に {\k50}　映{\k64}る{\k58}　問{\k21}え{\k15}に{\k23}い{\k65}つ{\k30}も {\k46}曖{\k39}昧{\k69}で
Comment:0,0:01:37.50,0:01:45.06,op-jp,NTP,0000,0000,0000,karaoke,{\k18}そ{\k21}の{\k36}言{\k24}葉{\k84}で{\k42}　何{\k31}か{\k104}が{\k24}　か{\k33}わ{\k20}る{\k34}な{\k24}ん{\k26}て{\k48}　思{\k23}い{\k21}も{\k27}し{\k37}な{\k34}く{\k53}て

**retime：用于拆分时间，重要不可缺的一个函数**

格式：retime(mode,调整时间 a,调整时间 b)
该函数通常在一个 template 或 code 中使用一次或多次。它可以以不同的方式来调整输出的每个词的开始结束时间，在做特效时经常用到。
mode 参数用来决定如何改变每个词的开始结束时间，它必须是下面的一个。因为它是一个字符串，所以须用半角引号将模式的名称括起来。
调整时间 a 和调整时间 b 参数可以在选用相应的 mode 中对开始结束时间进行调整，通常是一个以毫秒为单位的数，当然也可以用一些算式。如果数值大于0，是向后延迟，小于0则是提前。

下面开始详解

| | |
|---|---|
| abs 或 set | 两个参数功能相同。a 和 b 会从时间轴开始的时间（0:00:00.00）来确定这一词的开始结束时间，与行的开始结束时间无关。（很少用） |
| Preline | 提前行的开始时间 这个参数是为了在行开始前，让一些词先开始。结束时间固定为行的开始时间，所以 b 固定为0，可以省略。 |
| Line | 使用每一行的开始结束时间。a 和 b 在行的开始结束时间基础上可以改变词的开始和结束时间。 |
| start2syl | 是每个词的开始时间为行的开始时间，结束时间为每个词 K 的开始时间。 |
| Presyl | 与 preline 类似，只是结束时间换成了词 K 的开始时间。 |
| syl | 指的是 K 开始时间点到 K 结束时间点的一段时间。a 和 b 在词 K 的开始结束时间基础上可以改变词的开始和结束时间。 |
| Postsyl | 每个词开始时间为 k 的结束时间，结束时间为 k 的结束时间加上调整时间 b。a 和 b 可调整时间。 |
| syl2end | 每个词的开始时间为词 K 的结束时间，结束时间为行的结束时间，类似于 start2syl。 |
| Postline | 每一行开始时间为该行的结束时间，结束时间为该行开始时间加上调整时间 b。a 和 b 可调整时间。 |

## 例子分析

### 第一段代码：

templatesylnoblank,!retime("start2syl",-200+syl.i*30,0)!{\an5\fs1\pos($center,$middle)\t(0,250,\fs34)}

**template syl noblank:**启用 template 自动化，修饰语为 syl,对于空格无效。意思是启动自动化模板，对卡拉 ok 的每个音节进行自动化（这个根据 k 来决定），对于歌词中的空格无效

**!retime("start2syl",-200+syl.i*30,0)!:**启用 retime 函数，函数名为 start2syl（函数意思看上面 retime 解释），开始时间调整为-200+syl.i*30，结束时间不变，即为每个词的 k 的开始时间。关于-200+syl.i*30的解释：-200为负的200毫秒，syl.i 是个计数器函数，表示行中拆分字符的个数，简单来说，用 k 来分的字符有多少个，就有多少个 i 例如第一句第一个词【目】，则开始时间为-200+1*30=-170，就是说提前170毫秒，那么开始时间就是0:01:30.56减去170毫秒（17厘秒）=0:01:30.(56-17)=0:01:30.39，结束时间为第一个 k 的开始时间0:01:30.56第一句第二个词【の】，则开始时间为-200+2*30=-140，就是说提前140毫秒，那么开始时间就是0:01:30.56减去140毫秒（14厘秒）=0:01:30.(56-14)=0:01:30.42，结束时间为第二个 k 的开始时间【0:01:30.(56+17)】=0:01:30.73...以此类推对于所有的运算要用!运算式!半角感叹号括起来，这样 Aegisub 才知道你要进行数学运算，不扩的话按字符处理这样做的远因是为了提前每个词的显示时间用于制作入场特效，以及制造出时间差，并且为第一个字提供特效时间

**接着分析 ass 代码{\an5\fs1\pos($center,$middle)\t(0,250,\fs34)}：**我来逐一解释\an5，设定定位点，特效定位点用\an5，普通字幕的大家都知道用\an2\fs1，设置字体大小为1\pos($center,$middle)，获取每个字符的默认位置\t(0,250,\fs34)动态特效，在每个字符开始~250毫秒之内字体大小动态变为34总结一下，初始字体大小设置为1，然后变大为34，现在能想象出是什么入场特效了吧？入场特效就是字幕一个接一个有无到有显现出来

### 第二段代码：

Comment:0,0:00:00.00,0:00:00.00,op-jp,,0000,0000,0000,templatenoblank,!retime("syl",0,0)!{\an5\pos($center,$middle)\t(0,!$dur*0.2!,\rnd20)\t(!$dur*0.2!,!$dur*0.4!,\rnd0)\t(!$dur*0.4!,!$dur*0.6!,\rnd10)\t(!$dur*0.6!,!$dur*0.8!,\rnd20)\t(!$dur*0.8!,0,\rnd1)}

**template noblank：**启用 template 自动化 修饰语为 noblank，即特效对于空格无效，特别的：template = template syl 噗~~

**!retime("syl",0,0)!：**启用 retime 函数，函数名为 syl，0,0代表不增加并且不减少开始以及结束时间

**接着分析 ass 代码：**

{\an5\pos($center,$middle)\t(0,!$dur*0.2!,\rnd20)\t(!$dur*0.2!,!$dur*0.4!,\rnd0)\t(!$dur*0.4!,!$dur*0.6!,\rnd10)\t(!$dur*0.6!,!$dur*0.8!,\rnd20)\t(!$dur*0.8!,0,\rnd1)}{\an5\pos($center,$middle)}：设置定位点，获取每个字的中间默认位置{\t(0,!$dur*0.2!,\rnd20)}：$dur 为 k 的持续时间，,!$dur*0.2!为0.2个持续时间，就是对时间进行拆分了5分之1分，{\rnd20}边界象素变形，数字越大变形越大，可自己看看效果{\t(!$dur*0.2!,!$dur*0.4!,\rnd0)}，在5分之1~5分之2的时间里取消边界像素变形{\t(!$dur*0.4!,!$dur*0.6!,\rnd10)}.....类推吧.....类推能想象到效果嘛?这个特效称之为”表现特效“就是唱到那个歌词时所凸显的特效，特效效果就是字的不规则变形。

**第三段代码：**

Comment:0,0:00:00.00,0:00:11.64,op-jp,,0000,0000,0000,templatesylnoblank,!retime("syl2end",0,200+syl.i*30)!{\an5\pos($center,$middle)\t(!line.duration-200!,!line.duration!,\fscx200\fscy200\bord0\blur20\alpha&HFF&)}

**template syl noblank**，这个不解释了，上面说过了！

**retime("syl2end",0,200+syl.i*30)!**启用时间拆分的 syl2end 函数，省么意思 retime 的解释里有200+syl.i*30时间的拆分计算，就是第一个词结束时间加上200+1*30=230毫秒，第二个加上200+2*30=260毫秒以此类推

**下面解释 ass 代码**

{\an5\pos($center,$middle)\t(!line.duration-200!,!line.duration!,\fscx200\fscy200\bord0\blur20\alpha&HFF&)}\an5\pos($center,$middle)，这个不解释了 t(!line.duration-200!,!line.duration!，line.duration 为一行字的持续时间，但我们运用了 syl 拆分字符，所以一行就一个字，时间就是 k 的结束时间到这一行的结束时间，duration-200!,!line.duration! 就造成了在结束时有,200毫秒的特效持续时间\fscx200\fscy200\bord0\blur20\alpha&HFF&：同时在 x，y 轴扩大为原来字体的两倍，面积就增大了2*2=4倍。\bord0\blur20，边框厚度变为0，高斯模糊为20，这样造成的特效就是字体模糊雾化，blur 值越大效果越强，\alpha&HFF&透明代码，造成淡出特效

好吧，能想象出什么效果了么，如果想象不出来，那么说明你的 ass 功底严重不足....噗~就是每个字逐个快速放大消逝~这个称之为**“退场特效”**。

**第四段代码：**

Comment:0,0:00:07.01,0:00:07.98,op-jp,,0000,0000,0000,template char,!retime("line",math.random(-300,500),math.random(1000,1500))!{\moves4(!$scenter+math.random(-50,50)!,!$smiddle+math.random(-50,50)!,!$scenter+math.random(-60,60)!,!$smiddle+math.random(-60,60)!,!$scenter+math.random(-60,60)!,!$smiddle+math.random(-60,60)!,!$scenter+math.random(-50,50)!,!$smiddle+math.random(-50,50)!)\an5\fsc!math.random(5,20)!\blur6\fad(300,500)\p1}m21 16 b 17 16 17 23 21 23 b 25 23 26 16 21 16 m 24 19 l 43 20 l 24 20 m 18 19 l0 20 l 18 20 m 20 16 l 21 0 l 22 16 m 20 23 l 21 40 l 22 23

**template char**：启用 template，修饰语是 char（character 的缩写），意思是对每个字执行特效而不是 k 拆分出的音节

**!retime("line",math.random(-300,500),math.random(1000,1500))!**，启用retime函数，函数类型为line（行），注意,此时的开始时间与结束时间均为歌词每一行的开始于结束时间math.random(-300,500),math.random(1000,1500),随机调整每个行的开始时间与结束时间，math.random是产生随机数的函数，(-300,500）为随机数范围

**下面解释 ass 代码**

{\moves4(!$scenter+math.random(-50,50)!,!$smiddle+math.random(-50,50)!,!$scenter+math.random(-60,60)!,!$smiddle+math.random(-60,60)!,!$scenter+math.random(-60,60)!,!$smiddle+math.random(-60,60)!,!$scenter+

math.random(-50,50)!,!$smiddle+math.random(-50,50)!)\an5\fsc!math.random(5,20)!\blur6\fad(300,500)\p1}m21 16 b 17 16 17 23 21 23 b 25 23 26 16 21 16 m 24 19 l 43 20 l 24 20 m 18 19 10 20 l 18 20 m 20 16 l 21 0 l 22 16 m 20 23 l 21 40 l 22 23\moves4(x1,y1,x2,y2,x3,y3,x4,y4,t1,t2)
4次贝塞尔曲线，x1,y1,x2,y2,x3,y3,x4,y4为控制点坐标，t2，t2为控制时间，与move代码类似四个p点即代表x1,y1,x2,y2,x3,y3,x4,y4为控制点坐标!$smiddle+math.random(-50,50)!,获取默认中心坐标并且加上随机获得的随机数即为计算后x1坐标!$smiddle+math.random(-60,60)!，获取默认中心坐标并且加上随机获得的随机数即为计算后y1坐标以此类推获取最后一个定位坐标没有写t1，t2时间，默认就是此行时间
{\fsc!math.random(5,20)!\blur6\fad(300,500)\p1}m 21 16 b 17 16 17 23 21 23 b 2523 26 16 21 16 m 24 19 l 43 20 l 24 20 m 18 19 l 0 20 l 18 20 m 20 16 l 21 0 122 16 m 20 23 l 21 40 l 22 23：\fsc!math.random(5,20)!为随机全局缩放，百分比为5~20中的一个，blur6，\fad(300,500)就不要解释了吧，
{\p1}m 21 16 b 17 16 17 23 21 23 b 25 23 26 16 21 16 m 24 19 l 43 201 24 20 m 18 19 l 0 20 l 18 20 m 20 16 l 21 0 l 22 16 m 20 23 l 21 40 l 22 23，
矢量绘图代码，图形是个十字交叉的星星（？反正看起来像）
\fsc!math.random的目的是缩放此图形，因为为附加特效，不能够比子还大对吧~还有{\p1}代码，原始缩放等级不知道去看ass代码详解这个附加特效效果就是每行有多少字产生多少个星星，按照4次贝塞尔曲线移动

**第五段代码：**

Comment:0,0:00:00.00,0:00:00.00,op-jp,,0000,0000,0000,
template syl noblank notext loop5,
!retime("line",line.duration,math.random(4000,8000))!{\moves4($center,$middle,!$center+math.random(-70,100)!,!$middle+math.random(-80,-50)!,!$center+math.random(-80,50)!,!$middle+math.random(-40,80)!,!$center+math.random(-70,50)!,!$middle+math.random(-60,60)!)\\an5\fsc!math.random(5,20)!\blur6\fad(0,!math.random(100,700)!)\p1}m21 16 b 17 16 17 23 21 23 b 25 23 26 16 21 16 m 24 19 l 43 20 l 24 20 m 18 19 10 20 l 18 20 m 20 16 l 21 0 l 22 16 m 20 23 l 21 40 l 22 23

**template syl noblank notext loop 5**：前面大家都懂，按音节产生每一行，有多少音节就有多少行，对文字，空格无效，与上面的代码正好相反， loop 5表示此代码循环执行5次，，后面的和第四段代码差不多，就是时间控制上有不同，为了让星星飞的更久一点~哈

## retime

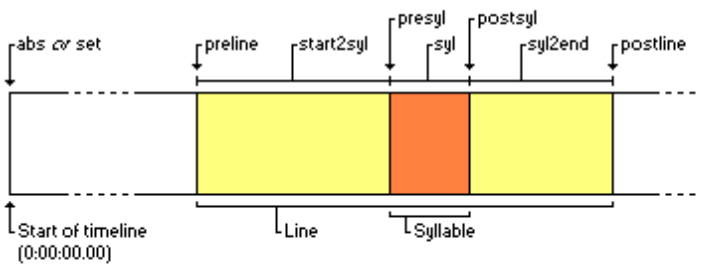
Synopsis: retime(mode, startadjust, endadjust)

**retime(mode, addstart, addend)**

Change the timing of the generated line. The *mode* parameter controls what the new timing is relative to.
The *addstart* and *addend* parameters adjust the start and ending time additionally.

The diagram to the right shows the possible values for *mode* and what they mean. For those values described with an arrow, the start and end times of the new line will be equal unless you also adjust them with *addstart* and *addend*.

There is one additional value for *mode*, "sylpct". That one interprets *addstart* and *addend* as percentages instead. Specifically it does:
start = sylstart + addstart×syldur/100
end = sylstart + addend×syldur/100



# 十七、remember & recall 函数

说白了，这个函数就是允许你将一个特定的数值赋给你声明的变量。
之后你就可以通过调用这个变量来获取原来的数值。

**例一： template line,{!remember("a",50)!}{\fs!recall.a!\fsp!recall.a!}**

套用至： {\k29}清{\k29}明{\k29}时{\k29}节
效果：{50}{\fs50\fsp50}清{50}{\fs50\fsp50}明{50}{\fs50\fsp50}时{50}{\fs50\fsp50}节
意思即将50赋值给a（要用双引号引起来），之后只要调用a（调用格式→recall.a）就可以获得50这个值
注意：!remember("a",50)!要用大括号框住的原因是，
如果撤掉大括号，数值就会显示在字幕内容之上。

当然，我更习惯这样做：
template line,{\fs!remember("a",50)!\fsp!recall.a!}
套用至： {\k29}清{\k29}明{\k29}时{\k29}节
效果：{\fs50\fsp50}清{\fs50\fsp50}明{\fs50\fsp50}时{\fs50\fsp50}节
看起来简单，美观。

**例二：template line,{!remember("qingming","xiayu")!}!recall.qingming!**

套用至：{\k29}清{\k29}明{\k29}时{\k29}节
效果：{xiayu}xiayu清{xiayu}xiayu明{xiayu}xiayu时{xiayu}xiayu节
意思是将xiayu这五个字符赋值给名为qingming的变量，
之后只要通过!recall.qingming!调用，Aeg娘就会显示xiayu这串字符

此例说明，变量不仅可以记录数值，还可以记录字符，不过字符要用双引号引起来。

温馨提示：
1、用remember函数记录的变量只能在本行内调用。诸如
template line,{!remember("a",50)!}
template line,{!recall.a!}
隔行的调用是不会成功的。所以请尽量把代码写在一行之内。

2、你现在可能觉得这个函数没什么作用。但随着研究的深入，
特别是在引入随机数后，你会发现他的用处会很大！

# 十八、random 函数

想必这是Aegisub里边最重要的函数了吧。
模板是固定的，你套用什么代码就会出现什么特效，只听你的指挥
但随机数却是活的，当你输入一个随机数的范围，
Aeg娘却会为你生成它随机想到的数值！

随机数在特效的各个方面都获得了广泛地应用，且听我慢慢道来。

**例：template syl,{\pos($x,$y)\fs!math.random(30,50)!}**

套用至：{\k29}清{\k29}明{\k29}时{\k29}节
效果：{\pos(613,710)\fs30}清{\pos(631,710)\fs49}明{\pos(649,710)\fs37}时{\pos(667,710)\fs32}节
关注!math.random(30,50)!的部分，他的意思是，生成一个30~50之间的随机数。
正如你所见，Aeg娘在套用模板时将!math.random(30,50)!换成了30~50之间的随机数。

此例意思为字体大小为30~50之间，随机。

注意，math.random是不能生成小数的，所以你可以采用这样的方法
!math.random(35,57)/10!

温馨提示：
当Aeg娘生成的随机数不理想时，你是否会重新点一次apply karaoke template
来调教她，生成一个理想的随机数？结果发现，生成的结果是一模一样的！

这和随机数的生成算法有关，只要你定下生成的值域，其实所谓的随机数已经定下来了。
Aeg娘也是根据一定的算式生成随机数的。

为了解决这个问题，教一些超纲的东西（代码有两行）：
code,math.randomseed(60)
template line,{\fs!math.random(30,50)!}
其精髓是上面一行，其修饰语为code，
为Aeg娘种下了大小为60的随机数种♂子，
随着种♂子数值的变更，!math.random(30,50)!生成的随机数也不一样。
你就可以更改种♂子的数值（一般在生成随机数的范围的几倍以内，不要太大），
来获取理想的数（孩）值（子）了。

# 十九、其他函数

1、relayer
语法:relayer(newlayernumber)
作用:改变套用该行模板所产生新行的层数,如果要使所产生行的层数都是一样的，可以只需在模板行设置一下模板行的层数，所有套用该模板行所产生的新行，会自动复制模板行的层数。
例子:
Comment: 0,0:00:00.00,0:00:00.00,Default,,0,0,0,template syl,!relayer(syl.i)!

2、restyle
语法:restyle(newstylename)
作用:改变套用该行模板所产生新行的样式名称，注意只是改名称，不是将原样式复制以后重新命名，同时line.styleref也将指向新样式，新改的样式名称需提前在样式库里设置好。在模板运行时改为新样式，模板所套用的大小与位置信息还是由原来的样式决定。
例子:无

3、maxloop
语法:maxloop(maxloopnumber)
作用:动态控制模板行的循环次数
例子:
Comment:0,0:00:00.00,0:00:00.00,Default,,0,0,0,Template syl,
!maxloop(syl.width2*line.styleref.outline)!{\clip(!line.left+syl.left-line.styleref.outline+j-1!,0,!line.left+syl.left-line.styleref.outline+j!,!meta.res_y!)\an5\move(!line.left+syl.center!,!line.middle!,!line.left+syl.center!,!line.middle+math.random(-20,20)!,$start,$end)\shad0}

maxloop(syl.width + 2*line.styleref.outline)
控制循环次数，为syl的像素宽度与两倍边框值的和
\clip(!line.left+syl.left-line.styleref.outline+j-1!,0,!line.left+syl.left-line.styleref.outline+j!,!meta.res_y!)
做该字的像素宽度个矩形切割（矩形宽度为一，高度为视频的高度）
line.left+syl.left-line.styleref.outline 为该所有syl像素的最左边(为什么这里line.left还要加上syl.left因为syl.left是一个相对值，相对于line.left)
meta.res_y 视频纵向长度
\move(!line.left+syl.center!,!line.middle!,!line.left+syl.center!,!line.middle+math.random(-20,20)!,$start,$end)
纵向垂直随机移动

4、loopctl
语法:loopctl(newj, newmaxj)
作用：动态改变控制循环的变量j和maxj。j与maxj，为使用修饰符loop时，j为当前循环的次数，maxj为最大的循环次数也就是loop后面所跟的数字，一般用不到
例子:
1：Comment: 0,0:00:00.00,0:00:00.00,Default,,0,0,0,
template syl noblank notext loop 10,!j! !maxj! !loopctl(j+1,maxj)!
2:Comment: 0,0:00:00.00,0:00:00.00,Default,,0,0,0,template syl noblank notext loop 10,!loopctl(j+1,maxj)!

!j! !maxj!

两个例子都是将j加一以后重新赋值给j，事实上运行时，j的值会每次加2，因为本身执行时计数会回加1。注意上面两个结果的例子的结果是不一样的。

# 二十、自动化添加文字顶注释

aegisub自动化后大家经常会忽视掉一个自动生成的style

```
[V4+ Styles]
Format: Name, Fontname, Fontsize,
Style: Default-furigana,Arial,25,
Style: Default,Arial,50,&H00FFFF
```

Default-furigana,这东西就可以自动表顶注,实现时使用furi
template furi
样式是: x|xx
竖线打出方法:键盘回车键上那个按键看见了没? shift+它

上面的特效例子增加三句即可

```
Comment:0,0:00:00.00,0:00:05.00,Default,,0,0,0,Template syl,
!retime("start2syl",-100+syl.i*30,0)!{\pos($scenter,$smiddle)\an5\alpha&HFF&\t(0,500,\alpha&H00&)}
Comment:0,0:00:05.00,0:00:05.00,Default,,0,0,0,template syl multi,
!retime("syl",0,300)!{\pos($scenter,$smiddle)\an5\bord5\blur5\t(0,50,\fscx130\fscy130)\t(50,$dur,\fscx100\fscy100\bord0\blur0)}
Comment:0,0:00:05.00,0:00:05.00,Default,,0,0,0,template syl,
!retime("syl2end",0,100+syl.i*30)!{\pos($scenter,$smiddle)\an5\alpha&H00&\t(!line.duration-500!,!line.duration!,\alpha&HFF&)}
Comment:0,0:00:00.00,0:00:05.00,Default,,0,0,0,template furi,
!retime("start2syl",-100+syl.i*30,0)!{\pos($scenter,$smiddle)\an5\alpha&HFF&\t(0,500,\alpha&H00&)}
Comment:0,0:00:05.00,0:00:05.00,Default,,0,0,0,template furi,
!retime("syl",0,300)!{\pos($scenter,$smiddle)\an5\bord5\blur5\t(0,50,\fscx130\fscy130)\t(50,$dur,\fscx100\fscy100\bord0\blur0)}
Comment:0,0:00:05.00,0:00:05.00,Default,,0,0,0,template furi,
!retime("syl2end",0,100+syl.i*30)!{\pos($scenter,$smiddle)\an5\alpha&H00&\t(!line.duration-500!,!line.duration!,\alpha&HFF&)}
Dialogue: 0,0:00:05.00,0:00:07.00,Default,,0,0,0,karaoke,{\k33}風|kaze{\k33}の{\k46}花|hana{\k57}よ
```


然后,大家拿出自己的特效脚本 安照方法修改
有兴趣可以自己替换{\k50}youkaze|四月老师组长大人
如果生成如下效果,说明你成功了

# 二十一、ASS 滚动字幕

在Aegisub中可以如下设定（在特效栏输入Banner;2;0;50）
Banner;2;0;50 第一个数字代表速度；数字越小速度越快
第二个数字0代表字幕从右往左走，1代表从左往右走
第三个数字代表字幕滚动到屏幕边缘的透明度，实践证明50为好

（以下不知道原文具体出自何处）
用记事本打开ASS文件.找到你之前做滚动的那句话.我们这里用塔塔曾经用过的例子来看.在ASS中会出现如下的效果.

Dialogue: 0,0:00:00.00,0:05:00.00,*Default,NTP,0000,0000,0000,Banner;15, {\an8}本字幕由XX字幕组倾情制作，仅限于用做学习交流，请勿用于商业用途，如发生纠纷，与本字幕组及其所属论坛无关。

这里解释一下
0:00:00.00,0:05:00.00　这个就是你做的时间轴,前半段是出现时间,后半段是结束时间.根据具体情况,可做调整.时间一到,滚动字幕不管有没滚完都会强制结束,这点请注意.

*Default,NTP,0000,0000,0000,　用我们老师的话来说."不要去想它...."这里不要乱动就行.

Banner;15　15代表的是速度,数字越小,速度越快,0的话,基本就不是人看的东西了.一般选用10-15左右的速度,在20-40秒能滚完.有人会问了,之前副组写的教程里,不是这样的格式啊,而是Banner;15;0;50 啊.没错,不过这是他手动修改好了的.我们在一开始打开ASS文件的时候,看到的只有Banner;15.那么Banner;15;0;50又是什么意思?

Banner;15;0;50　其中,第一个数字15代表的就是速度.我们不再提它.第二个0代表的是从右向左滚动,改成1的话,会变成从左向右滚.最后一个50代表的是滚动字幕在屏幕边缘的透明度.一般用50就好,别去改它.特别需要注意的是,这几个数字之间,必须要有;隔开.

最后出现的效果就是这样.
Dialogue: 0,0:00:00.00,0:05:00.00,*Default,NTP,0000,0000,0000,Banner;15;0;50, {\an8}{\fs30}本字幕由XX字幕组倾情制作，更多精彩游戏视频请关注XX字幕组 字幕组交流群：XXXXXXXX 片源：XX，翻译：XX，校润：XX, 后期：XX

诶?是不是多了什么?没错,在置顶代码之后多了一个代码{\fs30},这个是强制规定字号的代码.因为一般上屏滚动用的字号会与整体字幕有所区别,所以会用到它.30代表的就是字号,根据需要可任意更改.

总体回顾

首先是两个代码
{\an8}〈---置顶代码，作用是将此代码之后的文字放置于屏幕顶端
{\fs30}〈---字号代码，作用是改变字号的大小。

接着，PUP的工作很简单. 插入代码-->右键[滚动]-->滚动方式选择.

最后是ASS里的工作. 手动修改Banner;15;0;50　其中15为速度,0为滚动方式,50为滚动字幕在屏幕边缘的透明度.

# 二十二、自动化添加代码

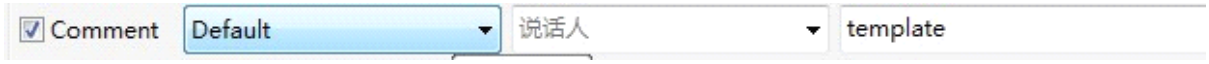
动漫OP ED以及MV中,中文字幕如果想加特效的话需要单独分k值,懒人如果不想那么麻烦就去用tcax tools文件夹中的KASS.exe工具,它可以自动分line的k值,如果不加特效的话一般会加入fad等代码效果来配合特效字幕的in和out方式,有时候也会加bord blur来使中文字幕更加漂亮.所以如果字幕少还好,可以一句句在ASS里每行左边复制粘贴,如果碰到懒人,如果同时又碰到line超多的那种,你想一句句复制也没人说啥,但是如果我们有其他办法可以方便的达到目的的话....

例子:

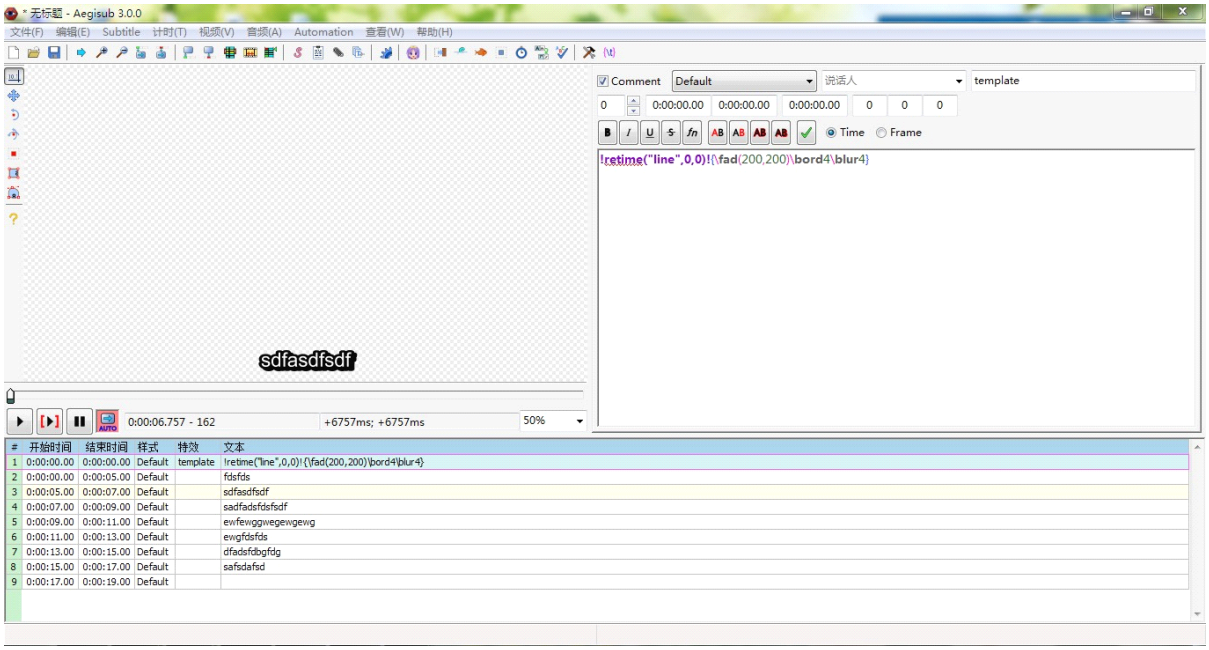
| # | 开始时间 | 结束时间 | 样式 | 文本 |
|---|---|---|---|---|
| 1 | 0:00:00.00 | 0:00:05.00 | Default | fdsfds |
| 2 | 0:00:05.00 | 0:00:07.00 | Default | sdfasdfsdf |
| 3 | 0:00:07.00 | 0:00:09.00 | Default | sadfadsfdsfsdf |
| 4 | 0:00:09.00 | 0:00:11.00 | Default | ewfewggwegewgewg |
| 5 | 0:00:11.00 | 0:00:13.00 | Default | ewgfdsfds |
| 6 | 0:00:13.00 | 0:00:15.00 | Default | dfadsfdbgfdg |
| 7 | 0:00:15.00 | 0:00:17.00 | Default | safsdafsd |
| 8 | 0:00:17.00 | 0:00:19.00 | Default | |

如上图字幕(随意打的),我们的目标是让每行出现{\fad(200,200)\bord4\blur4}效果

字幕最上面一行右键,插入一空白行(之前),自行键入代码



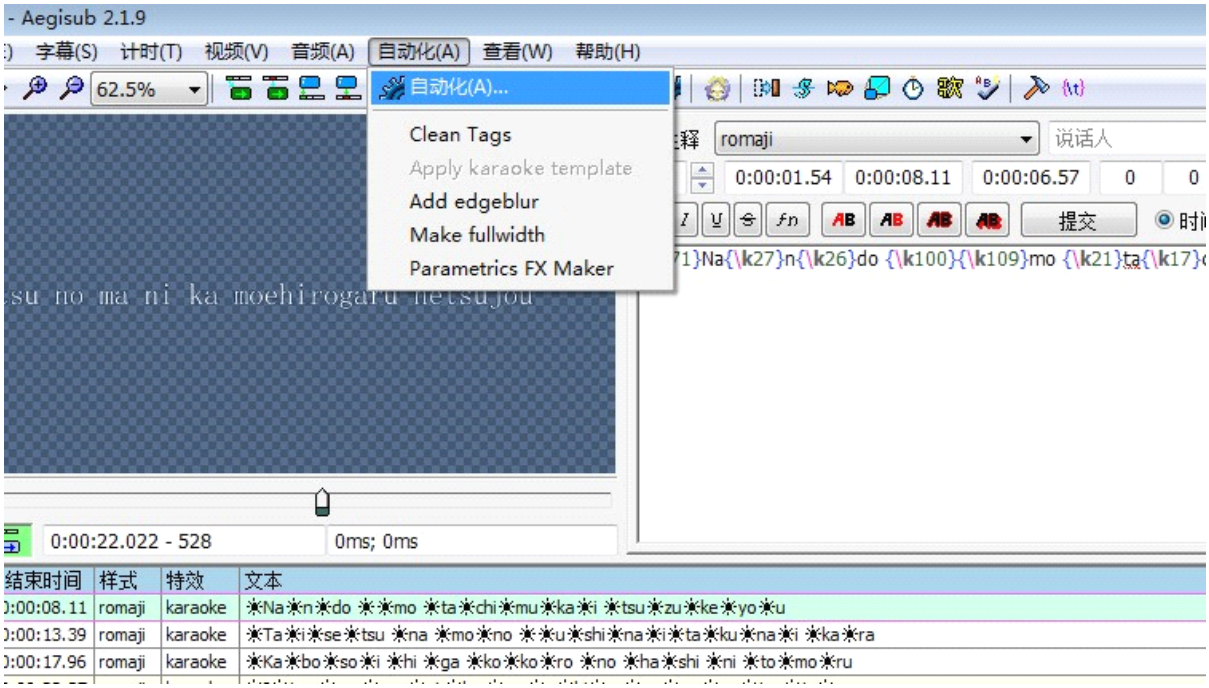

最左面不要忘记打钩[凡是要引用特效是都要打钩],template,!retime("line",0,0)!{\fad(200,200)\bord4\blur4}



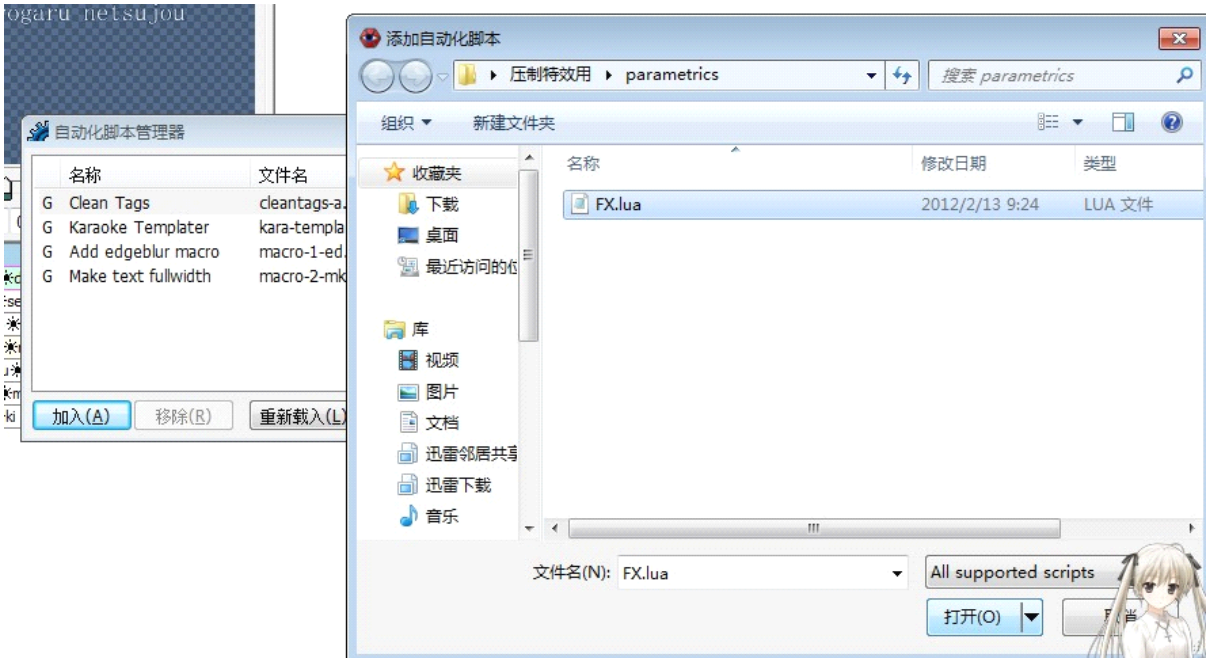

然后 自动化生成.之后预览下就会得到我们所需的效果

# 二十三、LUA 代码应用

首先，打开 aeg，如图：

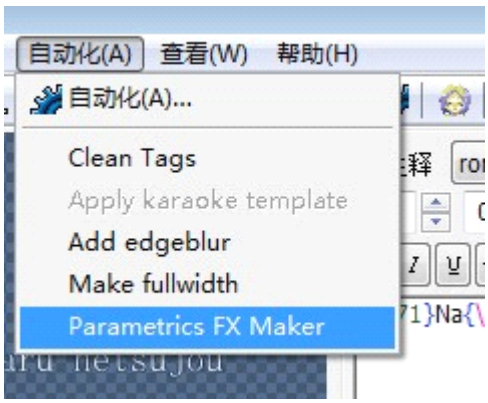



自动化（说白了，就是auto4的产物吧，吐槽而已。。。。- -）/加入,把那个lua加进来就好。。。。

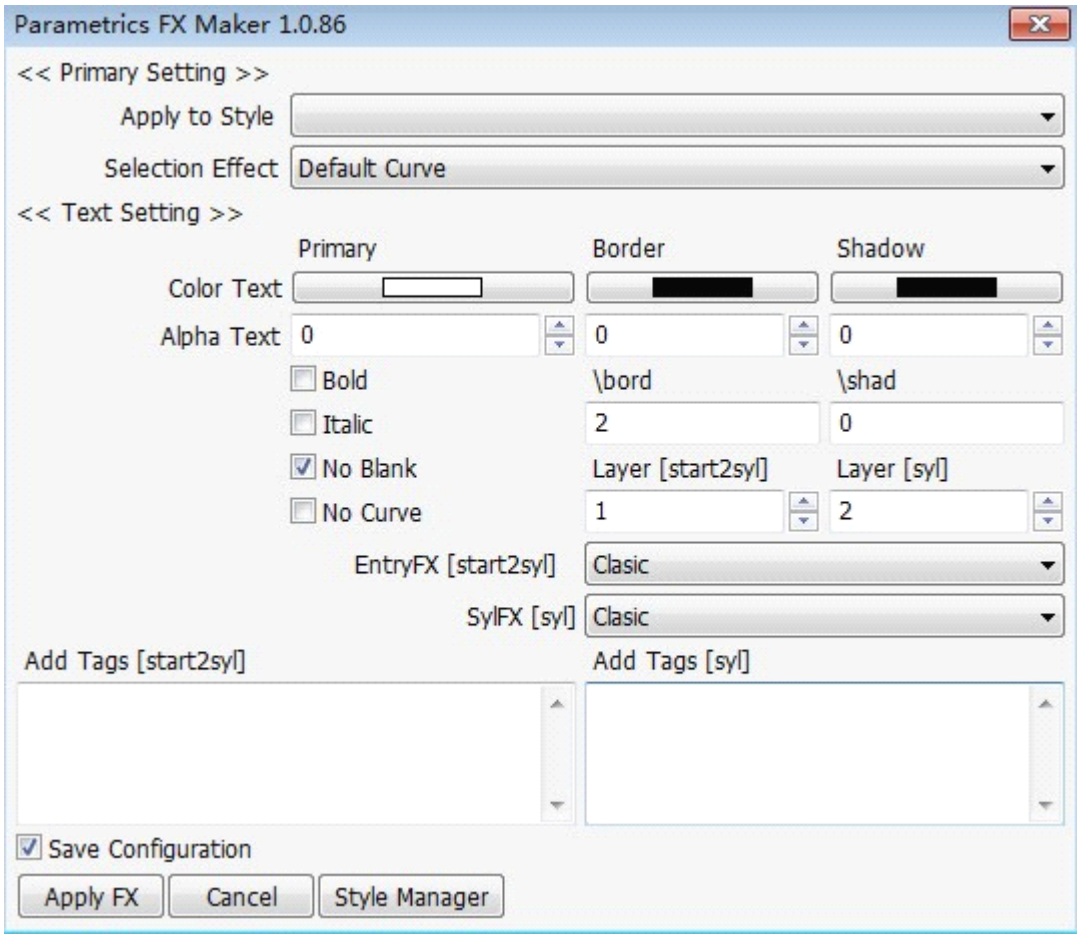




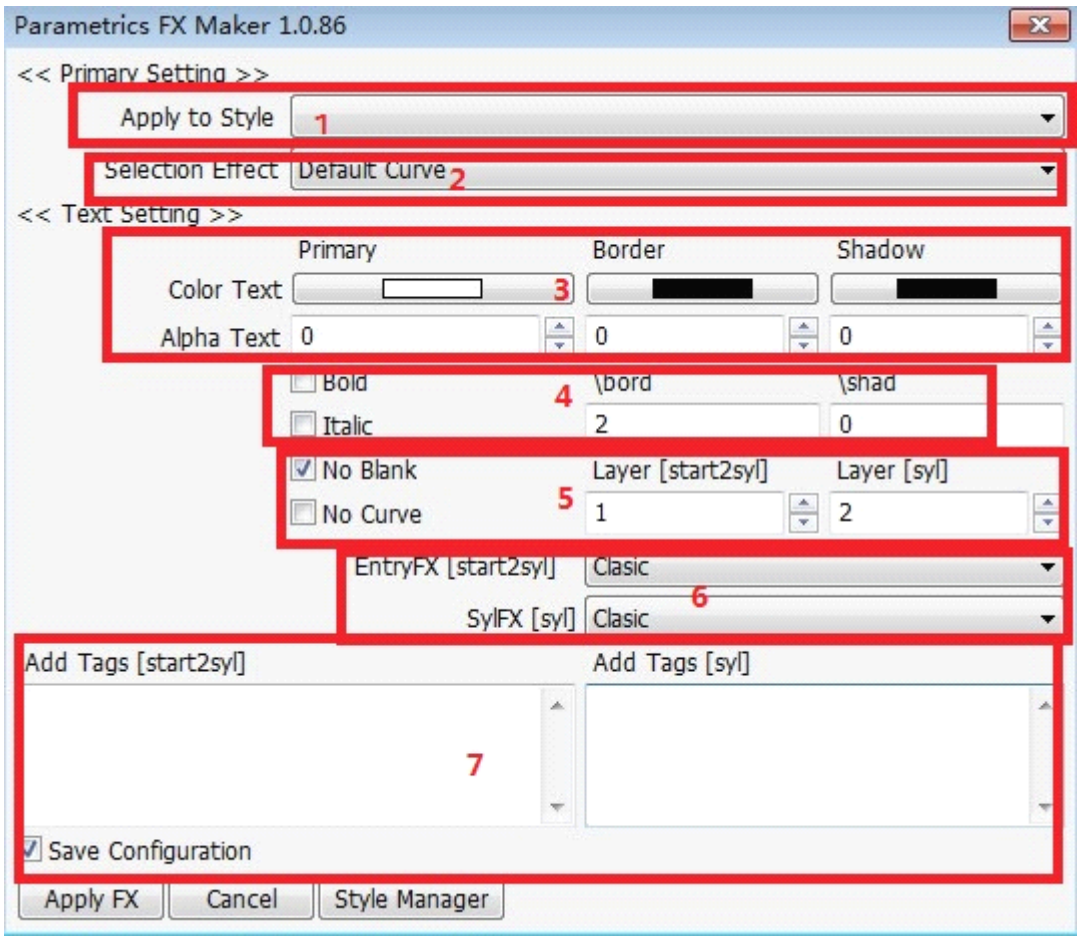
然后，再打开自动化。。。能看到多一个了吧。。。



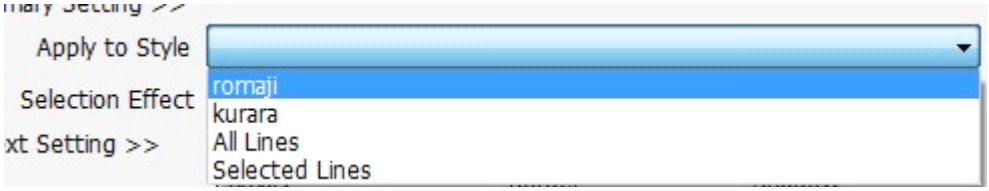

打开是这个界面



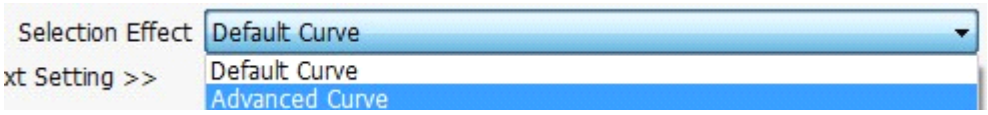

下面我快速讲一下，估计大家都很面熟。。。。都是些基础的东西。。。




1：应用于何种样式




2：选择的特效种类，有两种，一个是默认的，一个是高级的，默认就是纯字效果，高级的会加上图形特效，在这个lua里面统称为曲线（这也很好理解，所有的图形都可以广义的规成曲线。。。）

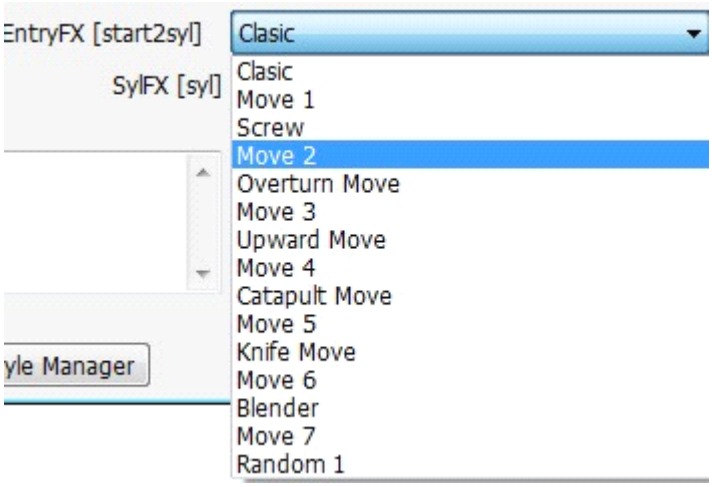


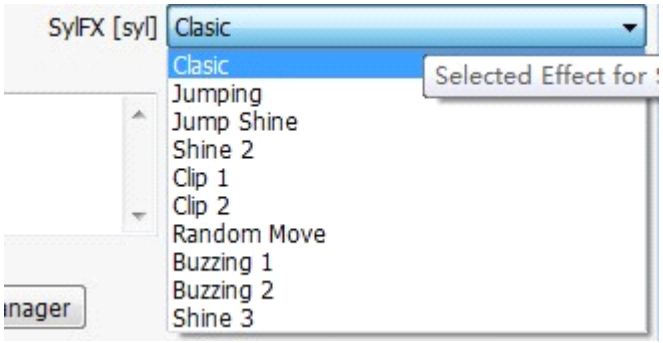

3：字的颜色，边框，阴影及不透明度（这个不用讲了吧）

4：边框粗细，倾斜。。。- -

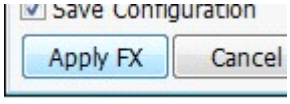
5:no blank 就是是否应用于空格，no curve就是是否还有图形效果，纯字就打勾吧，后面的层是设置前后顺序的，简单点说，就是你的字是在图形上面还是下面，自行设层。。。（实际这块有经验的就知道可以做好多层的）

6：内置了一些入场和高亮的特效，都很常见，不过的确省了不少事，这样就可以专心玩高端的图形了（偷懒去屎~



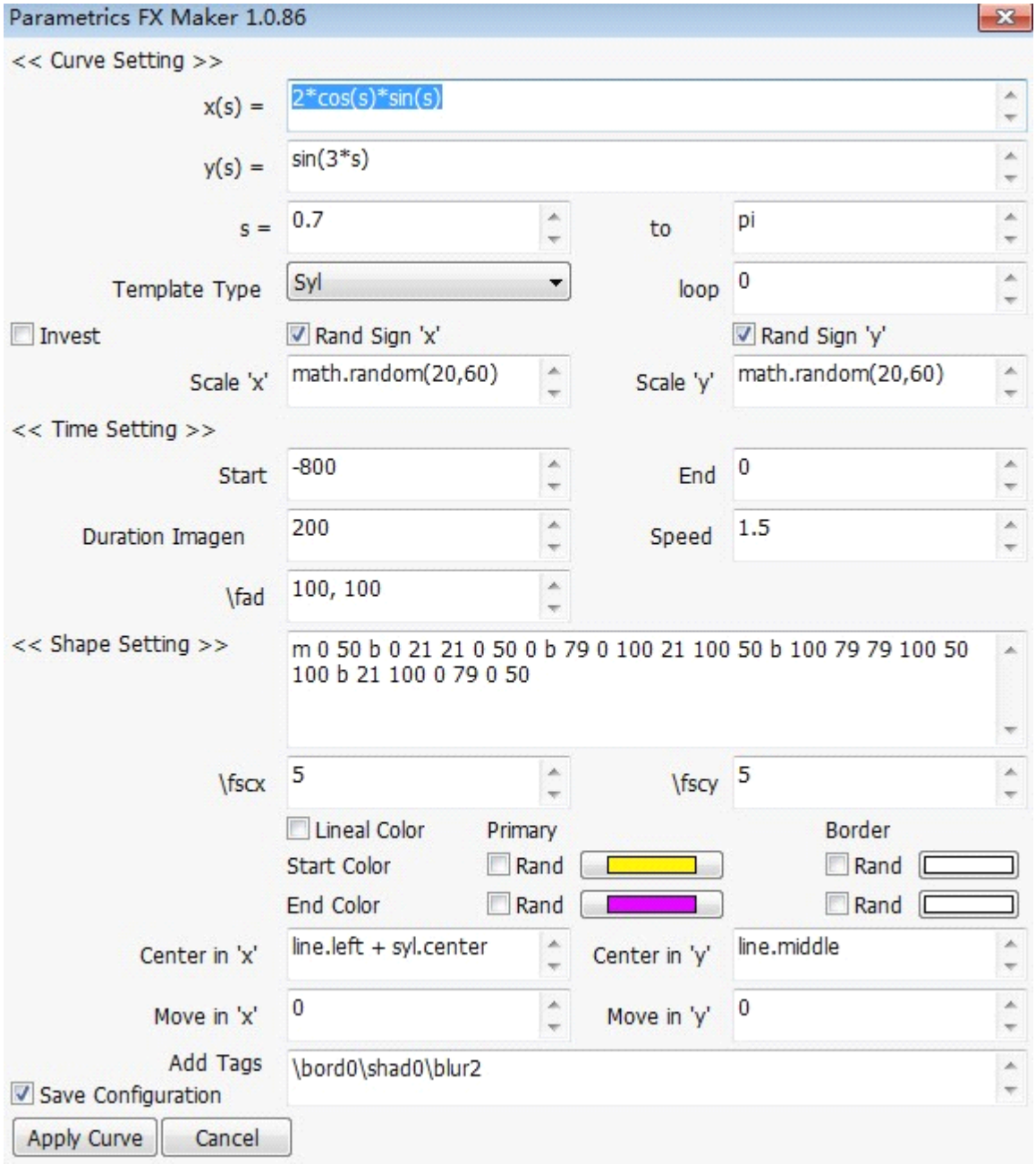




7：当然这里还是要照顾高端玩家的，这里可以写自己的特效，但是这里要填的特效要写出字符串，加上单引号，然后用转义字符，大神都懂的，不解释了。。。（后面我会举例的，表慌~

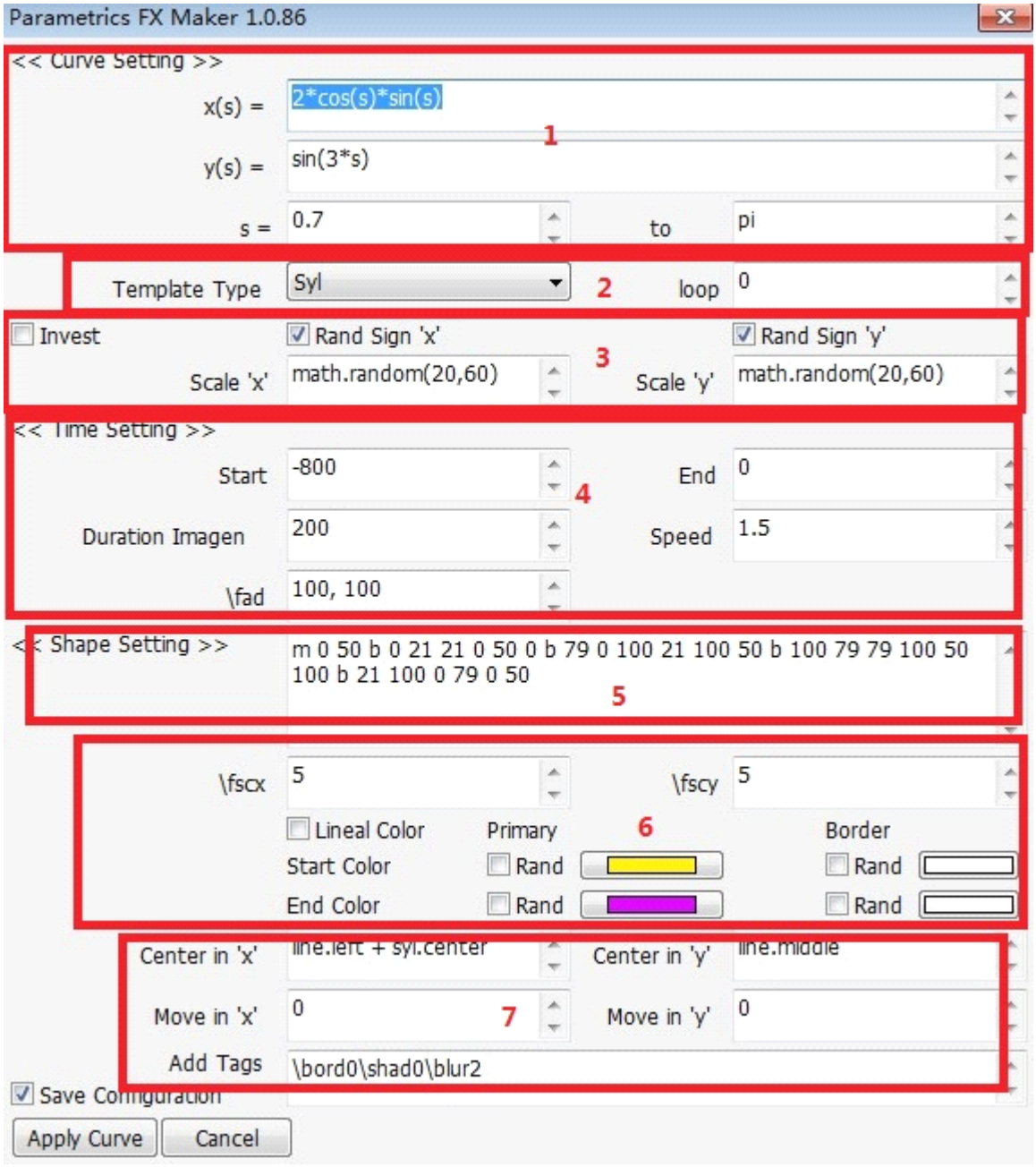
然后，点应用 

，如果是纯字的，就到此为止了，如果是选高级的话，还远远没完。。。（好累。。

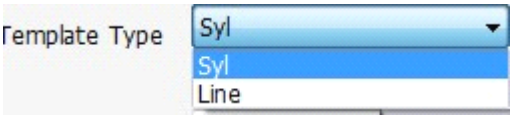
高级会出现这个界面




（哎，又要解释。。。稍微说一下吧。。。




1：为图形的函数，而且是参数方程，高数大家都被折磨过，不解释了（可以设为0，如果根本不需要的话。。。需要方程的原因，做曲线吗，贝塞尔啥的。。。）

2：特效是用于音节还是行 

，loop当然就是循环次数了。。。（我的机器就怕loop和clip，机器渣啊。

3：x，y的运动范围，还可以选择是否反向运动。。。（看到参数范围了吧，这个一般是从小到大的，但是可以反过来，这个我喜欢。。。

4：设置起始时间，持续时间，速度，还有是否淡入淡出。。。（我感觉我好啰嗦。。。

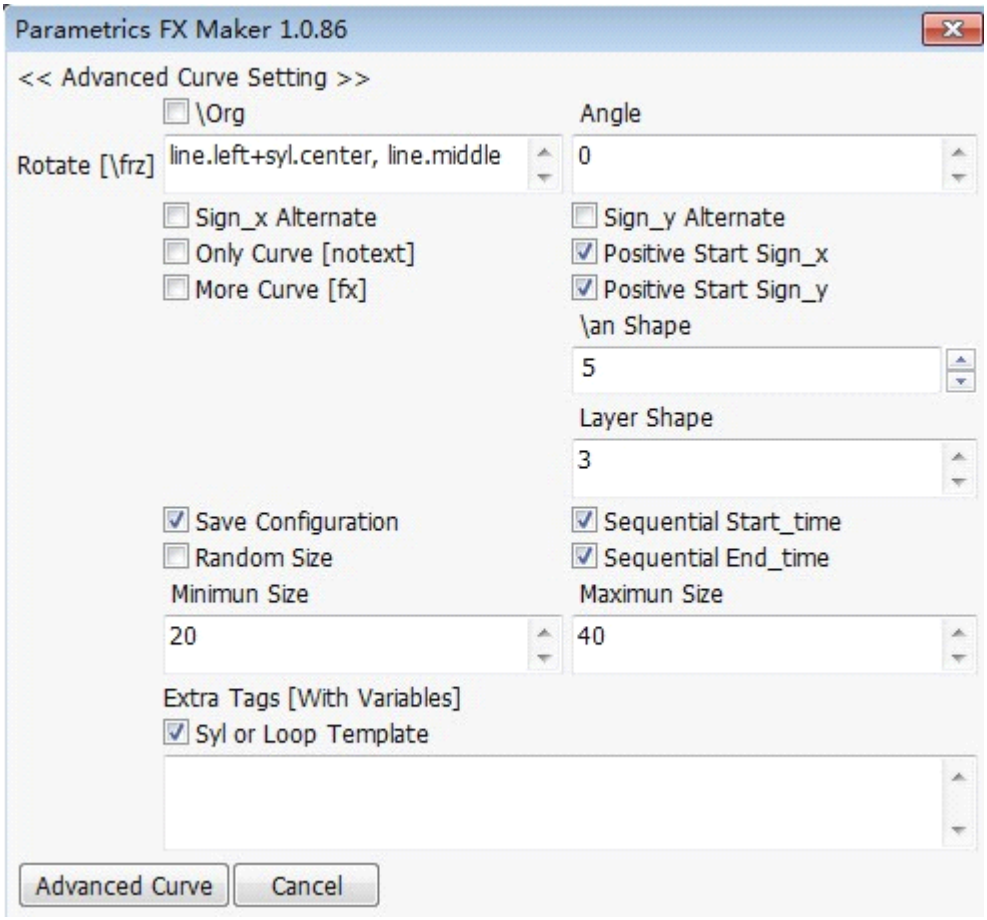
5：图形代码

6：设置图形大小，颜色（有是否线性，你自己看着办吧），还有随机色。。。起始，结束都可以设。。。

7：设置中心位置和运动效果。。。

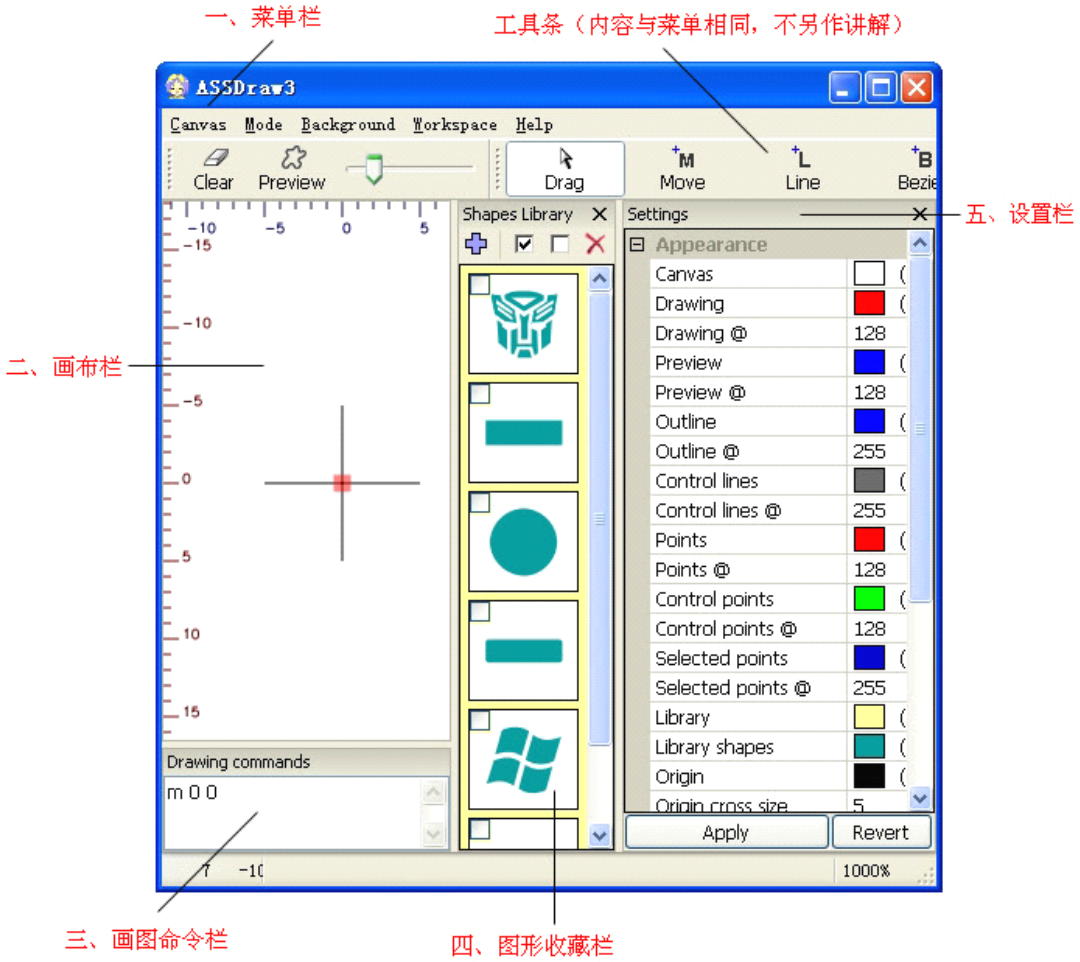
完了就点，应用吧，骚年，你以为完了吧，一般是这样，但是细心的都发现如果设置图形层的位置和旋转点的位置并没有，废话，当然有，不过如果需要才会有。。。好吧，这跟中心点的设置有莫大关系，设了当然还有。。。

如果有的话，骚年，会出现这个界面

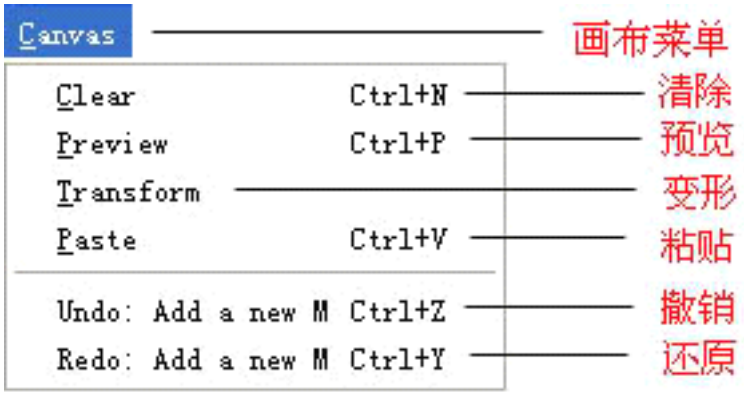



看到这个界面的往往就是比较高端了，这个自己看也明白吧，我就说一下那个， more curve好了，因为这个比较碉堡，说简单点就是可以反复添加新的图形效果。。。只要你机器比较土豪。。。

# 二十四、ASS Draw3 简要教程




## 一、Canvas（画布菜单）各项说明




①. 清除——清除画布上的图案。

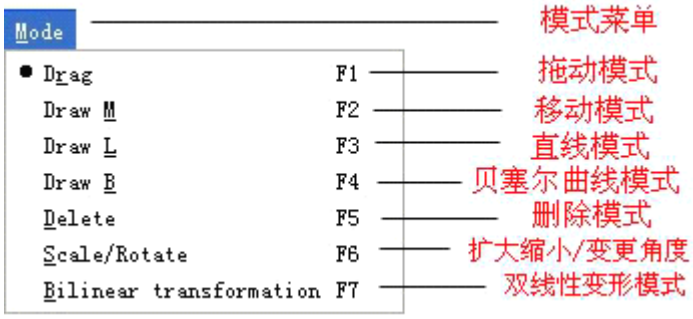
②. 预览——预览已画好的图案。

③. 变形——打开变形模板，在各个参数处输入数值可使图形变形。或直接使用下拉菜单处的模板可以进行一些简单的移动、角度变换、翻转等。各数值参照下面的公式。

④. 粘贴——此处粘贴内容只会显示在下面的命令栏里，而不会粘贴到画布中。

⑤、⑥两项就不用解释了吧。

## 二、Mode（模式菜单）各项说明




①. 拖动模式——可对画布中的已存在的顶点和中心（即十字标）进行拖动。

②. 移动模式——非常让新手混淆的模式。在此模式下，每点击一次鼠标就会产生一个新的顶点。
如果先选中已有的顶点再进行操作，移动的其实是新产生的顶点。

③. 直线模式——顾名思义，画直线使用。

④. 贝塞尔曲线模式——画曲线使用，更多的时候是用在勾勒图案的轮廓上。相当重要的一个模式。
想要画出好的图案，此模式必须熟练掌握。

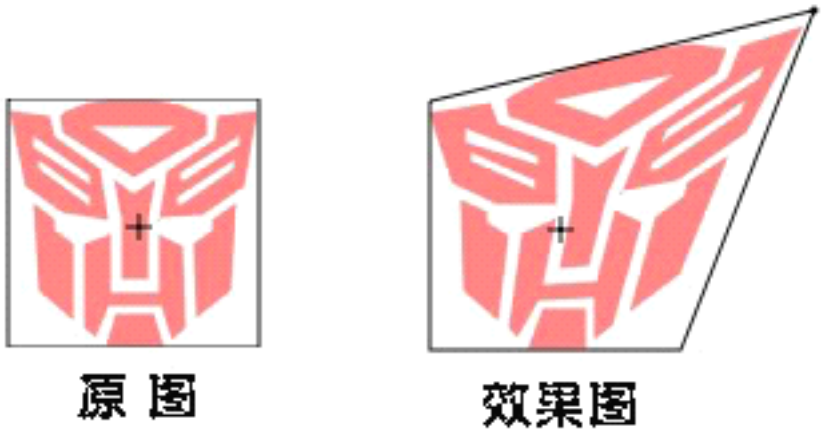
⑤. 删除模式——删除画布中的顶点。与Move正相反，一个是产生新的顶点，一个是删除已有的顶点。
但要注意的是，必须先选中此模式才能删除顶点。
其它模式下先选中顶点再选择此模式或点击删除工具是无效的。

⑥. 扩大缩小/变更角度——在此模式下，所有的图案都会被一个矩形框框起来，此时用鼠标左键拖动四个角的点，可扩大或缩小画布中的图案，而用右键在画布的任何位置都可改变图案的角度（即旋转图案）。

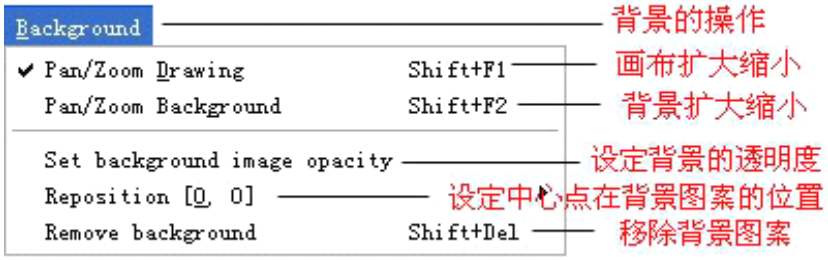
另外还一点需要说明的是，如果画布的缩放是100%（即滑块在最左端），那么画布中图案的大小即为反映到ASS文件中的大小；但一般情况下100%对于画图来说是很不方便的。因此初学者容易犯的另一个错误就是：以为画布图案的放大即为实际图案的放大。其实只有在此模式下放大或缩小的图案才能体现在ASS特效中。

⑦. 双线性变形模式——与第6项类似，选中此项后，图案同样会被矩形框框起来，鼠标左键拖动四个角上的点可进行双线操作。具体效果请图例。（效果图）

双线操作效果图




## 三、Background（背景操作菜单）




注意：把外部图片拖入画布中后，2-5项才会开启

①.画布扩大缩小——此项为默认选中，也只有选中此项后才可对画布进行缩放。

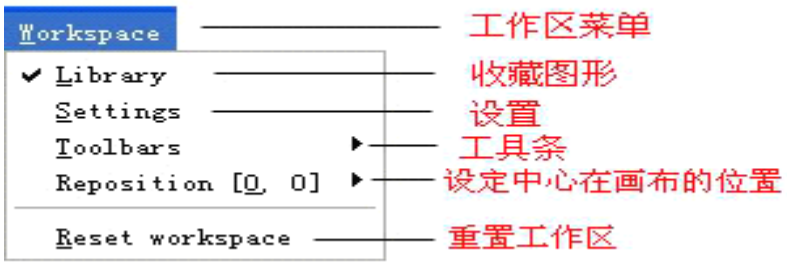
②.背景扩大缩小——扩大或缩小背景图案。勾勒图案的轮廓时经常使用。

③.设定背景的透明度——如果Ask for image opacity没有钩选，图片将会被直接载入。
此时可选择此项来设置或变更背景图案的透明度。

④.设定中心点在背景图案中的位置——可设定5个位置，分别是：左上、右上、中心、左下、右下。
除中心点外，其余4个位置指的是背景图案边框的四个角。

⑤.移除背景图案——注意，此操作不可逆，移除后Ctrl+Z无法恢复，只能重新载入。

## 四、Workspace（工作区）




①.收藏图形——打开图形收藏栏

②.设置——打开设置栏。而设置栏的各项说明会单独做图解。

③.工具条——控制工具条内各个工具的显示状态。菜单从上到下依次为：

Canvas　是否显示画布操作按钮（Clear、preview、滑块）；

Drawing　是否显示绘图操作按钮（Draw至Bilinear）；

Background　是否显示背景操作按钮（Pan drawing、Pan background）；

Show all　显示所有按钮；

Hide all　隐藏所有按钮；

Dock all　让分散的按钮重新回归工具条；

Undock all　分离所有按钮。

④.设定中心在画布的位置——与设定背景图案的位置类似，也是左、右上、中心、左下、右下5个方位，除中心点外，其余4个位置指的是画布的四个角。

### 五、Help（帮助菜单）

①.Manual（帮助手册）——本来这项不必说明的，但为了教程的完整还是加上了。另外，想要更好地了解和掌握ASSDraw，这本最权威的官方手册应该会有很大帮助。

②.About（关于）——软件的介绍及版本说明。

## 画布操作详解

### 一、画布空白处

①鼠标左键——Drag　单击无任何操作，按住左健拖动可圈选顶点。

Move 单击：曾加新的顶点（即画图命令栏多一个m），按住左健拖动则会移动当前创建的顶点。

Line　如果画布上已存在顶点，单击将会与最后一个顶点连成线条。否则会创建一个顶点（仅此一次）。按住左健拖动则会创建一条直线。

Bezier 与Line模式相同。

Delete、Scale/Rotate、Bilinear在空白处没有任何操作。

②鼠标右键——单击无任何操作。按往右键拖动则是移动画布或背景图案（Scale/Rotate模式除外），而移动画布还是移动背景图案取决于Pan drawing、Pan background是否被选中。双击会弹出菜单，分别是：

Move [0,0] here　移动中心到此处
Mode：Drag 切换至拖动模式（相当于点击工具条上的Drag按钮，下同）
Mode：Draw M　切换至移动模式
Mode：Draw L　切换到画直线模式
Mode：Draw B　切换到画曲线模式
Mode：Delete　切换到删除模式。

## 二、选中顶点后

①鼠标左键——Drag　单击、双击均无任何操作，按住左健拖动可移动当前顶点。
Move　操作和空白处相同。
Line　按住左健拖动将会以当前顶点为起点画一条直线。
Bezier　与Line模式类似，按住左健拖动将会以当前顶点为起点画一条曲线。
Delete　删除当前顶点。
Scale/Rotate、Bilinear均无顶点可供操作。

②鼠标右键——单击和拖动与“2.空白处”的操作相同。

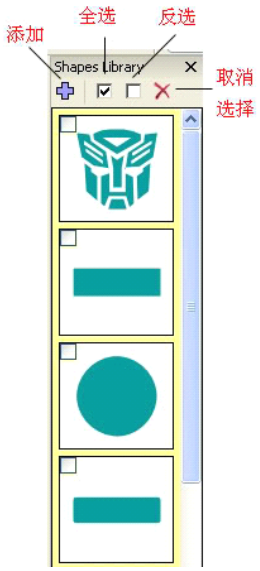
双击：如选择的是原始顶点（即初始状态的顶点和Move模式创建的顶点 m 0,0），弹出的菜单只会显示当前顶点的坐标而无其它操作。如果选择的是画直线或画曲线产生的顶点，弹出的菜单则会是切换顶点的状态（即直线、曲线互换）。分别是：Convert to Bezier curve （B command）切换顶点为贝塞尔模式、Convert to line （L command）切换顶点为直线模式。

此外，如果选择的是顶点是一条曲线的终点，而这个终点又与另一条曲线的起点相连，弹出的菜单中会多出一条：Smooth connection 两个贝塞尔曲线的顶点间光滑连接。至于具体效果是什么还有待研究。

## 三、命令栏

虽说是命令栏，其实可以当作状态栏来看，很多操作的结果都可实时地显示在此栏里。在输入画图代码方面，与Aegisub相比不用输入开始\p1与结束\p0命令。除此之外，具体操作没什么好说的。如果想分拆某个图案时，删除某段以m开头的坐标可以实现批量删除的效果。

## 四、图形收藏栏



## 一、按钮介绍

 ：Save canvas　添加：将画布中的图案添加到收藏栏。

 ：Select all　全选：选中收藏栏所有的图案。

 ：Select none　取消选择：清除选中图案单选框中的 √ 。

 ：Delete selected　删除当前选中的图案。注意：此过程不可逆，请小心操作。

## 二、右键菜单介绍


Load canvas　读取图案：读取当前鼠标所指位置的图案到画布中。此项操作与当前图案是否被选中无关。
在图案上双击左键也有同样效果。

Copy commands to clipboard　复制命令到剪贴板：直接复制该图案的画图代码到剪贴板。要比到命令栏复制更方便。

Save canvas here　保存图案到此处：执行此项操作前必须勾选一个图案框（如果是多选，会以最后一个为准），可以是空白框或已有图案框。如果是空白框则会直接添加到此框内；而如果是已有图案框则会被覆盖。

Delete from library → Confirm delete? 确认删除？说是确认删除，其实是没有确认直接删除的。
具体效果为删除鼠标当前指向的图案。
与读取类似，此项操作也与当前图案是否被选中无关。

## 五、设置栏

这里的设置都是无关紧要的，可以不用管。具体说明请大家看图吧。


外观（此处特指软件的外观属性）
画布（设置画布的颜色）
绘画（设置图形的颜色）
设置所绘制图形颜色的浓度
预览（此处指用何种颜色来预览图像效果）
设置预览图形颜色的浓度
轮廓线（设置轮廓线的颜色）
设置轮廓线的粗细（0-255）
控制线（设置控制线的颜色）
设置控制线的粗细（0-255）
顶点（设置顶点的颜色）
设置顶点颜色的浓度（0-255）
控制点（设置控制点的颜色）
设置控制点颜色的浓度（0-255）
选择点（设置当前选中点的颜色）
设置选择点颜色的浓度（0-255）
收藏（设置收藏栏的背景颜色）
收藏图形（设置已经收藏图形的颜色）
起点（设置原始起点「十字标」的颜色）
设置原始起点「十字标」的大小
设置水平标尺的颜色
设置垂直标尺的颜色
行为面板
大写字母命令（以大写字母M来出输出命令）
询问图像的不透明度
解析 S/P/C（意义不明）
无启动画面
确认退出
Settings
Appearance
Canvas (255,255,2
Drawing (255,0,0)
Drawing @ 128
Preview (0,0,255)
Preview @ 128
Outline (0,0,255)
Outline @ 255
Control lines (102,102,1
Control lines @ 255
Points (255,0,0)
Points @ 128
Control points (0,255,0)
Control points @ 128
Selected points (0,0,204)
Selected points @ 255
Library (255,255,1
Library shapes (1,152,153
Origin (0,0,0)
Origin cross size 5
H ruler (0,0,102)
V ruler (102,0,0)
Behaviors
Capitalize commands
Ask for image opacity
Parse S/P/C
No splash screen
Confirm quit
同意（指选项生效）
恢复
Apply
Revert