

HStream

StreamMaster

## **Configuration manual**

(V2.0)

Revision 6.0

(适合 StreamMaster v2.0.0 以上版本)

## 目 录

前 言 .....	2
第一章 简要说明 .....	2
1 找到正确的配置文件 .....	2
2 配置内容分区 .....	2
3 让配置生效 .....	2
第二章 系统配置文件 .....	3
第 1 节. <General> .....	3
第 2 节. <RAVIODevices> .....	6
第三章 流配置文件 .....	6
第 1 节. <General> .....	6
第 2 节. 关于 StreamMaster Worker .....	8
第 3 节. <MediaSources> .....	12
第 4 节. <VideoDecoders> .....	15
第 5 节. <VideoMidProcessors> .....	17
第 6 节. <VideoEncoders> .....	23
第 7 节. <MediaWriters> .....	25
第 8 节. <OSDs> .....	27
第 9 节. <OSGs> .....	29
附录 .....	34
1. 附录 I OSG 几何图形表 .....	34

## 前 言

StreamMaster（流影大师）（V2.0+）是高盛科技（超）高清音视频处理系列产品的核心模块。它主要依据配置文件的要求工作，因为配置文件条目较多，特提供此文档进行说明。用户根据本文的释义结合实际应用场景进行配置。

## 第一章 简要说明

### 1 找到正确的配置文件

- 配置文件所在路径：StreamMaster 配置文件位于 StreamMaster 主目录，即主程序文件“StreamMaster.exe”同一目录下；如缺失，请重新下载软件包或联系技术支持。
- 文件名：StreamMaster（V2.0+）的配置文件共有 2 个：
  - ✓ “SysConfig.xml”：**系统配置文件**；用于存储 StreamMaster 的系统基础配置（硬件和资源限制等高级配置选项；详见第二章），这些选项通常在 StreamMaster 版本和硬件环境确定后也就固定了。
  - ✓ “StreamCfg.xml”：**流配置文件**；用于定义 StreamMaster 中的 Worker，音视频流、OSD/OSG 等信息；详见第三章）。用户可根据应用上下游音视频资源的实际情况和需求进行配置；
- 文件格式：配置文件内容为 XML，可读字符（半角），编码格式为 Unicode(UTF-8)。

### 2 配置内容分区

配置文件内的条目，按照所控制的功能类型，化分为很多个区段（Section），（按照 XML 文件格式要求）每个区段从“<区段名称>”标记开始，到“</区段名称>”标记结束，例如：

```
<OSDs> (OSD 定义区段开始)
..... (配置项)
</OSDs> (OSD 定义区段结束)
```

### 3 让配置生效

可以在任何时候修改配置项，但是在 StreamMaster 下一次启动前，这些配置是不会生效的。因此，如果您是在 StreamMaster 运行期间修改了配置，需要重新启动 StreamMaster 来让它们生效。

## 第二章 系统配置文件

在系统基础配置文件“SysConfig.xml”中，主要包括以下几个配置区段。

### 第1节. <General>

General 区段中为针对整个 StreamMaster 的系统级配置杂项，具体如下：

- 1) <BindInterface>：绑定接口名称；当 StreamMaster 所在设备（如 UDC）有多个网卡时，使用该配置可以指定 StreamMaster 使用哪个网卡与外界通讯

举例：“Realtek PCIe GbE Family Controller”

- 2) <BindMAC>：绑定接口物理地址；类似“BindInterface”，也可以通过 MAC 地址来指定所使用的网卡。

举例：“40:B0:76:7C:9D:57”

- 3) <GPUSetting>：GPU 配置组；当您只使用一张显卡时无需配置它。
- 4) <EfficiencyCtrl>：流处理性能微调设置项组，通常保持默认配置即可。包括如下配置项：

- (1). <MTRTD>：最大容忍实时误差，单位是毫秒；对于每一路有高度实时性要求的视频源来说，所能容忍的最大实时误差；如果 StreamMaster 接收到某一视频帧时，该帧的时间戳证明它已经滞后了超过了该值，则该帧画面被丢弃。

◆ 有效值：【0, 10000】；

◆ 默认值：1000

- (2). <MTIFD>：最大容忍帧内同步误差，单位是毫秒；

◆ 有效值：【0, 5000】；

◆ 默认值：500

- (3). <AVSynLimit>：音视频同步误差上限，单位是毫秒；

◆ 有效值：【0, 3000】；

◆ 默认值：300

- (4). <RealtimeCtrl>：是否启用实时控制；如果该选项关闭，则 MTRTD 配置项无效。

- ◆ 有效值: "on", "off";
  - ◆ 默认值: "off";
- (5). <SIF>: 是否启用帧内同步; 如果该选项关闭, 则 MTIFD 配置项无效。
- ◆ 有效值: "on", "off";
  - ◆ 默认值: "on";
- (6). <AVSyn>: 是否启用音视频同步控制; 如果该选项关闭, 则 AVSynLimit 配置项无效。
- ◆ 有效值: "on", "off";
  - ◆ 默认值: "on";
- 5) <InitializeMonitors>: 启动后是否初始化显示器, 如果该项为 true, StreamMaster 在启动后会将所有显示器输出画面初始化为黑屏, 并在屏幕中央显示<StandByMsg>定义的文字; 这个选项在 StreamMaster 宿主机的显示输出接口被用于应用显示输出时会比较有用。该项默认值为 false。
- 6) <ShowProgress>: 运行时是否在 Console 显示流处理进度; 注意该选项在 -silent 模式下是无效的。
- 有效值: "on", "off";
  - 默认值: "on";
- 7) <DebugLevel>: Debug log 的等级; 该数字越大, debug log 输出的密度越大;
- 有效值: **【1, 8】**; 测试阶段建议值为 5, 发布版建议值为 3;
  - 默认值: 3
- 8) <DebuglogLife>: Debug log 文件的保存时间, 单位是天, 时间超过该数字的 debuglog 文件将被 StreamMaster 在启动时自动删除。
- 9) <DebuglogSizeLimit>: Debug log 文件的大小限制, 单位是 KB, 超过该大小的 Debug log 文件将被 StreamMaster 在启动时自动删除。
- 10) <DependedService>: StreamMaster 所依赖的服务列表, 服务名称之间用逗号','分隔; 如果配置了该项; 则 StreamMaster 启动时首先检查这些服务是否已经启动;
- 举例: DependedService = "EasyDarwin\_Service"

- 11) <TimeWaitingDpdc>: 依赖服务等待时间, 单位是毫秒; 如果 StreamMaster 启动时发现“<DependedService>”中所依赖的服务尚未启动, 会尝试启动该服务并等待其启动完成, 但如果在经过本项所定义的时间后仍未成功启动, 则视为启动失败。
- 12) <KeyStrmWaitTime>: 关键输入流等待时间, 单位是秒; StreamMaster 在启动时会尝试连接打开每个【StreamCfg.xml】的<MediaSources>中定义的每个输入流(inputStream), 如果某个流被定义为关键输入流却打开失败, StreamMaster 会尝试重新打开它, 但如果经过该时间后仍未成功打开, 则视为启动失败。

## 第2节. <RAVIODevices>

用于配置本机已安装的原始视频输入输出端口信息。

1. 原始 (Raw) 视频输入端口；一般指机载视频采集部件上的接口；
  - <FixedVideoIn>：正整数，原始视频输入端口的数量；该值的最大有效值为 16；
  - <FVIDN-##>：(##的取值为从 1 到 FixedVideoIn，为端口序号)；此配置项指明了该视频输入端口的别名。

举例：<FVIDN-1> "Live Streaming Video Device"</FVIDN-1>

用户可以使用 ffmpeg 命令行工具列出本机已安装的视频采集设备清单，具体格式如下：

```
"ffmpeg -list_devices true -f dshow -i dummy"
```

```
"ffplay -f dshow video="HDPro 1""
```

2. 原始 (Raw) 视频输出端口；一般指机载显示输出部件上的接口；
  - 该组配置目前为自动检测，暂未启用；

## 第三章 流配置文件

### 第1节. <General>

General 区段中为针对 StreamMaster 用户级配置和处理流程定义的普通级配置杂项，具体如下：

- 1) <Location>：安装地点；通常用来区分本 StreamMaster 实例负责处理的场合名称；
  - 举例：<Location> "南京高盛信息科技有限公司二楼会议室" </Location>
- 2) <DefaultBGC>：默认背景色：指如果某一路视频输入没有数据时，对应的显示输出窗口应显示怎样的背景色；名称包括：
  - 有效颜色："White", "Yellow", "Red", "Green", "Blue", "Black"；
  - 默认颜色："Black"
- 3) <ErrMsgColor>：故障信息颜色：指如果某一路视频输入出现严重故障时，对应的

显示输出窗口显示的故障信息文字颜色；

- 有效颜色: "White", "Yellow", "Red", "Green", "Blue";
- 默认颜色: "Red"

4) <InfoMsgColor>: 普通信息颜色: 指如果某一路视频输入存在一般异常时, 对应的显示输出窗口显示的提示信息文字颜色;

- 有效颜色: "White", "Yellow", "Red", "Green", "Blue";
- 默认颜色: "White"

5) <MsgHeight>: 故障或提示信息文字的高度;

- 有效值: 【16, 128】, 8 的整数倍;
- 默认值: 32

6) <StandByMsg>: 待机输出文字; 即当设置了视频输出时, StreamMaster 启动过程中显示的文字内容;

- 如果该路输出并未指定输出画面, 则运行过程中也显示该内容。
- 该文字的最大长度为 64 个字符。
- 举例: <StandByMsg> "-南京高盛信息科技有限公司研发部-"</StandByMsg>

7) <RecordWriter>: 录像所对应的 "MediaWriters "的别名;

- 举例: <RecordWriter>"TrumanShow\_Record"</RecordWriter>

8) <PreviewWriter>: 预览画面流 (一般为 RTSP) 所对应的 "MediaWriter" 的别名;

- 举例: <PreviewWriter>"TrumanShow\_Live\_HD "</PreviewWriter>

9) <LiveBCWriter>: 实况直播流 (一般为 RTMP) 所对应的 "MediaWriter" 的别名;

- 举例: <LiveBCWriter> "TrumanShow\_Live\_4K "</LiveBCWriter>

10) <GlobalFramerate> : 用于设置 StreamMaster 统一使用的视频帧率, 单位是 fps; 如果设置了该选项, 则 StreamMaster 在内部处理和对外输出视频时, 帧率一律统一到该数值, 从外界接收采集的画面也会先调整到该数值后再做进一步处理; 但如果某个 Worker 用 <OutputFramerate>定义了自有帧率, 则对它来说, 自有帧率的优先级要高于统一帧率。



## 第2节. 关于 StreamMaster Worker

从 `<MediaSources>` 到 `<MediaWriters>` 定义了 在 StreamMaster 中 实际使用的 StreamMaster Workers。每个 Worker 根据其设计功能完成音视频流的一个处理环节：或者负责从源（文件、输入设备或者网络流）读取音视频流（MediaSources）、或者负责将一个或多个音视频流进行加工处理或显示输出（Decoders、MidProcessors、Encoders），或者负责将处理好的流推送到网络或保存成文件（MediaWriters），从而协同组成 StreamMaster 对于流媒体的处理链条。

### 1. StreamMaster Worker 的定义：

每个 Worker 的实例都必须有一个别名，仅支持字母数字和下划线，最大长度为 31 个字符。因为用户针对输入源（MediaSources）的控制操作如焦点画面切换等是使用别名指定它们的，因此必须为每一路输入源 Worker 定义有效的别名（*WorkerAlias*），并注意所有输入源的别名是不可以重复的。

每个 Workers 主区段都会划分为多个 Worker 子区段（主区段仅有一个配置项 `<XXXCount>`，（XXX 在不同类型的 Workers 区段有别）它代表此类 Worker 的总数，其最大有效值范围是 **【0, 128】**）；每个子区段负责定义一个具体的 Worker，其区段格式如下：

```
<WorkerAlias id="##">
<DisplayName>"XXXXXX" </DisplayName>
.....
<PushAtBegin>true/false</PushAtBegin>
</WorkerAlias>
```

其中：*WorkerAlias* 是该 Worker 的别名；## 是 Worker 的序号。注意这个序号的有效值范围是 **【1, <XXXCount>】**；

`<DisplayName>`：该 Worker 的显示名称；主要是便于用户理解该 Worker 的意义；

`<PushAtBegin>`：是否在启动并与上游成功连接后即开始向下游推送数据。

举例：

```
"<JudgeCamera id = "03">
<DisplayName>" 法官摄像机" </DisplayName>
```

.....

*<PushAtBegin>true </PushAtBegin>*

*</JudgeCamera>"*

定义了一个别名为 "JudgeCamera", 序号为 3 的纯视频输入源 Worker。

## 2. StreamMaster Worker 的引用:

在 StreamMaster 任何配置文件里经常需要引用一个 Worker 或其某个输入输出流; 此时需要使用该 **Worker 全局名称(WorkerGlobalName)**或**流全局名称(StreamGlobalName)**;

**Worker 全局名称 (WorkerGlobalName)** 的格式如下:

*WorkerFroupName . WokerAlias*

**流全局名称 (StreamGlobalName)** 的格式如下:

*WorkerGlobalName.. [StreamType] . [identifyWay. identifier]*

上述格式中:

- WorkerFroupName: 是该 Worker 所属的群组名称 (也是 Worker 类型); 它是必须的字段; 其有效值包括:
  - "MediaSources"
  - "VideoDecoders"
  - "VideoMidProcessors"
  - "VideoEncoders"
  - "MediaWriters"
- WokerAlias: 是该 Worker 的别名; 它是必须的字段。
- [StreamType] : 是具体引用的该 Worker 输入输出流的类型; 它是必须的字段; 其有效值包括:

- "VideoOutput": 视频输出流;
- "VideoInput": 视频输入流;
- "AudioOutput": 音频输出流;
- "AudioInput": 音频输入流;
- "ToDownstream": 流往下游前;
- "ToRendering": 在本地显示播放前;
- *[identifyWay]*: 细分方式; 当 Worker 有多个输入/输出流时, 需要用细分方式指定所引用的是其中哪一个流; 其有效值包括:
  - "ByName": 使用流别名指定; 使用此方式的前提是定义了流别名。此时 *identifier* 应该是该流的别名 (*StreamAlias*);
  - "ById": 使用流编号指定; 此时 *identifier* 应该是该流在 Worker 内部的流编号 (*StreamSmallId*) 而不是全局流 ID。

举例:

例 1: "*MediaSources.VideoBooth* " : 引用了别名为 VideoBooth 的多媒体输入源 Worker.

例 2: "*MediaSources.EDU.VideoOutput.ByName.HD1720P*": 引用了别名为 EDU 的多媒体输入源 Worker 的别名为"HD1720P"的视频输出流。

例 3: "*VideoMidProcessors.VideoMatrix.VideoOutput.ById.07*": 引用了别名为"VideoMatrix"的 VMP 的 07 号视频输出流。

例 4: "*VideoMidProcessors.VideoMuxer4K.ToRendering*:" 引用了别名为"VideoMuxer4K"的 VMP 的视频输出的用于现场显示输出的分支流。

### 3. 视频直接显示输出:

对于每个输出单一裸视频流的 Worker 来说, 它们都有一个**【视频直接显示输出配置项组】**, 该组包括如下配置项:

- (1). <DirectOutput>: 此配置项指明了该视频输出是否要在本机显示输出;

- 有效值: "true", "false";

- 默认值: " false";

(2). <outputOn>显示输出的目标, 用来定义画面显示的窗口, 其取值决定了本组剩余配置项的有效性, 它包括以下有效值:

- <FullScreen>: 全屏显示; 此时必须用<OutputDeviceNo>来定义所用显示器的编号;

- <NewWindow>: 新建窗口显示, 即由 StreamMaster 新建一个专用窗口来显示视频; 此时必须用<WindowWidth> 和<WindowHeight>定义新窗口的大小。新窗口的标题优先使用 Worker 的<DisplayName>, 其实是 WorkerName。

### 第3节. <MediaSources>

MediaSources 区段通过定义 MediaReader 这种 StreamMaster Worker 的方式，配置了 StreamMaster 的输入音视频源的信息；在这里<XXXCount>实际为<MScount>，它代表输入多媒体源的总数，有效值范围是【0，128】。

每个输入源定义子区段可能包括如下配置项：

- 1) <MSTYPE>：输入源的类型；
  - 有效值为“RTSP”、“RTMP”、“FILE”、“VCAP”；
  - 默认值：无；该项必须配置为正确值。
- 2) <MSFullPath>：输入源的完整路径；
  - 对于 VSTYPE 为“RTSP”、“RTMP”的源，它是指完整的 URL 链接；
  - 对于 VSTYPE 为“FILE”的源，它是指全路径文件名；
  - 对于 VSTYPE 为“VCAP”的源，该项无效。
- 3) <Authorization>：连接是否需要认证；目前仅对 VSTYPE 为“RTSP”、“RTMP”的源适用；有些网络音视频源的接收需要客户端进行身份认证才能成功；此时该项应该为 true，此时必须为 AuthUN 和 AuthPwd 两项提供正确的用户名和密码值。
  - 有效值：“true”，“false”；
  - 默认值：“false”；
  - <AuthUN>，<AuthPwd>：上述认证（如果需要）用的有效用户名和密码。两者的最大有效长度均为 31 字符。
- 4) <UseUTC>：输入源是否使用 UTC 时间，注意只有属于网络流媒体类型的音视频源才可能支持 NTP，否则此项应为 false；注意支持该音视频还需要跟 UDC 保持精确校时才能发挥作用。
- 5) <UseNTP>：输入源是否使用 NTP 校时，该值为 false 则意味着其使用 localtime。
  - 有效值：“true”，“false”；
  - 默认值：“true”；
- 6) <UseTCP>：是否使用 TCP 连接；主要针对 RTSP 源；基于稳定性考虑，一般默认

情况下 StreamMaster 使用 TCP 连接视频源，如果该项为 false，则使用 UDP。

◆ 有效值: "true", "false";

■ 默认值: "true";

7) <CapDvcNo>: 仅当 VSTYPE 的为 **"VCAP"**时有效, 表示此路输入源为本地视频 (或音频) 源通过视频线 (SDI/HDMI/DP) 直接接入本机的视频输入端口; 注意这个编号所对应的输入设备必须已经在系统配置文件"SysConfig.xml"的<RAVIODevices>区段中明确定义。

8) <STRMCOUNT>: 计划从输入源读取的音视频流的数量, 注意该数字不可能超过该输入源实际所包含的流数量。

9) **输入流配置项组:**

每个音视频流的配置项组由<Stream id="#">……</Stream>所界定; 其中#为该流的小号, 其有效范围是 **【0, STRMCOUNT】**; 组中可能包含下列流配置项:

(1). <avStrmId>: 所要读取的流在链接或文件中的流 ID;

◆ 有效值为非负整数; 默认值为-1;

◆ 如果不确定该值, 且该源中只有一路视频流, 可将其设为-1, 此时 StreamMaster 会自动搜索该源中的有效视频流;

◆ 如果该源中复合有多个视频流, 则应该明确指定流 ID。

(2). <StrmAlias>: 该流的别名; 规则同 Worker 别名; 流别名不是必须的, 但在同一个输入源包含多个同类流 (比如多个视频流) 的场景下有利于用户区分哪个是自己想要使用或控制的流。

(3). <StrmType>: 流的类型; 必须指定, 有效值包括 **Video** 和 **Audio**。

(4). <CruailStream>: 是否系统关键输入流。关于“关键输入流”的定义, 请参考 第二章-第 1 节-第 12 条-<KeyStrmWaitTime> 的说明。

(5). <PiADTC>: 是否参与同步估算; 只有延时小 (对于 IPCam 还需要 NTP 时钟准确) 的视频源才应该参与估算; 其它视频源建议不要参与 ADTC。

◆ 有效值: "true", "false";

◆ 默认值: " false";

- (6). <EstmtDly>: 预估延时, 该视频源估算的延时时间, 单位是毫秒; 如果一个视频源不支持精确可靠的 NTP, 它的延时也就无法通过自动计算获知; 这种情况下, 为了后续同步的需要, 应该根据现场实测情况, 给出该路视频的估算延时。

◆ 有效值: **[-1, 10000]**;

◆ 默认值: -1

- (7). <ProactiveCache>: 接收缓冲长度, 单位是毫秒; 对于质量不好的流 (如远程或者输出不稳定的编码器的视频流), 建议为其设置一个合适的缓冲值以保证其解码播放的流畅度;

◆ 有效值: **[0, 60000]**;

◆ 默认值: 0

- (8). <PktInBlock>: 音视频数据包是否呈块状传输。对于一些流畅度不好的流 (例如某些版本 ffmpeg 采集推送的音频流), 其推出来的流可能是脉冲式的; 这种情况下, 为了在流畅度与实时性之间获得平衡, 需要预知其分块规律, 即脉宽;

◆ 有效值: "true", "false";

◆ 默认值: "false";

提示: 建议将这种流的 UseNTP 设置为 false;

- (9). 仅在此配置为"true"时, 下列配置项才有效:

◆ PktItvMax: 块内帧间隔上限; 单位是毫秒;

◆ PktGrpItvMax: 块与块间隔上限; 单位是毫秒;

◆ BlocksSkipAtBegin: 评估阶段跳过的块数量。

举例: 假设一路音频源的推送规律为每组连续推送 500ms 长度的数据帧, 每组内数据包间隔小于 20ms, 间隔期也为 500 毫秒; 则可将对应的配置写为:

*PktItvMax = 50*

*PktGrpItvMax = 800*

*BlocksSkipAtBegin = 4*

以下 3 个配置项仅当<StrmType>=Video, 且 VSTYPE 为 "VCAP"时有效:

(10). < OutputFramerate>: 指定的图像采集帧率; 单位是帧/秒;

◆ 有效值: 【10, 120】;

◆ 默认值: 0

(11). FrameWidth, FrameHeight: 图像采集所使用的画面尺寸, 注意图像尺寸必须是该采集部件支持的分辨率;

(12). 视频直接显示输出配置项组, 请参考关于 StreamMaster Worker 中的相关说明。

## 第4节. <VideoDecoders>

VideoDecoders 区段用于配置 StreamMaster 的视频解码器的信息; 在这里<XXXCount>实际为<VDCDCount>, 代表视频解码器的数量; 该值的实际最大有效值取决于版本和硬件; 一般来说越高的版本和越强的 GPU 总能力可支持的解码器数量就越多。

每个视频解码器定义子区段可能包括如下配置项:

- <VideoSource>: 此配置项指明了该解码器的视频源流全局名称, 它通常是某个 MediaReader 的一个视频输出。
- <DecoderType>: 此配置项指明了该解码器所使用的解码器类型;
  - ✓ 有效值: 当前版本只支持取值“NVGPU”或“FFMPG”。
- <OutputFramerate>: 输出视频帧率; 如果有效设置了该项, 则 Worker 会按照该帧率输出视频画面, 如果输入帧率与之不同, 则 Worker 会在输出前对帧率进行自动调整。
- 视频直接显示输出配置项组:

举例:

```
<VdcdJudgeVideo id="03">
  <DisplayName>" 法官画面" </DisplayName>
  <VideoSource>"MediaSources.JudgeCamera.VideoOutput"</VideoSource>
  <DecoderType> NVGPU </DecoderType>
```



```
<OutputFramerate>30</OutputFramerate>  
  
<DirectOutput>true</DirectOutput>  
  
<outputOn>NewWindow</outputOn>  
  
<WindowWidth>1280</WindowWidth>  
  
<WindowHeight>720</WindowHeight>  
  
</VdcdJudgeVideo>
```

上述子区段定义了一个别名为“*VdcdJudgeVideo*”的视频解码 Worker，它使用别名为“*JudgeCamera*”的 MediaSources 的默认视频输出流作为输入源，输出帧率锁定为 30fps；解码后的输出画面会用一个 1280×720 的新窗口显示输出。

## 第5节. <VideoMidProcessors>

<VideoMidProcessors>区段用于配置视频中间处理（VMP）Workers 的信息；在这里 <XXXCount>实际为<MProcCount>：代表 VMP Workers 数量。

VMP 之间是可以分级串联的，即一个 VMP（上游）的输出可以作为另一个 VMP（下游）的输入，但是不允许出现逆流。

每个 VMP 定义子区段可能包括如下配置项：

- 1) MPType：视频中间处理模块的类型；当前版本支持类型包括：
  - ◆ “VMux”：画面拼接器；其主要功能是将多个输入画面拼接成一个大画面；支持多输入单输出；
  - ◆ “VSplt”：画面分割器；其主要功能是将一个输入画面分割为多个小画面；支持单输入多输出；
  - ◆ “VMtx”：视频矩阵；其主要功能是实现多输入画面与多输出画面的动态切换；支持多输入多输出；
  - ◆ “VUnify”：其主要功能是将（可动态变化的）输入稳定为指定画质的输出。支持单输入单输出；
- 2) <VideoSource>：此配置项仅对单输入 VMP 有效；它指明了 VMP 的视频源名称，这个视频源通常是某个 VideoDecoders 或者 Video capture 的视频输出，也可能是其它上游 VMP 的一个输出。
- 3) <RVSIInCount>：此配置项仅对多输入 VMP 有效；指输入的 Raw 视频源的数量；
- 4) <IRVS id="##">：（##的取值范围【1, RVSIInCount】，为视频源在本模块的序号）；此配置项仅对多输入 VMP 有效，用于指明该输入视频源的全域名称，
- 5) <RVOutCount>：此配置项仅对多画面输出 VMP 有效；指本模块输出画面数量；
- 6) <VODftVI id="##">：（##的取值范围【1, RVOutCount】，为视频输出在本模块的序号；）此配置项仅对 VMtx 有效；配置项的值是指该路输出默认使用的输入编号；注意该输入编号必须已经在<IRVS id="##">组中定义。
- 7) <Output id="##">配置项组：（##的取值范围【1, RVOutCount】，为视频输出在

本模块的序号;) 此配置项组仅对 VSplt 有效; 包括如下配置:

- (1). <Alias>: 输出子画面别名;
- (2). <RegionLeft>: 子画面 (在源画面中的) 的左边界;
- (3). <RegionTop>: 子画面 (在源画面中的) 的上边界;
- (4). <RegionWidth>, <RegionHeight>: 子画面的宽度和高度;

以下配置项仅对单一输出 VMP 有效:

- 8) <OutputWidth>, <OutputHeight>: 输出画面宽度与高度;
- 9) <OutputFramerate>: 输出视频帧率; 如果有效设置了该项, 则 Worker 会按照该帧率输出视频画面, 如果输入帧率与之不同, 则 Worker 会在输出前对帧率进行自动调整。
- 10) 视频直接显示输出配置项组, 请参考关于 StreamMaster Worker 中的相关说明。

以下配置项仅对“VMux”有效:

- 11) <IRVSSIFRequired>: 要求参与帧内同步的输入源列表, 列表中的数字必须是<IRVS id="##">中已经定义的“##”, 多个数字之间用半角逗号','分隔。

注:“参与帧内同步”, 是指 VMux 是每生成一帧输出画面前, 需要根据时间戳在该路视频流中找到能跟其它输入对齐的画面帧然后才会完成对应的拼接输出。

- 12) <MuxLayout>: 输出画面拼接合成模式:

- ◆ 有效值: (*RegionCount*: 字母画面总数);
  - ✓ “PIP-1-On-1”: 1+1 画中画; *RegionCount*=2;
  - ✓ “PIP-2-On-1”: 2+1 画中画; *RegionCount*=3;
  - ✓ “4Divide-Equ”: 2 行 2 列等分 4 画面; *RegionCount*=4;
  - ✓ “PIP-4-On-1”: 4+1 画中画; *RegionCount*=5;
  - ✓ “1F-With-5S”: 1+5 画面; *RegionCount*=6;
  - ✓ “1F-With-7S”: 1+7 画面; *RegionCount*=8;
  - ✓ “9-Divided-Equ”: 等分 9 画面; *RegionCount*=9;

- ✓ “1Rows-with-4Cols”: 单行水平 4 画面; *RegionCount*=4;
- ✓ “2Rows-with-3Cols”: 2 行 3 列 6 画面; *RegionCount*=6;
- ✓ “2Rows-with-4Cols”: 2 行 4 列 8 画面; *RegionCount*=8;
- ✓ “3Rows-with-4Cols”: 3 行 4 列 12 画面; *RegionCount*=12;
- ✓ “2Row-with-6Cols”: 2 行 6 列 12 画面; *RegionCount*=12;
- ✓ “16-Divided-Equ”: 4 行 4 列 16 画面; *RegionCount*=16;
- ✓ “2Rows-with-8Cols”: 2 行 8 列 16 画面; *RegionCount*=16;
- ✓ “3Row-with-8Cols”: 3 行 8 列 24 画面; *RegionCount*=24;
- ✓ “4Row-with-6Cols”: 4 行 6 列 24 画面; *RegionCount*=24;
- ✓ “2Row-with-12Cols”: 2 行 8 列 12 画面; *RegionCount*=24;
- ✓ “4Row-with-8Cols”: 4 行 8 列 32 画面; *RegionCount*=32;

**注意: RVSIInCount 必须  $\geq$  RegionCount**

- ◆ 默认值: 即**该值必须正确配置**。
- ◆ 常用画面布局及其编号规则如下图【3.4.1】所示:



【图 3.4.1】

13) <RgnVs id="##">: (##的取值范围【1, *RegionCount*】, 合成画面中的每个区域的序号); 此配置项指明了该区域使用的视频源序号, 即<IRVS id="XX">中的 XX 值;

- ◆ 有效值: 【1, *RVSInCount*】
- ◆ 默认值: 无, 该值不得重复;
- ◆ 关于区域序号与其在合成画面中的位置对应关系如下:

- ✓ 对于非画中画模式，1 指左上角的子画面，即示意图中的“子画面 1”，然后按顺时针递增；但例外是 9 画面模式，其排列顺序如图。
- ✓ 对于画中画模式，1 指母画面；2~5 指叠加于母画面上的子画面，此时子的区域号为子窗口定义的序号，即“<ChildWindow id="X ">”中 X 的取值。

**14) <ChildWindow id="#">配置项组：**（##的取值范围【1, RegionCount-1】，为视频输出在本模块的序号；）此配置项组仅对 PIP（画中画）类的合成模式有效；包括如下配置项：

- (1). <onLayer>：子画面所在的图层；
- (2). <posBase>：子画面的定位基准点：有效值包括：
  - ✓ "Center" 或者 "ScreenCenter": 屏幕/画面正中间
  - ✓ "LeftTopCorner" 或者 "LTCorner": 屏幕/画面左上角
  - ✓ "RightTopCorner" 或者 "RTCornet": 屏幕/画面右上角
  - ✓ "RightBottomCorner" 或者 "RBCorner": 屏幕/画面右下角
  - ✓ "LeftBottomCorner" 或者 "LBCorner": 屏幕/画面左下角
  - ✓ "TopMiddle": 屏幕/画面顶部正中
  - ✓ "BottomMiddle": 屏幕/画面底部正中
  - ✓ "Specified": 指定位置
- (3). <xShift>：相对于定位基准点的水平偏移量（像素）；>0 为右偏，<0 为左偏；  
**必须为 16 的整数倍。**
- (4). <yShift>：相对于定位基准点的垂直偏移量（像素）；>0 为下偏，<0 为上偏；  
**必须为 4 的整数倍。**
- (5). <Width>：子画面的宽度；建议取值区间为【160~输出画面宽度/2】，且必须为 **16 的整数倍。**
- (6). <Height>：子画面的高度；建议取值区间为【90~s 输出画面宽度/2】，且必须为 **4 的整数倍。**

**15) IPGNum：**并行运算组的数量；理论上运算组数量越多图像处理越快，但又受到硬件算力的约束；

- ◆ 默认值： 3
- ◆ 当下建议值是：
  - ✓ UDC51X 系列： IPGNum=3
  - ✓ UDC52X 系列： IPGNum=6
  - ✓ UDC53X 系列： IPGNum=9

16) CylinderHeight: 功能跟 IPGNum 类似, 建议值为:

- ◆ 当 IPGNum=3 时, CylinderHeight = 12;
- ◆ 当 IPGNum=6 时, CylinderHeight = 24;
- ◆ 当 IPGNum=9 时, CylinderHeight = 36;
- ◆ 默认值: 6

## 第6节. <VideoEncoders>

<VideoEncoders>区段用于配置 StreamMaster 的视频编码器的信息; 在这里<XXXCount>实际为

VECDCount: 它代表编码器的数量; 该配置项的最大有效值主要取决 GPU 的硬件编码能力, 一般来说性能越高的 GPU 能支持的编码器数量越多。

每个视频编码器定义子区段可能包括如下配置项:

- 1) <VideoSource>: 此配置项指明了该编码器的视频源名称, 这个视频源必须是一个未经编码压缩的裸视频流 (Raw Video)。
- 2) <Width>, <Height>: 编码后的画面宽度和高度;
  - ◆ Width 必须是 16 的整数倍, Height 必须是 8 的整数倍;
  - ◆ 编码画质一般选择源画质, 但也可以低于源画质; 比如编码前裸视频的尺寸视 4K, 编码画质可以是 1080P 或 720P; 但是不应该高于源画质。
  - ◆ 如果这两项其一为 0, 则表示采用输入原视频的尺寸。
- 3) <OutputFramerate>: 编码后的视频帧率; 如果有效设置了该项, 则 Worker 会按照该帧率输出视频画面, 如果输入帧率与之不同, 则 Worker 会在编码前先对帧率进行自动调整。
- 4) <BitRate>: 编码的码流限制: 单位是 kbps
- 5) <GopLength>: 编码视频的 GOP 长度, 即每个多少帧出现一个关键帧;
  - ◆ 有效值: 【1, 300】
  - ◆ 默认值: 90
- 6) <EncoderType>: 所使用的编码器类型;
  - 有效值:
    - ◆ “NVGPU”: 使用 Nvidia 硬件编码器;
    - ◆ “FFMPG”: 使用软件编码器 ffmpeg 内置的软件编码器;
  - 默认值: “NVGPU”



7) <HwAcclrt>: 硬件加速, 仅当<EncoderType>= “FFMPG”时有效。

■ 有效值:

- ◆ “NVCODEC”: 使用 Nvidia 的 GPU 进行硬件加速
- ◆ “LIBMFX” 使用 Intel 的核心显卡进行硬件加速
- ◆ “AMDAMF”.: 使用 AMD 的核心显卡进行硬件加速

■ 默认值: 不启用 (None)

8) <Vcodec>: 编码格式;

- ◆ 有效值: H.264, AVC, H.265, HEVC
- ◆ 默认值: H.265

9) <Profile>: 编码等级

- ◆ 对于 H264 编码格式来说, 有效值包括: baseline, main, high
- ◆ 对于 HEVC 编码格式来说, 有效值包括: main, main10, frext
- ◆ 默认值: main

10) <Usage>: 应用场景; 当场景被选择后, 各主要的编码参数会被自动设定为与之匹配的最优值。

■ 有效值:

- ◆ “Record”: 用于录像存储; 此场景下编码画质较高, 输出流中包含大量的 (双向) 预测帧, 因而压缩率也比较高, 但如果用于实时传输其解码时延会比较大。
- ◆ “Live”; 用于实时传输; 此场景下编码画质中等, 输出流中 (双向) 预测帧占比低甚至没有, 因而压缩率相对较低, 但解码时延相对很小且容错能力强。

■ 默认值: Record

## 第7节. <MediaWriters>

<MediaWriters>区段用于配置 StreamMaster 输出（网络推流或保存文件）的音视频流的信息；在这里<XXXCount>实际为 MWCount：它代表输出的多媒体流的数量（注意一个多媒体流可能包含多个音视频流）；该值的最大有效值主要取决于 StreamMaster 版本；

每个视频编码器定义子区段可能包括如下配置项：

- 1) <MWType>：输出流的类型，有效值包括：“RTSP”、“RTMP”、“FILE”
  - 默认值：无；该项必须配置为正确值。
- 2) <outputURL>：输出流的 URL；仅在输出类型**不是**“FILE”的情况下有效；
- 3) <VideoStreams>：输出流中包含的视频流名称列表，其间用半角逗号‘,’分隔；
- 4) <AudioStreams>：输出流中包含的音频流名称列表，其间用半角逗号‘,’分隔；

提示：如果<VideoStreams>与<AudioStreams>列表均为空（包括未配置）意味着所定义的此 MediaWriter 输出中不包含任何音视频流，那么这一组配置是无效的。

对于文件输出流（MWType=‘FILE’），即录像，MediaWriter 包括如下配置项：

- 5) <RootFolder>：录像存储的根目录，该目录在 StreamMaster 启动时必须已经存在并且可写，期间不能删除；StreamMaster 会根据规则在该目录下生成录像文件。
- 6) <Container>：录像文件容器类型，目前仅支持 MP4；
- 7) <SubFolderDate>：是否根据日期建立子目录；如果为 true，录像文件会根据其生成的月份分目录存储；
  - ✓ 有效值：“true”，“false”；
  - ✓ 默认值：“false”；
- 8) <ContentsSuffix>：指定（文字）内容的录像文件名前缀。最大有效长度为 15 个字符；未配置则表示不使用指定前缀。
  - ✓ 举例：ContentsSuffix = “庭审录像”
- 9) <SegmentLength>：录像分隔长度；单位是分钟；如果使用该值，StreamMaster

会在当前录像文件时长达到该值时，自动重新建立一个新的录像文件。

✓ 有效值范围：【5, 120】

✓ 默认值：30

10) <SaveRecordFileList>：是否保存录像文件列表文件；如果该项为"true"，StreamMaster 会在录像文件存储的每个最底层目录下创建一个名为"RecordFileList.txt"的录像文件清单列表文件。

11) <StorageLife>：录像保存时间；单位是天；StreamMaster 会定期检查存量录像的时间，如果其生成日期距离当下的天数已经超过该值，则会将其自动删除以腾出存储空间来存储后续录像。

✓ 有效值范围：【7, 366】

✓ 默认值：183

12) <SpaceWarnLimit>：存储空间不足警戒线；单位是 MB，如果录像文件夹所在存储设备（硬盘）剩余空间低于该值，将会触发告警。

✓ 有效值范围：【40\*1024, 400\*1024】

✓ 默认值：80\*1024 (80GB)

## 第8节. <OSDs>

<OSD>区段用于配置 StreamMaster 的裸 (Raw) 视频流上叠加 OSD 文字信息；它分为多个子区段，每个子区段包含一个 OSD 的定义；其主区段[OSD]仅包含了 OSD 的数量；

- <OSDCount>： OSD 的总数。

<OSD id="##">子区段：(##的有效范围【1, OSDCount】，为该 OSD 的序号)；子区段中定义了序号为##的 OSD 的全部信息；包含如下配置项：

- 1) <DrawOn>： 目标 Stream 的全局名称；亦即此 OSD 将叠加在哪个 Worker 的哪一个输入或者输出视频画面上；；
- 2) <OsdType>： OSD 类型；有效值包括：
  - ◆ "Time"： 当前时间
  - ◆ "Date"： 当前日期
  - ◆ "Datetime"： 当前日期+时间
  - ◆ "StaticText"： 静态文字
  - ◆ "DynamicText"： 动态文字
- 3) <PosBase>： OSD 在画面中的位置基准，有效值包括：
  - ✓ "Center" 或者 "ScreenCenter"： 屏幕/画面正中间
  - ✓ "LeftTopCorner" 或者 "LTCorner"： 屏幕/画面左上角
  - ✓ "RightTopCorner" 或者 "RTCornet"： 屏幕/画面右上角
  - ✓ "RightBottomCorner" 或者 "RBCorner"： 屏幕/画面右下角
  - ✓ "LeftBottomCorner" 或者 "LBCorner"： 屏幕/画面左下角
  - ✓ "TopMiddle"： 屏幕/画面顶部正中
  - ✓ "BottomMiddle"： 屏幕/画面底部正中
  - ✓ "Specified"： 指定位置。
- 4) <XShift>： OSD 相对于 PosBase 所述位置在水平方向上的偏移量（像素），负值为左偏，正值为右偏；建议为 4 的整数倍。
- 5) <YShift>： OSD 相对于 posBase 所述位置在竖直方向上的偏移量（像素），负值为

上偏，正值为下偏；建议为 4 的整数倍。

6) <TextColor>: OSD 文字颜色；有效值包括：

◆ "White", "Yellow", "Red", "Green", "Blue"

7) <TextHeight>: OSD 字体的高度，建议值为 16~128 之间，且为 8 的整数倍。

## 第9节. <OSGs>

<OSGs>区段用于配置在 StreamMaster 视频流的裸视频阶段的每一帧上绘制几何图形或者叠加图片/动画;它分为多个子区段, 每个子区段包含一个 OSG 的定义;其主区段<OSGs>仅包含了 OSG 的数量定义;

- <OSGCount>: 有效 OSG 的总数;

<OSG id="##">子区段: (##的有效范围【1, OSGCount】, 为该 OSG 的序号); 子区段中定义了序号为##的 OSG 的全部信息; 包含如下配置项:

- 1) <DrawOn>: 目标 Stream 的全局名称; 亦即此 OSD 将叠加在哪个 Worker 的哪一个输入或者输出视频画面上;
- 2) <OSGType>: OSG 的类型; 有效的类型名称包括:
  - ◆ Geometry: 几何图形
  - ◆ PICTURE: 图片;
  - ◆ ANIMATE: 动画。
- 3) <CordOrgPos>: 坐标原点位置; 即横纵坐标 0 点在视频画面中的位置; 视频画面的坐标与屏幕相同, 左上角为 0 点, 向右为横坐标递增方向, 向下为纵坐标递增方向。基准点有以下几个别称:
  - ◆ "LTCorner": 画面左上角, 这也是默认值;
  - ◆ "LBCorner": 画面左下角; 向右为横坐标递增方向, 向上为纵坐标递增方向;
  - ◆ "Center": 画面中央; 向右为横坐标递增方向, 向上为纵坐标递增方向;
  - ◆ "Specify": 指定位置; 此时的坐标原点位置由 CntrCrdX 和 CntrCrdY 两个参数来指定。
- 4) <CntrCrdX>, <CntrCrdY>: 指定的 OSG 坐标原点在画面上的位置。

以下配置项仅对图片或动画 (OSGType = PICTURE/ ANIMATE) OSG 有效:

- 5) <SourceFile>: 图片/动画文件名 (全路径)
  - ◆ 当前版本支持的图片格式包括: Jpeg, Gif (推荐), PNG (推荐), BMP。
  - ◆ 当前版本支持的动画/视频格式包括: Gif (推荐), Motion-Jpeg, AVC/MP4;

- 6) <PosX>: 绘制的起点横坐标;
- 7) <PosY>: 绘制的起点纵坐标;
- 8) <fResize>: 缩小比例; 大于 0 小于 1.0 的正浮点数;
  - ◆ 例如: 当该值=0.5 时, 表示其宽和高均缩小至原始尺寸的 1/2 来显示。
  - ◆ 默认值: 1.0, 即按原始尺寸大小显示;
- 9) <StartDelay>: 显示延时; 即 OSG 在视频流开始后多久开始叠加显示, 单位是秒, 该值等于 0 表示视频流开始即叠加 OSG;
- 10) ShowTime: OSG 显示的时长, 单位是秒; 如果大于 0, 则 OSG 在自 StartDelay 起算 ShowTime 秒后停止显示; 该值等于 0 表示一直显示。

补充: OSG 模块在叠加显示图片或动画时, 是支持底色透明的, 这也是用户在视频上叠加图形 (如 Logo) 最常用的选择; 此时请确保所选用的图形是支持透明的格式 (Gif/PNG), 并且在调色板中使用了透明通道 (Alpha) 。

以下配置项仅对动画 OSG (OSGType = ANIMATE) 有效:

- 11) <Repeat>: 是否重复播放,
  - ◆ 有效值: "false": 不重复, 仅播放一次; True: 反复播放。
  - ◆ 默认值: True;
- 12) <FromFrame>, <ToFrame>: 实际所使用的帧范围; 即告诉 StreamMaster 仅使用动画中从 FromFrame 到 ToFrame 之间的帧作为 OSG。
  - ◆ 有效值: 非负整数。
  - ◆ 默认值: 0; 表示使用动画中所有的有效画面帧。
  - ◆ 如果 FromFrame 或 ToFrame 大于动画文件中实际的帧数, 则 OSG 会加载失败; 所以如果不知道文件中有多少帧, 最好的办法是让这两个值均为零;

以下配置项仅对几何图形 OSG (<OSGType>= Geometry) 有效:

- 13) <GmtType>: 几何图形的名称; 当前版本支持的几何图形名称请参考附录 I。
- 14) <posUnit>: 尺寸单位; 有效值包括:

- ◆ “PIXEL”：位置或尺寸数据以像素为单位；
- ◆ “RATIO”：位置或尺寸数据以所占画面长宽的万分比为单位；例如一个点的横坐标在宽度为 1280 的画面 60%（768）的位置，则该坐标对应的配置项的值应该为 6000。

15) <Weight>：线条的粗细，单位是像素；暂未支持。

16) <lineColor>：线条的颜色；有效的颜色别名如下：

- ◆ Blue：蓝色；
- ◆ Green：绿色；
- ◆ Red：红色；
- ◆ Yellow：黄色
- ◆ White：白色。

17) <lineType>：线条的类型；包括以下类型：

- ◆ “Solid”：实线；默认值。
- ◆ “Dashed”：虚线；暂未支持
- ◆ “DashDot”：点划线；暂未支持

18) <Transparent>：图形是否填充（仅对封闭图形（如圆、矩形等）有效）；

- ◆ “false”：不透明，填充图形；
- ◆ “true”：透明，不填充。

19) <fillingColor>：填充色，仅当 Transparent=false 时有效；有效的颜色别名同上 lineColor。

20) <XRagneL>, <XRagneR>：横轴（x）取值范围，仅对于函数型的图形 OSG（如幂函数或正弦函数曲线）有效。

21) <GmtNeedPrmtrs>：Geometry 是否需要而额外配置的参数；

22) <Parameters>附加参数列表（仅当 GmtNeedPrmtrs=true 时有效）：

附件参数是指每种 Geometry 根据自身需要而额外配置的参数，参数数量、名称及其取值因 GmtType 的不同而有别。具体每种 OSG 需要配置哪些参数请参考附录 I。



(1). `<PamName>`: *PamName* 是参数名称; 其中的内容是参数值。

举例:

某椭圆 (`<OSGType>=Geometry`, `<GmtType>=ELLIPSE`) OSG 的参数集定义如下:

```
<Parameters>
```

```
    <CentreX>959</CentreX>
```

```
    <CentreY>539</CentreY>
```

```
    <MajorSemiaxis>600</MajorSemiaxis>
```

```
    <MinorSemiaxis>400</MinorSemiaxis>
```

```
</Parameters>
```

该定义表示, 椭圆的中心在画面坐标的 (959,539) 处, 长半轴为 600, 短半轴为 400。



感谢您选择 StreamMaster（流影大师），由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导。本文档中的所有陈述、信息和建议不构成任何明示或暗示。如您需要更具体的指导或个性化定制，请随时联系我们的技术支持团队。



高盛信息科技研发部

2024年5月

## 附录

### 1. 附录 I OSG 几何图形表

序号	别名	意义	附加参数表		
			参数名称	参数意义	参数值
1	STRAIGHT-LINE	直线	StartX	起点横坐标	整数
			StartY	起点纵坐标	整数
			EndX	终点横坐标	整数
			EndY	终点纵坐标	整数
2	CIRCLE	圆	CentreX	圆心横坐标	整数
			CentreY	圆心纵坐标	整数
			Radius	半径	整数
3	ELLIPSE	椭圆	CentreX	圆心横坐标	整数
			CentreY	圆心纵坐标	整数
			MajorSemiaxis	长半轴	整数
			MinorSemiaxis	短半轴	整数
4 5 6	ARC ARCH SECTOR	圆弧 弓形 扇形	CentreX	圆心横坐标	整数
			CentreY	圆心纵坐标	整数
			Radius	半径	整数
			RadianStart	起点弧度	浮点数
			RadianEnd	终点弧度	浮点数
7	TRIANGLE	三角形	Vertex1X	顶点 1 横坐标	整数
			Vertex1Y	顶点 1 纵坐标	整数
			Vertex2X	顶点 2 横坐标	整数
			Vertex2Y	顶点 2 纵坐标	整数
			Vertex3X	顶点 3 横坐标	整数
			Vertex3Y	顶点 3 纵坐标	整数
8	RECTANGLE	矩形	StartX	起点横坐标	整数

			StartY	起点纵坐标	整数
			Width	宽度	整数
			Height	高度	整数
9	POLYGON	多边形	VertexNum	顶点数量	整数
			Vertex#X	顶点#横坐标	整数
			Vertex#Y	顶点#纵坐标	整数
10	LINEAR	线性函数曲线 (y=kx+b)	K	参数 K	浮点数
			B	参数 B	浮点数
11	POWER	幂函数曲线 (y=x^α)	A	参数 A	正整数
12	SQRT	平方根曲线 y=sqrt(x)	无		
13	PARABOLA	抛物线 y=ax^2+bx+c	A	参数 A	浮点数
			B	参数 B	浮点数
			C	参数 C	浮点数
14	CUBIC	三次方程曲线 y=ax³+bx²+cx+d	A	参数 A	浮点数
			B	参数 B	浮点数
			C	参数 C	浮点数
			D	参数 D	浮点数
15	EXPONENTIAL	指数函数曲线 (y=a^x)	A	底数 A	非 1 正数
16	EXPONENTIAL_E	以 e 为底的指数曲线 (y=e^x)	无		
17	LOGARITHM	对数函数曲线 (y=logaX)	A	底数 A	非 1 正数
18	LOGARITHM_E	以 e 为底的对数函数 曲线 (y=logeX)	无		

19	SINUSOID	正弦曲线	A		
		$(y=A*\sin(Kx-Wt-Q)+D)$	K		
		余弦曲线	W		
20	COSINWAVE	$y=A*\cos(kx-Wt-Q)+D$	T		
			Q		
			D		
21	TANGENT	正切曲线			
22	COTANGENT	余切曲线			
23	ARCSIN	反正弦曲线			
24	ARCCOS	反余弦曲线			
25	ARCTAN	反正切曲线			
26	ARCCOT	反余切曲线			

高盛信息科技研发部

2024年1月