

## 对象创建

---

在页面嵌入 <object ID="pusher" .../> 标签，即创建了 pusher 对象

```
<!-- 注意 pusher 对象的 clsid 为 01502AEB-675D-4744-8C84-9363788ED6D6 -->
<object ID="pusher" CLASSID="CLSID:01502AEB-675D-4744-8C84-9363788ED6D6"
        codebase="./LiteAVAX.cab#version=1,0,0,1" width="640" height="480">
</object>

<!-- 调用pusher对象方法 >
<script>
    function setRenderWndSize() {
        var vW = 640;
        var vH = 480;
        pusher.setRenderWndSize(vW, vH );
    }
</script>
```

## 接口列表

---

名称	描述
getVersion()	获取插件SDK版本号
setRenderWndSize(width, height)	设置当前视频渲染窗口的大小，
enumCameras():	枚举当前的摄像头，
startPreview()	启动摄像头预览
stopPreview()	关闭摄像头预览
stopPush()	停止推流
switchCamera(cameraIndex)	切换摄像头，支持在推流中动态切换，
setMute(bMute)	静音接口
setRenderMode(modeType)	设置图像的渲染（填充）模式
setRotation(rotationType )	设置图像的顺时针旋转角度
setVideoResolution(resolutionType )	设置视频分辨率
setBeautyStyle(beautyStyle , beautyLevel, whitenessLevel)	设置美颜和美白效果
setRenderYMirror(bMirror)	设置预览渲染的镜像效果
setOutputYMirror(bMirror)	设置推流画面的镜像效果
setVideoBitRate(bitrate)	设置视频码率
setAutoAdjustStrategy(adjuststrategy )	设置流控策略
setVideoBitRateMin(videoBitrateMin)	配合 setAutoAdjustStrategy 使用
setVideoBitRateMax(videoBitrateMax)	配合 setAutoAdjustStrategy 使用
setVideoFPS(fps)	设置视频帧率
setPusherEventCallBack(callbackfun, objectid)	设置回调接口

## 1.getVersion()

获取插件版本号，和标签 <object ... codebase='...&version=1.0.0.1'> 上的 version 对应。

- 返回值说明

参数	类型	说明
vRet	String	版本号:x.x.x.x

- 示例代码：

```
var vVersionString = pusher.getVersion()
```

## 2.setRenderWndSize()

设置当前视频渲染窗口的大小，需要和标签 <object ... width="x", height="x" /> 上的 width 和 height 值对齐。启动推拉流前设置。

- 参数说明

参数	类型	说明
width	Int	视频窗口的长
height	Int	视频窗口的宽

- 示例代码：

```
function setRenderWndSize() {
    var vW = 640;
    var vH = 480;
    pusher.setRenderWndSize(vW, vH );
}
```

## 3.enumCameras()

枚举当前的摄像头，推流前需要检查当前 PC 是否安装了摄像头，以及安装了几个摄像头。

- 返回值说明

参数	类型	说明
vRetJson	String	Json格式摄像头列表

- vRetJson样式

- camera\_cnt : Int (摄像头个数)
- cameralist : List (摄像头列表)
  - camera\_name : String (像头名称)
  - id : Int (摄像头编号，注意设置可用摄像头时，是设置编号，参考switchCamera接口)
- 示例 {"camera\_cnt":1,"cameralist":[{"camera\_name":"HD Pro Webcam C920","id":"0"}]}

- 示例代码：

```
function refreshCamera() {
    var vRetString = pusher.enumCameras();
    var obj = JSON.parse(vRetString);
    if (obj.camera_cnt != 0) {
        for (var i = 0; i < obj.camera_cnt; ++i) {
            var objSelect = document.getElementById('cameralistselect');
            objSelect.add(new Option(obj.cameralist[i].camera_name,
            obj.cameralist[i].id));
        }
    }
    else {
        alert("无可用的摄像头");
    }
}
```

## 4.startPreview()

启动摄像头预览，接口调用成功（>0） or 失败（0）

- 返回值说明

参数	类型	说明
vRet	Int	0失败，>0成功

- 示例代码：

```
var vRetInt = pusher.startPreview()
```

## 5.stopPreview()

关闭摄像头预览，stopPush 之前调用此函数并不会停止推流，而是让 SDK 只推送音频。

- 示例代码：

```
pusher.stopPreview()
```

## 6.startPush(sUrl)

启动推流（在 startPush 之前需要先检测摄像头，否则推流可能失败），

- 参数说明

一个合法的推流地址，支持 rtmp 协议（URL 以“rtmp://”打头，腾讯云推流 URL 的获取方法见 [DOC](#)

参数	类型	说明
sUrl	String	推流链接

- 返回值说明

成功 or 失败，内存分配、资源申请失败等原因可能会导致返回失败

参数	类型	说明
vRet	Int	0失败，>0成功

- 示例代码：

```
function doStartPush(sUrl) {  
    var vRetInt = pusher.startPush(sUrl);  
}
```

## 7.stopPush()

停止推流，注意推流 URL 有排他性，也就是一个推流 Url 同时只能有一个推流端向上推流

- 示例代码：

```
pusher.stopPush()
```

## 8.switchCamera(cameraIndex)

切换摄像头，支持在推流中动态切换，cameraIndex 可以通过 enumCameras 函数获取。

- 参数说明

参数	类型	说明
cameraIndex	Int	摄像头编号，默认值-1，第一个摄像头值为0，

- 示例代码：

```
function switchCameraSelect() {  
    var obj = document.getElementById('cameralistselect');  
    var index = obj.selectedIndex; //序号，取当前选中选项的序号  
    var val = obj.options[index].value;  
    pusher.switchCamera(parseInt(val));  
}
```

## 8.setMute(bMute)

关闭麦克风，SDK 会停止采集麦克风的数据，当用户不希望自己的声音被对方听到时，可以使用这个功能。

- 参数说明

参数	类型	说明
bMute	Int	是否静音， 0非静音， 1静音

- 示例代码：

```
function setMute() {
    pusher.setMute(1);
}
```

## 9.setRenderMode(modeType)

设置图像的渲染（填充）模式

- 参数说明

参数	类型	说明
modeType	Int	值为 1：适应，此模式下会显示整个画面的全部内容，但可能有黑边的存在 值为 2：填充，此模式下画面无黑边，但是会裁剪掉一部分超出渲染区域的部分，裁剪 更多可参考：AxTXERenderMode定义

- 示例代码：

```
function setRenderMode() {
    pusher.setRenderMode(AxTXERenderMode.AX_TXE_RENDER_MODE_ADAPT);
}
```

## 10.setRotation(rotationType )

设置图像的顺时针旋转角度

- 参数说明

参数	类型	说明
rotationType	Int	值为 1：保持原图像的角度 值为 2：顺时针旋转90度，最终图像的宽度和高度互换 值为 3： 顺时针旋转180度，最终图像颠倒 值为 4： 顺时针旋转270度，最终图像的宽度和高度互换 更多可参考:AxTXEVideoRotation定义

- 示例代码：

```
function setRotation() {  
    pusher.setRotation(AxTXEVideoRotation.AX_TXE_VIDEO_ROTATION_NONE);  
}
```

## 11.setVideoResolution( )

设置视频分辨率，和码率配合使用，比如 640x360 配合 800kbps 的视频码率，320x240配合 400kbps的视频码率。

- 参数说明

参数	类型	说明
resolutionType	Int	值为 1：RESOLUTION320x240 值为 2：RESOLUTION640x480 值为 3：RESOLUTION480x272 值为 4：RESOLUTION640x360 更多可参考:AxTXEVideoResolution定义

- 示例代码：

```
function setVideoResolution() {  
  
    pusher.setVideoResolution(AxTXEVideoResolution.AX_TXE_VIDEO_RESOLUTION_320x240);  
}
```

## 12.setBeautyStyle

设置美颜和美白效果

- 参数说明

参数	类型	说明
beautyStyle	Int	值为 1：光滑 值为 2：自然 值为 3：朦胧 更多可参考:AxTXEBeautyStyle定
beautyLevel	Int	美颜级别取值范围 1 ~ 9； 0 表示关闭，1 ~ 9值越大，效果越明显
whitenessLevel	Int	美白级别取值范围 1 ~ 9； 0 表示关闭，1 ~ 9值越大，效果越明显

- 示例代码：

```
function setBeautyStyle() {
    var vlBeautyLevel = 5;
    var vlWhitenessLevel = 6;
    pusher.setBeautyStyle(AxTXEBeautyStyle.AX_TXE_BEAUTY_STYLE_SMOOTH,
        vlBeautyLevel, vlWhitenessLevel );
}
```

## 13.setRenderYMirror(bMirror)

设置预览渲染的镜像效果，ActiveX导出接口没有Boolean类型，用Int代替。

- 参数说明

参数	类型	说明
bMirror	Int	1表示画面左右反转，0表示保持原样

- 示例代码：

```
function setRenderYMirror() {
    pusher.setRenderYMirror(1);
}
```

## 14.setOutputYMirror(bMirror)

设置推流画面的镜像效果，ActiveX导出接口没有Boolean类型，用Int代替。

- 参数说明

参数	类型	说明
bMirror	Int	1表示画面左右反转，0表示保持原样

- 示例代码：

```
function setOutputYMirror() {
    pusher.setOutputYMirror(1);
}
```

## 15.setVideoBitRate(bitrate)

设置视频码率，注意，不是分辨率越高画面越清晰，是码率越高画面越清晰

- 参数说明



参数	类型	说明
bitrate	Int	视频码率，单位 kbps，比如 640x360 分辨率需要配合 800kbps 的视频码率

- 示例代码：

```
function setVideoBitRate() {
    pusher.setVideoBitRate(500);
}
```

## 16.setAutoAdjustStrategy(adjuststrategy)

设置流控策略，即是否允许 SDK 根据当前网络情况调整视频码率，以避免网络上传速度不足导致的画面卡顿

- 参数说明

参数	类型	说明
adjuststrategy	Int	<p>值为 -1：无流控，恒定使用 setVideoBitRate 指定的视频码率</p> <p>值为 0：适用于普通直播推流的流控策略，该策略敏感度比较低，会缓慢适应带宽变化，有利于在带宽波动时保持画面的清晰度。</p> <p>值为 1：适用于普通直播推流的流控策，差别是该模式下 SDK 会根据当前码率自动调整出适合的分辨率</p> <p>值为 5：适用于实时音视频通话的流控策略，该策略敏感度比较高，网络稍有风吹草动就会进行自适应调整</p> <p>更多可参考:AxTXEAutoAdjustStrategy定义</p>

- 示例代码：

```
function setAutoAdjustStrategy() {

    pusher.setAutoAdjustStrategy(AxTXEAutoAdjustStrategy.TXE_AUTO_ADJUST_NONE);
}
```

## 17.setVideoBitRateMin & setVideoBitRateMax

配合 setAutoAdjustStrategy 使用，当 AutoAdjust 策略指定为 TXE\_AUTO\_ADJUST\_NONE 时，如下的两个函数调用均视为无效

- 参数说明

参数	类型	说明
videoBitrateMin	Int	允许 SDK 输出的最小视频码率，比如 640x360 分辨率下这个值适合设置为 300kbps
videoBitrateMax	Int	允许 SDK 输出的最小视频码率, 比如 640x360 分辨率下这个值适合设置为 1000kbps

- 示例代码：

```
function setVideoBitRateMin() {
    pusher.setVideoBitRateMin(300);
}
```

## 18.setVideoFPS(fps)

设置视频帧率，请注意：这里的 fps 只是设置最大帧率，具体视频画面的帧率还是由摄像头本身的采集帧率所决定的，不同的摄像头支持的最大帧率有所不同。

- 参数说明

参数	类型	说明
fps	Int	fps - 视频帧率，默认值为15，重启后生效

- 示例代码：

```
function setVideoFPS() {
    pusher.setVideoFPS(15);
}
```

## 19.setPusherEventCallback(callbackfun, objectid)

设置事件回调，用于接收在推流过程中 SDK 所抛出的各种事件，事件列表详见文档接下来的部分。

- 参数说明

参数	类型	说明
callbackfun	Func	回调函数
objectid	Int	回调对象ID

- **paramJson**样式[回调参数JSON格式]
  - eventId : Int （事件ID，参考PusherCallBackEvent定义）
  - objectId : Int （和setPusherEventCallback::objectid一致）
  - paramCnt : Int (JSON携带的Key-Value键值对个数)

- paramJson : List (键值对String)
  - key : String (参考: CBParamJsonKey定义)
  - value: String (参考: CBParamJsonKey定义指向的值含义)
- 示例 {"eventId":200001,"objectId":1,"paramCnt":9,"paramlist":
  - ["key":"AUDIO\_BITRATE","value":"0"],{"key":"CACHE\_SIZE","value":"571"},
  - {"key":"CODEC\_CACHE","value":"329"},{"key":"NET\_SPEED","value":"0"},
  - {"key":"SERVER\_IP","value":""},{"key":"VIDEO\_BITRATE","value":"0"},
  - {"key":"VIDEO\_FPS","value":"14"},{"key":"VIDEO\_HEIGHT","value":"240"},
  - {"key":"VIDEO\_WIDTH","value":"320"}]}
- 示例代码\*\* :

```
//推流前调用。
pusher.setPusherEventCallBack(PusherEventListener, 1);

var PusherEventListener = function (paramJson) {
    var obj = JSON.parse(paramJson);
    if (parseInt(obj.eventId) ==
PusherCallBackEvent.TXE_STATUS_UPLOAD_EVENT && parseInt(obj.objectId) == 1)
    {
        doUpdatePluserStatusInfo(paramJson);
    }
};

function doUpdatePusherStatusInfo(paramJson) {
    var obj = JSON.parse(paramJson);
    if (obj.paramCnt != 0) {
        for (var i = 0; i < obj.paramCnt; ++i) {
            if(obj.paramlist[i].key == CBParamJsonKey.KEY_VIDEO_BITRATE)
                document.getElementById('PUSHVIDEO_BITRATEID').innerHTML =
obj.paramlist[i].value;
            else if(obj.paramlist[i].key ==
CBParamJsonKey.KEY_AUDIO_BITRATE)
                document.getElementById('PUSHAUDIO_BITRATEID').innerHTML =
obj.paramlist[i].value;
            else if(obj.paramlist[i].key == CBParamJsonKey.KEY_VIDEO_FPS)
                document.getElementById('PUSHVIDEO_FPSID').innerHTML =
obj.paramlist[i].value;
            ....
        }
    }
}
```

## 事件列表

## 1. 常规事件

事件ID	数值	含义说明
PLAY_EVT_CONNECT_SUCC	2001	已经连接服务器
PLAY_EVT_RTMP_STREAM_BEGIN	2002	已经连接服务器，开始拉流
PLAY_EVT_RCV_FIRST_I_FRAME	2003	渲染首个视频数据包(IDR)
PLAY_EVT_PLAY_BEGIN	2004	视频播放开始
PLAY_EVT_PLAY_PROGRESS	2005	视频播放进度
PLAY_EVT_PLAY_END	2006	视频播放结束
PLAY_EVT_PLAY_LOADING	2007	视频播放loading
PLAY_EVT_START_VIDEO_DECODER	2008	解码器启动
PLAY_EVT_CHANGE_RESOLUTION	2009	视频分辨率改变

## 2. 错误通知

SDK 遭遇了一些严重错误，比如网络断开等等，这些错误会导致播放无法继续，因此您的代码需要对这些错误进行相应的处理。

事件ID	数值	含义说明
PLAY_ERR_NET_DISCONNECT	-2301	网络断连，且重试亦不能恢复，将导致播放失败
PLAY_ERR_GET_RTMP_ACC_URL_FAIL	-2302	获取加速拉流地址失败，会导致播放失败

## 3. 警告事件

SDK 发现了一些非严重错误，一般不会导致播放停止，所以您可以不关注如下事件，其中：

**PLAY\_WARNING\_VIDEO\_PLAY\_LAG** 是 SDK 对外通知播放卡顿的事件信号，它指的是视频画面的卡顿（两帧画面的刷新时间）超过 500ms。

事件ID	数值	含义说明
PLAY_WARNING_VIDEO_DECODE_FAIL	2101	当前视频帧解码失败
PLAY_WARNING_AUDIO_DECODE_FAIL	2102	当前音频帧解码失败
PLAY_WARNING_RECONNECT	2103	网络断连, 已启动自动重连 (自动重连连续失败超过三次会放弃)
PLAY_WARNING_RECV_DATA_LAG	2104	网络来包不稳: 可能是下行带宽不足, 或由于主播端出流不均匀
PLAY_WARNING_VIDEO_PLAY_LAG	2105	当前视频播放出现卡顿 (用户直观感受)
PLAY_WARNING_HW_ACCELERATION_FAIL	2106	硬解启动失败, 采用软解 (暂不支持)
PLAY_WARNING_VIDEO_DISCONTINUITY	2107	当前视频帧不连续, 可能丢帧
PLAY_WARNING_FIRST_IDR_HW_DECODE_FAIL	2108	当前流硬解第一个I帧失败, SDK自动切软解
PLAY_WARNING_DNS_FAIL	3001	RTMP -DNS解析失败
PLAY_WARNING_SEVER_CONN_FAIL	3002	RTMP服务器连接失败
PLAY_WARNING_SHAKE_FAIL	3003	RTMP服务器握手失败
PLAY_WARNING_SERVER_DISCONNECT	3004	RTMP服务器主动断开

## 枚举类型

### AxTXERenderMode

- 渲染窗口的渲染方式: 4:3的窗口, 渲染16:9视频  

```
var AxTXERenderMode = {
    AX_TXE_RENDER_MODE_ADAPT: 1,           // 适应, 此模式下会显示整个画面的全部内容, 但可能有黑边的存在
    AX_TXE_RENDER_MODE_FILLScreen: 2,     // 填充, 此模式下画面无黑边, 但是会裁剪掉一部分超出渲染区域的部分, 裁剪模式为局中裁剪
};
```

### AxTXEVideoRotation

- 渲染视频旋转  

```
var AxTXEVideoRotation = {
    AX_TXE_VIDEO_ROTATION_NONE: 1,        // 保持原图像的角度
    AX_TXE_VIDEO_ROTATION_90: 2,         // 顺时针旋转90度, 最终图像的宽度和高度互换
};
```

```
AX_TXE_VIDEO_ROTATION_180 : 3,      // 顺时针旋转180度，最终图像颠倒
AX_TXE_VIDEO_ROTATION_270 : 4,      // 顺时针旋转270度，最终图像的宽度和高度互换
};
```

## AxTXEVideoResolution

- 推流视频分辨率 var AxTXEVideoResolution = { // 普屏 4:3  
AX\_TXE\_VIDEO\_RESOLUTION\_320x240 : 1, AX\_TXE\_VIDEO\_RESOLUTION\_640x480 : 2,  
// 宽屏16:9 AX\_TXE\_VIDEO\_RESOLUTION\_480x272 : 3,  
AX\_TXE\_VIDEO\_RESOLUTION\_640x360 : 4, AX\_TXE\_VIDEO\_RESOLUTION\_1280x720 : 7,  
AX\_TXE\_VIDEO\_RESOLUTION\_1920x1080 : 8, };

## AxTXEBeautyStyle

- 设置美颜风格  
var AxTXEBeautyStyle = {  
AX\_TXE\_BEAUTY\_STYLE\_SMOOTH : 0, // 光滑  
AX\_TXE\_BEAUTY\_STYLE\_NATURE : 1, // 自然  
AX\_TXE\_BEAUTY\_STYLE\_BLUR : 2, // 朦胧  
};