

## 对象创建

---

在页面嵌入 <object ID="player" .../> 标签，即创建了 player 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()                                  | 关闭图像渲染        |
| setRenderWndSize(width, height)               | 设置当前视频渲染窗口的大小 |
| startPlay(sUrl)                               | 开始播放拉流        |
| stopPlay()                                    | 停止播放          |
| pause()                                       | 暂停播放          |
| resume()                                      | 恢复播放          |
| isPlaying()                                   | 是否正在播放        |
| setMute(bMute)                                | 静音接口          |
| setRenderMode(modeType)                       | 设置图像的渲染（填充）模式 |
| setRenderYMirror(bool)                        | 设置渲染的镜像效果     |
| setRotation(TXEVideoRotation)                 | 设置图像的顺时针旋转角度  |
| setTXEPlayType(streamType)                    | 标准流或低延时流模式    |
| setPlayerEventCallBack(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 = player.getVersion()
```

## 2.setRenderWndSize(width, height)

设置视频播放窗口的大小，需要和标签 <object ... width="x", height="x" /> 上的 width 和 height 值对齐，启动播放前设置。

- 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() {
    player.setRenderWndSize(640, 480);
}
```

### 3.startPlay(sUrl)

- **参数说明** 开始播放，sURL 为播放地址，目前 ActiveX 插件仅支持 RTMP 播放协议，推流地址

`rtmp://8888.livepush.myqcloud.com/live/8888_teststream?`

`bizid=8888&txSecret=6e18e8db0ff2070a339ab739ff46b957&txTime=5A3E7D7F`

对应的播放地址即为：`rtmp://8888.liveplay.myqcloud.com/live/8888_teststream`

- **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 doStartPlay(sUrl) {
    var vRetInt = player.startPlay(sUrl);
}
```

### 4.stopPlay()

停止播放

- 示例代码：

```
player.stopPlay()
```

### 5.pause()

暂停播放，对于直播或者实时音视频的场景没有暂停的说法：pause = stopPlay + 保留最后一帧视频画面

- 示例代码：

```
player.pause()
```

## 6.resume()

恢复播放

- 示例代码：

```
player.resume()
```

## 7.isPlaying()

是否正在播放

- 返回值说明

| 参数   | 类型  | 说明        |
|------|-----|-----------|
| vRet | Int | 0 未开播，1开播 |

- 示例代码：

```
var vRetInt = player.isPlaying()
```

## 8.setMute(bMute)

静音接口，也就是停止播放对方传来的声音

- 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() {  
    player.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<br>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<br>更多可参考：AxTXERenderMode定义 |

- 示例代码：

```
function setRenderMode() {
    player.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<br>值为 2： 顺时针旋转90度，最终图像的宽度和高度互换<br>值为 3： 顺时针旋转180度，最终图像颠倒<br>值为 4： 顺时针旋转270度，最终图像的宽度和高度互换<br>更多可参考:AxTXEVideoRotation定义 |

- 示例代码：

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

## 11.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() {
    player.setRenderYMirror(1);
}
```

## 12.setTXEPlayType(streamType)

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                                       |
|------------|-----|--|
| streamType | Int | 0表示标准直播流，1低延时流，默认1 更多可参考:AxTXEPlayType定义 |

- 示例代码：

```
function setRenderYMirror() {

player.setTXEPlayType(AxTXEBeautyStyle.AX_PLAY_TYPE_LIVE_RTMP_LOC_DELAY);
}
```

## 13.setPlayerEventCallback(callbackfun, objectid)

设置播放事件回调。

- 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PlayerCallBackEvent定义）
- objectId : Int （和setPlayerEventCallback::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。
player.setPlayerEventCallBack(PlayerEventListener, 1);

var PlayerEventListener = function (paramJson) {
    var obj = JSON.parse(paramJson);
    if (parseInt(obj.eventId) ==
PlayerCallBackEvent.TXE_STATUS_DOWNLOAD_EVENT && parseInt(obj.objectId) ==
1) {
        doUpdatePlayerStatusInfo(paramJson);
    }
};

function doUpdatePlayerStatusInfo(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('PLAYVIDEO_BITRATEID').innerHTML =
obj.paramlist[i].value;
            else if(obj.paramlist[i].key ==
CBParamJsonKey.KEY_AUDIO_BITRATE)
                document.getElementById('PLAYAUDIO_BITRATEID').innerHTML =
obj.paramlist[i].value;
            else if(obj.paramlist[i].key == CBParamJsonKey.KEY_VIDEO_FPS)
                document.getElementById('PLAYVIDEO_FPSID').innerHTML =
obj.paramlist[i].value;
            ....
        }
    }
}
```

## 事件处理

### 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  
};
```

## AxTXEPlayType

- 设置播放端的流类型 var AxTXEBeautyStyle = { AX\_PLAY\_TYPE\_LIVE\_RTMP\_HIGH\_DELAY : 0, // RTMP直播,延时比较高, 1s左右, 适用一个主播, 大量观众场景, 但是可以超过10个用户请求播放视频

```
AX_PLAY_TYPE_LIVE_RTMP_LOC_DELAY : 1, // ActiveX默认采用ACC, RTMP直播加速播放, 延时较低, 500ms左右, 适用双向视频的双人场景或多人视频场景, };
```