

# 视频 WEB 插件开发指南

**版权所有©杭州海康威视数字技术股份有限公司 2020。保留一切权利。**

本文档的任何部分，包括文字、图片、图形等均归属于杭州海康威视数字技术股份有限公司或其关联公司（以下简称“海康威视”）。未经书面许可，任何单位或个人不得以任何方式摘录、复制、翻译、修改本手册的全部或部分。除非另有约定，海康威视不对本手册提供任何明示或默示的声明或保证。

## **责任声明**

在法律允许的最大范围内，本文档是“按照现状”提供，可能存在瑕疵或错误。海康威视不对本文档提供任何形式的明示或默示保证，包括但不限于适销性、质量满意度、适合特定目的、不侵犯第三方权利等保证；亦不对使用或是分发本文档导致的任何特殊、附带、偶然或间接的损害进行赔偿，包括但不限于商业利润损失、系统故障、数据或文档丢失产生的损失。

# 目录

简介.....	5
1.1    前言.....	5
1.2    术语.....	5
1.3    运行环境.....	3
1.4    约束说明.....	3
2.    版本更新.....	3
3.    JS 接口说明.....	4
3.1    JS_STARTSERVICE 启动插件服务接口.....	4
3.2    JS_DISCONNECT 断开服务接口.....	5
3.3    JS_CREATEWND 创建插件窗口接口.....	5
3.4    JS_RESIZE 调整插件窗口大小、位置接口.....	6
3.5    JS_CUTTINGPARTWINDOW 扣除部分插件窗口接口.....	6
3.6    JS_REPAIRPARTWINDOW 扣除插件窗口还原接口.....	7
3.7    JS_HIDEWND 隐去插件窗口接口.....	7
3.8    JS_SHOWWND 插件窗口显示接口.....	8
3.9    JS_DESTROYWND 插件窗口销毁接口.....	9
3.10   JS_WAKEUP 唤醒 WEBCONTROL.EXE 接口.....	9
3.11   JS_SETDOCOFFSET 设置 IFRAME 偏移量接口.....	10
3.12   JS_REQUESTINTERFACE 通用请求响应接口.....	10
3.12.1   申请 RSA 公钥.....	11
3.12.2   初始化.....	12
3.12.3   反初始化.....	16
3.12.4   根据监控点编号视频预览.....	16
3.12.5   根据监控点编号录像回放.....	18
3.12.6   停止所有视频预览.....	20
3.12.7   停止所有录像回放.....	20
3.12.8   销毁播放实例.....	21
3.12.9   获取当前布局.....	21
3.12.10  设置当前布局.....	22
3.12.11  播放抓图.....	23
3.12.12  画面字符叠加.....	24
3.12.13  根据监控点编号批量视频预览.....	25
3.12.14  根据监控点编号批量录像回放.....	26
3.12.15  批量停止播放.....	28
3.12.16  设置接口认证信息参数.....	29
3.12.17  进入全屏.....	30
3.12.18  退出全屏.....	31
3.12.19  获取版本号.....	31
3.12.20  设置时间轴级别.....	32
3.13   JS_SETWINDOWCONTROLCALLBACK 设置消息回调接口.....	33

3.13.1	窗口选中消息.....	33
3.13.2	预览/回放播放消息.....	34
3.13.3	抓图结果消息.....	35
3.13.4	预览紧急录像/回放录像剪辑结果消息.....	36
3.13.5	进入全屏/退出全屏消息.....	36
3.13.6	切换布局消息.....	37
3.13.7	播放窗口双击事件消息.....	37
3.13.8	时间轴级别变化消息.....	38
<b>4.</b>	<b>视频 WEB 插件对接指南.....</b>	<b>38</b>
4.1	开发流程 .....	38
4.2	网页开发指南.....	39
4.2.1	引用 JS 文件 .....	39
4.2.2	创建 WebControl 实例.....	40
4.2.3	启动插件.....	40
4.2.4	监听事件.....	41
4.2.5	申请 RSA 公钥.....	42
4.2.6	插件初始化.....	42
4.2.7	视频预览业务功能.....	43
4.2.8	断开插件服务连接.....	44
4.3	IFRAME 对接指南 .....	44
4.3.1	iframe demo 使用 .....	44
4.3.2	iframe 对接步骤.....	44
<b>5.</b>	<b>视频 WEB 插件 UI 集成控件列表.....</b>	<b>45</b>
<b>6.</b>	<b>错误码定义 .....</b>	<b>47</b>
<b>7.</b>	<b>附录.....</b>	<b>48</b>
7.1	如何获取 APPKey 和 APPSECRET .....	48
7.2	依赖 OPENAPI 接口汇总 .....	48
7.3	问题排查 .....	48

# 简介

## 1.1 前言

非常感谢您使用我们公司的产品，我们将竭诚为您提供最好的服务。

本手册可能包含技术上不准确的地方或文字错误。

本手册的内容将做定期的更新，恕不另行通知,更新的内容将会在本手册的新版本中加入。

我们随时会改进或更新本手册中描述的产品或程序。

当您阅读该使用手册时，同时应该拿到本文档、以及以下文件：

表 1.1-1 文件清单

文件名称	描述
VideoWebPlugin.exe	视频 WEB 插件安装包
demo_window_integration.html	视频预览、回放全功能 Demo，可使用此 Demo 验证功能
demo_window_simple_preview.html	视频预览简化 Demo，可参照此 Demo 开发
demo_window_simple_playback.html	录像回放简化 Demo，可参照此 Demo 开发
demo_for_iframe.html	iframe 对接 demo，依赖 WEB 服务，请参照 <a href="#">iframe 对接指南</a>
demo_embedded_for_iframe.html	iframe 对接嵌入 html 文件，依赖 WEB 服务，请参照 <a href="#">iframe 对接指南</a>
webs.exe	WEB 服务程序，请参照 <a href="#">iframe 对接指南</a>
jsencrypt.min.js	用于 RSA 公钥加密的 js 库文件
jsWebControl-1.0.0.min.js	用于与视频 WEB 插件交互的 js 文件

## 1.2 术语

本文涉及相关术语如表 1.2-1 和图 1.2-1、图 1.2-2：

表 1.2-1 术语解释

名称	描述
视频 WEB 插件	视频 WEB 插件（以下简称 VideoWebPlugin）用于跨浏览器开发 WEB 视频应用，提供了实时视频播放、历史视频回放功能。安装 VideoWebPlugin.exe 后，系统会自动运行 WebControl.exe 的程序，并由 jsWebControl-1.0.0.min.js 与之交互完成视频的各项功能
DIV 窗口	指前端提供的非弹出页面的 DIV 标识的窗口
插件窗口	指视频 WEB 插件提供的窗口，该窗口是 DIV 窗口的子窗口，若 DIV 窗口尺寸发生变化，该窗口尺寸也随之变化。如图 1.2-1，整个图中的窗口称为插件窗口
插件服务	指管理插件窗口与 DIV 窗口交互、前端与视频 WEB 插件交付的服务，该服务由 WebControl.exe 统一管理

播放窗口	播放视频的窗口，如图 1.2-1 和图 1.2-2 中的“播放窗口”
工具条	指播放窗口播放视频时，鼠标移动至播放窗口内弹出来的上下工具条，如图 1.2-1 和图 1.2-2 所示“上工具条”和“下工具条”
工具栏	指集合对所有播放窗口控制都起效（如“全部抓图”）、或对选中窗口控制起效（如录像回放中的“暂停”按钮）、或对整个插件窗口控制（如“进入全屏模式”按钮）按钮的窗口，如图 1.2-1 和图 1.2-2 所示“工具栏”
时间轴	指录像回放中展示当前选中播放窗口在查询录像时间段内的录像段，如图 1.2-2 所示“时间轴”
智能信息	指在视频编码设备上配置的一些事件后，显示在播放画面上的信息，如配置移动侦测事件后再画面上显示的线框
多平台接入（接入多平台）	指在基于插件开发的客户端上支持同时预览多个平台的实时视频或播放多个平台上的录像。此种情况下，在插件初始化中不能指定 appkey、secret、ip 和 port，必须初始化完后通过接口来设置这些信息
父页面	相对子页面而言，在 iframe 对接场景下，在该页面嵌入子页面，这个页面称为父页面
子页面	在 iframe 对接场景下，嵌入父页面的页面称为子页面

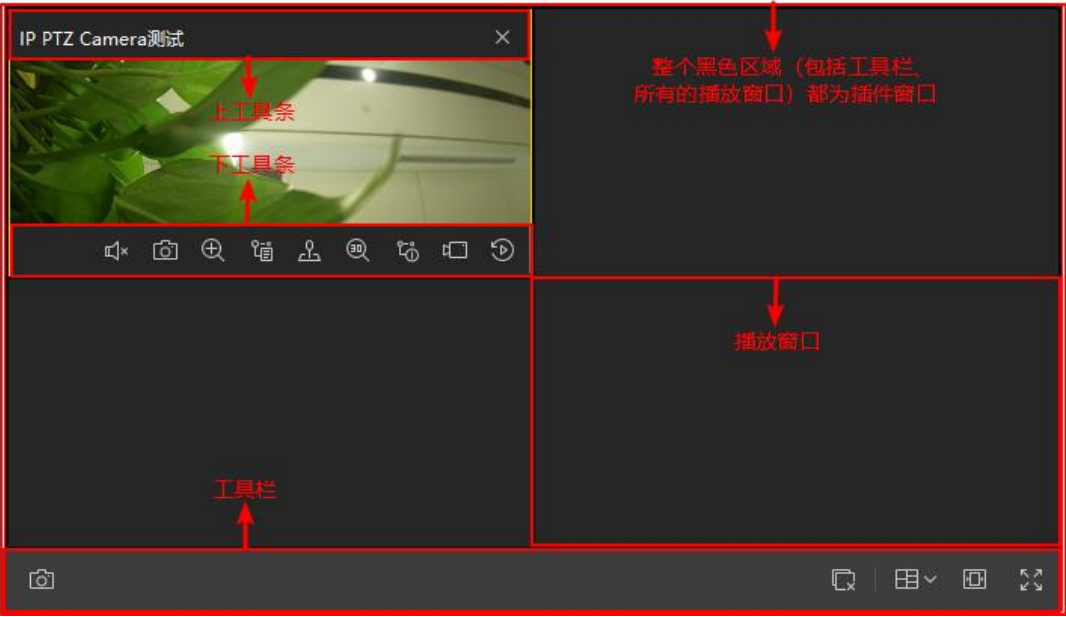


图 1.2-1 视频 WEB 插件（实时预览）UI 模块分类

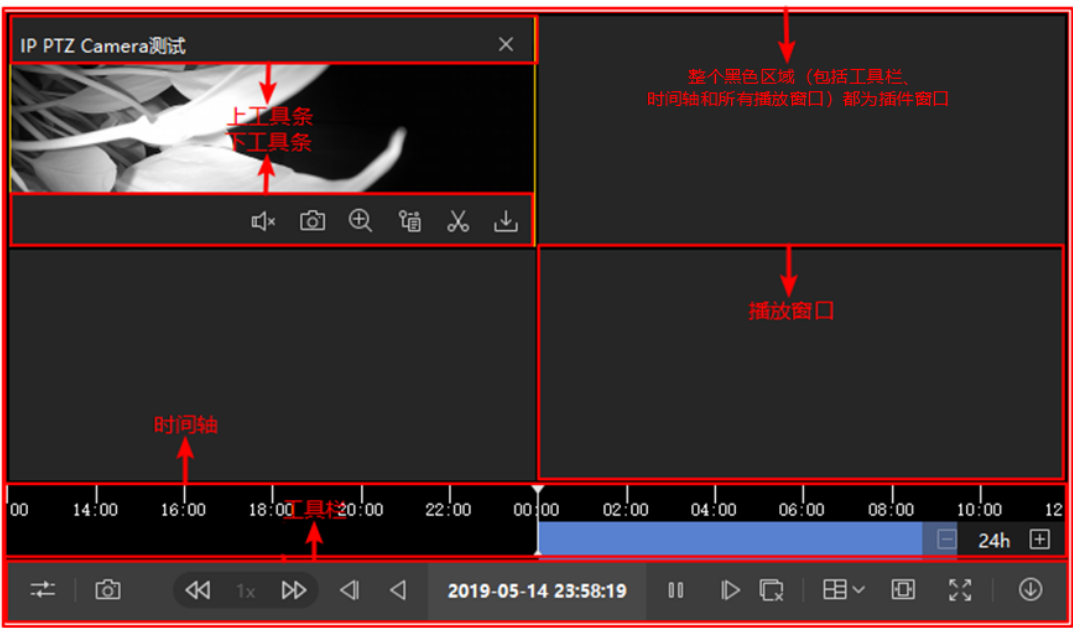


图 1.2-2 视频 WEB 插件（录像回放）UI 模块分类

### 1.3 运行环境

- Win7 32/64 位操作系统：32 位 IE10（兼容 64 位 IE10，64 位 IE10 环境下浏览器会默认启动 32 位 IE）、32/64 位 IE11、32/64 位 Chrome45.0 及以上版本、32/64 位 Firefox52.0 及以上版本，
- Win10 32/64 位操作系统：32/64 位 IE11、32/64 位 Chrome45.0 及以上版本、32/64 位 Firefox51.0 及以上版本

### 1.4 约束说明

- 触摸屏 Windows 操作系统未经全面测试，不推荐使用
- 非企业版 Windows 操作系统未经测试，请使用企业版 Windows 操作系统
- 语音对讲只支持海康 SDK 协议、ehome 协议、国标协议、ISUP 协议和直连萤石接入设备
- iframe 对接存在插件窗口无法及时更新位置导致用户体验问题，并且被嵌入的页面需要禁用滚动条，请谨慎使用。
- IE、Chrome、Firefox 浏览器对接时不建议启动多个插件服务，一般情况下一个 Tab 页只启动一个插件服务
- 低帧率（帧率低于 5FPS）情况下即时回放打开声音会出现声音慢出或少几秒情况

## 2. 版本更新

#### V1.5.2

1、初始化支持设置时间轴级别，详见 <a href="#">3.12.3 初始化</a>
2、新增设置时间轴级别接口，支持跨级别设置时间轴，详见 <a href="#">3.12.20 设置时间轴级别</a>
3、新增时间轴级别变化消息，详见 <a href="#">3.13.8 时间轴级别变化消息</a> ；

#### V1.5.1

- |  |
|--|
| 1、新增支持工具栏按钮自定义，详见 <a href="#">3.12.2 初始化</a> ；   |
| 2、新增支持判断点位是否为级联点位相关功能，详见 <a href="#">3.12.13 根据监控点编号视频预览</a> 和 <a href="#">3.12.14 根据监控点编号录像回放</a> ； |
| 3、新增支持播放窗口双击事件消息，详见 <a href="#">3.13.7 播放窗口双击事件消息</a> ；  |

#### V1.5.0

- |                                    |
|------------------------------------|
| 1、新增支持同时接入多个平台的实时预览、录像回放；          |
| 2、新增支持录像锁定相关功能；                    |
| 3、新增支持紧急录像、录像剪辑展示已录像时长，支持布局切换消息回调； |
| 4、新增云台控制状态下滚动滚轮调焦功能；               |
| 5、新增进入全屏、退出全屏请求功能；                 |
| 6、支持 iframe 对接，嵌入页面需禁用滚动条；         |

#### V1.4.1

- |                                    |
|------------------------------------|
| 1、新增一键静音等功能；                       |
| 2、画面字符叠加支持自定义字号、字体是否加粗以及指定字符串对齐方式； |
| 3、新增批量预览回放、批量停止功能；                 |

#### V1.4.0

- |                 |
|-----------------|
| 1、新增断流重连功能；     |
| 2、新增直连萤石预览回放功能； |
| 3、新增直连萤石语音对讲功能； |
| 4、解决已知缺陷问题；     |

## 3. JS 接口说明

### 3.1 JS\_StartService 启动插件服务接口

#### ■ 接口定义

JS\_StartService(szType, options)

#### ■ 功能说明

启动插件服务

#### ■ 使用条件

插件实例化成功后启动插件服务

#### ■ 入参

名称	描述	是否必填	备注
szType	服务类型	是	请固定填充"window"
options	可选参数对象	否	请固定填充{dllPath: "./VideoPluginConnect.dll"}

#### ■ 返回值

Promise。接口成功说明服务启动成功，失败则说明服务启动失败

#### ■ 使用示例

```
oWebControl.JS_StartService("window", { // oWebControl 为 WebControl 的对象
    dllPath: "./VideoPluginConnect.dll"
}).then(function(){
```



```
        // 服务启动成功
    },function(){
        // 服务启动失败
    });
```

- 特殊使用场景说明  
无

### 3.2 JS\_Disconnect 断开服务接口

- 接口定义  
JS\_Disconnect()
- 功能说明  
断开与插件服务的连接
- 使用条件  
插件服务已启动，一般在前端 unload 事件、路由离开页面时调用
- 入参  
无
- 返回值  
Promise。接口成功说明服务启动成功，失败则说明服务启动失败
- 使用示例

```
oWebControl.JS_Disconnect().then(function(){    // oWebControl 为 WebControl 的对象
    // 断开与插件服务的连接成功
},function(){
    // 断开与插件服务的连接失败
});
```

- 特殊使用场景说明  
IE 浏览器中使用插件存在插件窗口销毁耗时导致浏览器页面消息后可能还会显示一会，可在调 JS\_Disconnect 前调 JS\_HideWnd 接口先隐去插件窗口。

### 3.3 JS\_CreateWnd 创建插件窗口接口

- 接口定义  
JS\_CreateWnd(szId, iWidth, iHeight, options)
- 功能说明  
创建插件窗口，并且插件窗口始终置顶。若不想插件置顶，显示其它控件，详见 3.6 和 3.7 中的接口描述。
- 使用条件  
插件实例化成功后启动插件服务
- 入参

名称	描述	是否必填	备注
szId	元素 ID	是	该元素 ID 标识的窗口作为插件的父窗口
iWidth	元素 ID 标识窗口的宽度	是	使用元素 ID 标识的窗口的宽度使插件窗口与 DIV 窗口重叠
iHeight	元素 ID 标识窗口的高度	是	使用元素 ID 标识的窗口的高度使插件窗口与 DIV 窗口重叠

options	可选参数对象	否	详见示例。非 iframe 对接场景无需填充此参数，iframe 对接需填充此参数。iframe 对接场景该参数使用请参照 demo_embedded_for_iframe.html 文件
---------	--------	---	--

■ 返回值

Promise。接口成功说明创建插件窗口成功，失败则说明创建插件窗口失败

■ 使用示例

```
oWebControl.JS_CreateWnd("playWnd", 600, 400, {
  cbSetDocTitle: function (uuid){    // uuid 为插件提供的 UUID
  }
}).then(function(){    // oWebControl 为 WebControl 的对象
  // 创建插件窗口成功
},function(){
  // 创建插件窗口失败
});
```

■ 特殊使用场景说明

无

## 3.4 JS\_Resize 调整插件窗口大小、位置接口

■ 接口定义

JS\_Resize(iWidth, iHeight)

■ 功能说明

插件窗口无法感知 DIV 窗口的大小、位置变化，通过此接口来调整插件窗口大小与位置

■ 使用条件

创建插件顶层窗口后，前端 DIV 窗口 resize、页面 scroll 事件触发时都需调此接口来调整插件窗口大小与位置

■ 入参

名称	描述	是否必填	备注
iWidth	DIV 窗口宽度	是	
iHeight	DIV 窗口高度	是	

■ 返回值

无

■ 使用示例

```
oWebControl.JS_Resize(700, 500);    // oWebControl 为 WebControl 的对象
```

■ 特殊使用场景说明

无

## 3.5 JS\_CuttingPartWindow 扣除部分插件窗口接口

■ 接口定义

JS\_CuttingPartWindow(iLeft, iTop, iWidth, iHeight)

■ 功能说明

扣除插件窗口部分窗口。

#### ■ 使用条件

插件窗口创建后，会始终置顶，因此当和其它组件一同使用时，会遮挡其它组件内容，该接口的作用是，当部分插件窗口不需要置顶时，我们会隐去部分插件窗口。

#### ■ 入参

名称	描述	是否必填	备注
iLeft	距离插件窗口左边距	是	
iTop	距离插件窗口上边距	是	
iWidth	需要扣除的宽度	是	
iHeight	需要扣除的高度	是	

#### ■ 返回值

无

#### ■ 使用示例

```
oWebControl.JS_CuttingPartWindow(0, 0, 100, 100); // oWebControl 为 WebControl 的对象
```

#### ■ 特殊使用场景说明

无

## 3.6 JS\_RepairPartWindow 扣除插件窗口还原接口

#### ■ 接口定义

JS\_RepairPartWindow(iLeft, iTop, iWidth, iHeight)

#### ■ 功能说明

还原扣除部分窗口后的插件窗口

#### ■ 使用条件

和 3.5 中的接口配合使用，当需要完全显示插件窗口时，使用该接口显示已经隐去的部分插件窗口。

#### ■ 入参

名称	描述	是否必填	备注
iLeft	扣除窗口的顶点距离插件窗口左边距	是	
iTop	扣除窗口的顶点距离插件窗口上边距	是	
iWidth	扣除窗口的宽度	是	
iHeight	扣除窗口的高度	是	

#### ■ 返回值

无

#### ■ 使用示例

```
oWebControl.JS_RepairPartWindow(0, 0, 100, 100); // oWebControl 为 WebControl 的对象
```

#### ■ 特殊使用场景说明

无

## 3.7 JS\_HideWnd 插件窗口隐去接口

#### ■ 接口定义

#### JS\_HideWnd()

- 功能说明  
不在页面展示插件窗口

- 使用条件  
插件窗口创建成功

- 入参  
无

- 返回值  
无

- 使用示例

`oWebControl.JS_HideWnd();` // oWebControl 为 WebControl 的对象

- 特殊使用场景说明

一个浏览器页面使用多个 DIV 窗口加载了多个插件窗口时，插件窗口全屏后其它窗口会处于全屏窗口上，针对此场景如下处理：

- ✓ JS\_SetWindowControlCallback 设置的消息回调中监听窗口全屏事件
- ✓ 监听到窗口全屏事件时调 JS\_HideWnd 插件窗口隐去接口，不显示除接收到全屏事件的插件窗口外的其它窗口
- ✓ 监听到窗口退出全屏事件时调 JS\_ShowWnd 插件窗口显示窗口对接收到退出全屏事件的插件窗口外的其它窗口显示

### 3.8 JS\_ShowWnd 插件窗口显示接口

- 接口定义

JS\_ShowWnd()

- 功能说明  
显示插件窗口

- 使用条件  
可调此接口来恢复插件窗口的显示

- 入参  
无

- 返回值  
无

- 使用示例

`oWebControl.JS_ShowWnd();` // oWebControl 为 WebControl 的对象

- 特殊使用场景说明

一个浏览器页面使用多个 DIV 窗口加载了多个插件窗口时，插件窗口全屏后其它窗口会处于全屏窗口上，针对此场景如下处理：

- ✓ JS\_SetWindowControlCallback 设置的消息回调中监听窗口全屏事件
- ✓ 监听到窗口全屏事件时调 JS\_HideWnd 插件窗口隐去接口，不显示除接收到全屏事件的插件窗口外的其它窗口
- ✓ 监听到窗口退出全屏事件时调 JS\_ShowWnd 插件窗口显示窗口对接收到退出全屏事件的插件窗口外的其它窗口显示

### 3.9 JS\_DestroyWnd 插件窗口销毁接口

- 接口定义  
JS\_DestroyWnd()
- 功能说明  
销毁插件窗口
- 使用条件  
当不需要视频播放时，通过此接口来销毁插件窗口
- 入参  
无
- 返回值  
Promise。接口成功说明销毁插件窗口成功，失败则说明销毁插件窗口失败
- 使用示例  

```
oWebControl.JS_DestroyWnd().then(function(){ // oWebControl 为 WebControl 的对象
    // 销毁插件窗口成功
},function(){
    // 销毁插件窗口失败
});
```
- 特殊使用场景说明  
浏览器页面需随时启用和禁用视频播放的场景，可通过 JS\_DestroyWnd 销毁插件窗口来禁用视频播放，可通过 JS\_CreateWnd 来重新启用视频播放

### 3.10 JS\_WakeUp 唤醒 WebControl.exe 接口

- 接口定义  
JS\_WakeUp(szProtocal)
- 功能说明  
当 WebControl.exe 未启动时唤醒 WebControl.exe。若 WebControl.exe 已启动则忽略
- 使用条件  
VideoWebPlugin.exe 已正确安装，若 Webcontrol.exe 进程异常退出或者 Webcontrol 进程连接失败后，可以使用该接口启动 Webcontrol 进程。Webcontrol 的静态方法，无需实例化，直接调用即可，详见 4.2.3 [启动插件](#)
- 入参

名称	描述	是否必填	备注
szProtocal	唤醒协议	是	由 VideoWebPlugin.exe 安装时写入注册的唤醒协议，固定为 "VideoWebPlugin://"

- 返回值  
无
- 使用示例  

```
WebControl.JS_WakeUp("VideoWebPlugin://") //详见 3.1 创建插件实例
```
- 特殊使用场景说明  
无

### 3.11 JS\_SetDocOffset 设置 iframe 偏移量接口

- 接口定义  
JS\_SetDocOffset(offset)
- 功能说明  
iframe 对接时，通过此接口来设置 DIV 窗口与文档的偏移量，使达到插件窗口与 DIV 窗口贴合的效果
- 使用条件  
iframe 对接场景。
- 入参

名称	描述	是否必填	备注
offset	当 iframe 嵌套时，需要传入该 iframe 相对最顶层窗口的位置	是	形式： { left: 100, top: 100 }

- 返回值  
无
- 使用示例  

```
WebControl.JS_SetDocOffset ({  
    left: 100,  
    top: 100  
}) //详见 3.1 创建插件实例
```
- 特殊使用场景说明  
iframe 对接场景

### 3.12 JS\_RequestInterface 通用请求响应接口

- Js\_RequestInterface 是通用请求接口，用于完成各种功能，功能参数由其 json 报文参数决定。json 报文参数格式：

```
{  
    funcName: "funcName", // 功能标识，详见下表  
    argument: "argument" // 功能标识的参数，如果无参数可不传  
}
```

例如，对申请 RSA 公钥的请求 json 报文如下：

```
{  
    funcName: "funcName",  
    argument: "{  
        keyLength: 1024  
    }"  
}
```

- 通用请求接口返回 json 报文，只需要关注 responseMsg 信息，其格式如下：

```
{
```

```
code: 0,           // 错误码 0-成功 1-失败
msg: "invalid param", // 错误描述, 仅当 errorCode 非 0 时才有错误描述
data: ""           // 返回的数据, 如 RSA 公钥
}
```

例如, 对申请 RSA 公钥请求的响应, 报文如下:

```
{
  code: 0, // 错误码 0-成功 1-失败
  data: "{
    rsaPubKey: "MIBISFNUKEMGGNJSIGWMFIGGIMWUEIGOMWIUFIEMGKOMGJ" // RSA 公钥
  }"
}
```

3.12.1 申请 RSA 公钥

- 功能说明  
申请 RSA 公钥, 用于对敏感信息 (如 appkey、secret) 加密。
- 使用条件  
创建插件实例之后, 销毁插件实例之前。使用完后如还需加密传输, 建议重新申请公钥。
- 功能标识: getRSAPubKey
- 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“getRSAPubKey”
argument	string	功能参数	是	object 格式的字符串
+keyLength	int	秘钥长度, 可选 1024 或 2048	是	如指定非可选值, 返回错误

- Js\_RequestInterface 通用请求入参举例

```
{
  funcName: "getRSAPubKey",
  argument: "{
    keyLength: 1024
  }"
}
```

- 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	
data	string	RSA 公钥	否	base64 编码后的 RSA 公钥字符串。当 code 为 -1 时无此字段, 使用时填充方式为 RSA_PKCS1_PADDING

- 返回值举例

```
{
  code: 0, // 错误码 0-成功 1-失败
  data: {
    rsaPubKey: "MIBISFNUKEMGGNJSIGWMFIGGIMWUEIGOMWIUFIEMGKOMGJ" // 公钥
  }
}
```

```
}
}
```

### 3.12.2 初始化

#### ■ 功能说明

设置平台信息（APPKey、APPSecret、平台 IP 地址和平台端口），设置界面显示效果（显示预览还是回放界面，初始化布局）以及抓图存储文件路径。

#### ■ 使用条件

创建插件实例后除申请 RSA 公钥功能外，调其它接口前应先初始化。

#### ■ 功能标识：init

#### ■ 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“init”
argument	string	功能参数	是	object 格式的字符串
+appkey	string	平台提供的 APPKey	否	如何获取 APPKey 详见 <a href="#">如何获取 APPKey 和 APPSecret</a> ，指定此字段时只能接入一个平台，并且“secret”、“ip”、“port”字段必须指定，否则失败。不指定“appkey”、“secret”、“ip”、“port”，需通过“ <a href="#">设置接口认证信息参数</a> ”接口设置这些信息
+secret	string	平台提供的 APPSecret	否	如何获取 APPKey 详见 <a href="#">如何获取 APPKey 和 APPSecret</a> ，指定此字段时只能接入一个平台，并且“appkey”、“ip”、“port”字段必须指定，否则失败。不指定“appkey”、“secret”、“ip”、“port”，需通过“ <a href="#">设置接口认证信息参数</a> ”接口设置这些信息
+ip	string	平台（Nginx）IP 地址	否	指定此字段时只能接入一个平台，并且“appkey”、“secret”、“port”字段必须指定，否则失败。不指定“appkey”、“secret”、“ip”、“port”，需通过“ <a href="#">设置接口认证信息参数</a> ”接口设置这些信息
+enableHTTPS	int	是否启用 HTTPS 协议与平台交互	否	如指定为非 0 值，则表示使用 HTTPS 协议与平台进行网络通信，如未指定该字段或指定为 0，表示使用 HTTP 协议与平台进行网络通信。目前仅支持 HTTPS，请填写非 0 值
+port	int	平台端口	否	目前只支持 HTTPS 协议端口。一般情况下，HTTPS 协议端口默认为 443。使用时请根据实际环境指定。指定此



				字段时只能接入一个平台，并且“appkey”、“secret”、“ip”字段必须指定，否则失败。不指定“appkey”、“secret”、“ip”、“port”，需通过 <a href="#">“设置接口认证信息参数”</a> 接口设置这些信息
+playMode	int	初始播放模式	否	0-预览 1-回放，如未指定则使用默认值 0，如指定非可选值，返回错误
+encryptedFields	string	加密字段	否	填充哪些是加密了的字段，多个间以“，”分割，当前支持对 appkey、secret、ip、snapDir、layout、videoDir 中的一个或多个加密，注意加密使用同一个公钥分别对各个字段的值加密
+snapDir	string	抓图存储文件路径	否	如未指定或指定值无效，使用默认值：{VideoPlugin 运行目录}/SnapShot，默认抓图类型为 JPG，命名规则为{监控点名称}_{抓图时间戳}（如“测试监控点_1543390276324”）
+layout	string	playMode 指定模式的布局	否	如未指定或指定值无效，使用默认值“2x2”布局，可选值有“1x1”、“2x2”、“3x3”、“4x4”、“5x5”、“1x2”、“1+2”、“1+5”、“1+7”、“1+8”、“1+9”、“1+12”、“1+16”、“4+9”、“1+1+12”、“3+4”、“1x4”、“4x6”
+videoDir	string	紧急录像或录像剪辑存储文件路径	否	如未指定或指定值无效，使用默认值：{VideoPlugin 运行目录}/Video，命名规则为{监控点名称}_{录像时间戳}.mp4（如“测试监控点_1543390276324.mp4”），请确保磁盘剩余控件大于 1G，否则会影响工具条上的紧急录像或者录像剪辑功能
+showToolBar	int	是否显示工具栏	否	不指定或指定字段值为非 0，则默认显示，如指定字段值为 0，则不显示
+showSmart	int	是否显示智能信息	否	不指定或指定字段值为非 0，则默认显示，如指定字段值为 0，则不显示
+buttonIDs	string	工具条按钮字符串，多个之间以“，”分割	否	工具条分为上下两个工具条，上工具条包括监控点名称控件，监控点类型控件和关闭按钮；预览下工具条按钮有声音、抓图、云台控制、3D 放大、电子放大、语音对讲、显示监控点信息、主子码流切换、紧急录像、鹰眼和即时回放按钮；自定义按钮定义（括号内为对应 10 进

				<p>制 ID 值):</p> <p>0x0000(0)- 监 控 点 名 称 按 钮</p> <p>0x0001(1)-监控点类型 0x0010(16)-关闭按钮</p> <p>0x0100(256)- 预 览 回 放 声 音</p> <p>0x0101(257)- 预 览 回 放 抓 图</p> <p>0x0102(258)- 预 览 回 放 电 子 放 大</p> <p>0x0103(259)-预览回放显示监控点信息 0x0104(260)-小鹰眼 0x0200(512)-预览云台控制 0x0201(513)-预览 3D 放大 0x0202(514)- 预 览 语 音 对 讲</p> <p>0x0203(515)- 预 览 主 子 码 流 切 换</p> <p>0x0204(516)- 预 览 紧 急 录 像</p> <p>0x0205(517)- 预 览 即 时 回 放</p> <p>0x0300(768)- 回 放 录 像 剪 辑</p> <p>0x0301(769)-回放录像下载</p> <p>0x0302(770)-回放录像锁定</p> <p>下工具条按指定按钮字符串中按钮 ID 顺序来显示</p> <p>未指定该字段，则默认显示该模式下支持的所有工具条按钮；指定为空字符串时不显示工具条；指定重复、超出可选范围值、非法值将返回失败；指定功能不支持的按钮 ID 将不显示该按钮（如预览模式指定下载按钮），如去除不支持的按钮 ID 后无其它指定按钮 ID 时，工具条不显示</p> <p>注：0x0201(513)- 预 览 3D 放 大 0x0202(514)- 预 览 语 音 对 讲 0x0302(770)-回放录像锁定 三个按钮需要 openapi 相关接口(参照 7.2 章节)支持，若无接口支持，不建议指定相关按钮。</p>
+reconnectTimes	int	重连次数	否	0-无限重连 >0-重连次数 <0-不重连，不填默认无限重连
+reconnectDuration	int	重连间隔时间	否	取值不小于 1 秒，如小于 1 秒，按 1 秒计算。不填默认重连间隔时间 5 秒
+language	string	配置多语言模式	否	由使用方保证该参数的有效性，不填默认为中文。目前支持“zh_CN”-中文 “en_US”-英文。
+ toolBarButtonIDs	string	工具栏按钮字符串，多个之间以“,”分割	否	工具栏上，预览回放通用按钮有：全部静音，全部抓图，全部关闭，切换布局，切换全屏，全部自适应，分隔

				<p>条；</p> <p>工具栏上仅和回放相关的按钮有：同步异步，速度控件，倒放切换按钮，单帧退，时间，正放切换按钮，单帧进，下载中心；</p> <p>对应的按钮定义为（括号内为对应 10 进制 ID 值）：</p> <p>0x800(2048)-同步异步 0x801(2049)-全部静音 0x802(2050)全部抓图 0x900(2304)速度控件 0x901(2305)倒放切换按钮 0x902(2306)单帧退 0x903(2307)时间 0x904(2308)正放切换按钮 0x905(2309)单帧进 0x1000(4096)全部关闭 0x1001(4097)切换布局 0x1002(4098)切换全屏 0x1003(4099)全部自适应 0x1004(4100)下载中心 0x1200 (4608)分隔条 0x1200 (4609)分隔条 2</p> <p>该字段仅控制工具栏上的按钮显/隐，不支持自定义顺序，配合 showToolbar 字段使用。显示工具栏时，不传该字段则默认显示该模式下所有工具栏按钮；指定为空字符串时，显示工具栏，但无按钮；指定重复、非法值将返回失败。指定功能不支持的按钮 ID 将不显示该按钮（如预览模式指定下载中心按钮）。预览仅支持分隔条 1，回放支持分隔条 1 和分隔条 2</p>
+ levelType	string	时间轴级别	否	<p>如未指定或指定值为字符串但是参数无效，使用默认值 24h。该字段取值范围”24h, 12h, 6h, 1h, 36m, 24m, 12m”，超出取值范围无效。</p>

#### ■ 通用请求入参举例

```
{
  funcName: "init",
  argument: "{
    appkey: "afsgnhmj34567dgh",    // API 网关提供的 appkey
    secret: "vgkk3g0jaoj0igoigj",  // API 网关提供的 secret
    ip: "10.33.31.4",               // API 网关 IP 地址
    port: 9016,                     // API 网关端口
    playMode: 0,                    //播放模式（决定显示预览还是回放界面），0-预览 1-录像播放
    encryptedFields: "appkey,secret", //secret 和 appkey 已加密，对多个字段加密存在初始化耗时问题
    snapDir: "D:\\SnapDir",         // 抓图存储文件路径
  }
```

```
        layout: "2x2"                // 初始化 2x2 布局
        showToolBar: 1,              // 显示工具栏
        showIntelligent: 1,          // 显示智能信息
        buttonIDs: "0,16,256,257,258"
        toolBarButtonIDs: "2048,2049,2050,2304,2306,2305,2307,2308,2309,4096,4608,4097,4099,4098,4609,4100",
    }"
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
    code: -1,                        // 错误码
    msg: "invalid param: missing field of secret" // 错误描述
}
```

3.12.3反初始化

■ 功能说明

插件反初始化。

■ 使用条件

销毁插件前需要对插件进行反初始化，关闭网页或页面时视频 WEB 插件 V1.1.0 开始不用反初始化，视频 WEB 插件内部可捕获到网页或页面的关闭触发自动反初始化。

■ 功能标识：uninit

■ 输入参数

funcName	string	功能标识	是	只能填写“uninit”
----------	--------	------	---	--------------

■ 通用请求入参举例

```
{
    funcName: "uninit"
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
    code: 0 // 错误码
}
```

3.12.4根据监控点编号视频预览

■ 功能说明

实时视频播放。

■ 使用条件

初始化为预览模式后即可进行视频预览。注意，使用直连萤石预览功能时，在初始化时要选择启用 https，若使用 http，则不支持直连萤石预览。

■ 功能标识：startPreview

■ 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“startPreview”
argument	string	功能参数	是	
+cameraIndexCode	string	监控点编号	是	
+authUuid	string	<a href="#">设置接口认证信息参数</a> 接口中设置 authUuid	否	多平台接入时需要提供此字段用于区分具体的平台，来源于 <a href="#">设置接口认证信息参数</a> 接口设置的 authUuid 参数。对于非多平台接入则忽略此参数
+streamMode	int	主子码流标识	否	0-主码流 1-子码流，如未指定则使用默认值 0，如指定非可选值，返回错误
+transMode	int	传输协议	否	0-UDP 1-TCP，如未指定则使用默认值 1，如指定非可选值则返回错误。国标协议（GB/T28181）2011 版本只支持 UDP 传输协议，2016 版本开始支持 TCP 协议，使用国标协议接入设备的视频预览时，建议传输协议参数使用 UDP。直连萤石时该字段无效
+gpuMode	int	是否启用 GPU 硬解	否	0-不启用 1-启用，如未指定则使用默认值 0，如指定非可选值则返回错误，如无特殊需求建议不启用
+wndId	int	播放窗口序号	否	-1 或未指定-空闲窗口预览（不管成功与失败，预览后会自动跳到下一个空闲窗口） 0-选中窗口预览（预览后不会自动跳到下一个空闲窗口） 大于 0 为由 wndId 指定的窗口上播放（最大窗口数由 <a href="#">获取当前布局</a> 查询得到，预览后不会跳到下一个空闲窗口），若超出当前布局的窗口序号，则返回失败
+ezvizDirect	int	是否直连萤石预览	否	未指定或为 0-非直连 其它值-直连。多平台接入时不支持直连萤石，如指定直连，则返回失败
+cascade	int	是否为级联监控点	否	未指定或为 0-非级联设备 其它值-级联设备。

■ 通用请求入参举例

```
{
    funcName: "startPreview",
```

```
argument: "{
    cameraIndexCode: "afsgnhmj34567dgh",    // 监控点编号
    streamMode: 0,                          // 主子码流标识, 0-主码流 1-子码流
    transMode: 1,                          // 传输协议, 0-UDP 1-TCP
    gpuMode: 0                             // 是否开启 GPU 硬解, 不建议开启, 0-不开启 1-开启
}"
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
    code: 0,                                // 错误码 0-成功 1-失败
    msg: "missing field of cameraIndexCode" // 错误描述, 仅当 errorCode 非 0 时才有错误描述
}
```

### 3.12.5根据监控点编号录像回放

■ 功能说明

查询录像并播放。

■ 使用条件

初始化为回放模式后即可进行录像回放。注意。萤石回放（包括直连，非直连方式）不支持倍速、录像下载、录像锁定等功能。

■ 功能标识：startPlayback

■ 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“startPlayback”
argument	string	功能参数	是	
+cameraIndexCode	string	监控点编号	是	
+authUuid	string	<a href="#">设置接口认证信息参数</a> 接口中设置 authUuid	否	多平台接入时需要提供此字段用于区分具体的平台，来源于 <a href="#">设置接口认证信息参数</a> 接口设置的 authUuid 参数。对于非多平台接入则忽略此参数
+startTimeStamp	string	回放开始时间戳	是	单位：秒，开始播放时间，开始时间必须小于结束时间
+endTimeStamp	string	回放结束时间戳	是	单位：秒，播放结束时间，开始时间必须小于结束时间
+playTimeStamp	string	播放时间戳	否	单位：秒，如果指定了播放时间戳，则从此时间开始播放，否则从开始播放时间播放，如果大于或等于 endTimeStamp，则失败
+recordLocation	int	录像存储位置	否	0-中心存储 1-设备存储，如未指定则

				使用默认值 0，如指定了非可选值则返回错误。对于直连萤石回放，此参数无效
+transMode	int	传输协议	否	0-UDP 1- TCP，如未指定则使用默认值 1，如指定了非可选值则返回错误。 ehome 协议接入的设备的录像取流只支持 TCP 传输协议，传输协议参数请使用 TCP。直连萤石时该字段无效
+gpuMode	int	是否启用 GPU 硬解	否	0-不启用 1-启用，如未指定则使用默认值 0，如指定非可选值则返回错误，如无特殊需求建议不启用
+wndId	int	播放窗口序号	否	-1 或未指定-选中窗口回放（回放后不会跳到下一个空闲窗口） 0-空闲窗口回放（不管成功与失败，回放后自动跳到下一个空闲窗口） 大于 0 为由 wndId 指定的窗口上播放（最大窗口数由 <a href="#">获取当前布局</a> 查询得到，回放后不会跳到下一个空闲窗口），若超出当前布局的窗口序号，则返回失败
+ezvizDirect	int	是否直连萤石回放	否	未指定或为 0-非直连 其它值-直连。多平台接入时不支持指定直连萤石回放，如指定直连，则返回失败
+cascade	int	是否为级联监控点	否	未指定或为 0-非级联设备 其它值-级联设备。

■ 通用请求入参举例

```
{
  funcName: "startPlayback",
  argument: "{
    cameraIndexCode: "afsgnhmj34567dgh",    // 监控点编号
    startTimeStamp: "10237898985",           // 录像查询开始时间戳，单位：秒
    endTimeStamp: "10237899985",             // 录像查询结束时间戳，单位：秒
    recordLocation: 0,                       // 录像存储类型 0-中心存储 1-设备存储
    transMode: 1,                            // 传输协议，0-UDP 1-TCP
    gpuMode: 0                               // 是否开启 GPU 硬解，0-不开启 1-开启
  }"
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
    code: 0          // 错误码 0-成功 1-失败
}
```

### 3.12.6 停止所有视频预览

- 功能说明  
停止所有实时视频。
- 使用条件  
初始化为预览模式后即可停止所有视频预览。
- 功能标识: stopAllPreview
- 输入参数  
无
- 入参举例

funcName	string	功能标识	是	只能填写“stopAllPreview”
----------	--------	------	---	----------------------

- Js\_RequestInterface 通用请求入参举例

```
{
    funcName: "stopAllPreview"
}
```

- 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

- 返回值举例

```
{
    code: 0          // 错误码
}
```

### 3.12.7 停止所有录像回放

- 功能说明  
停止所有录像回放。
- 使用条件  
初始化为回放模式后即可停止所有录像回放。
- 功能标识: stopAllPlayback
- 输入参数  
无
- 入参举例

funcName	string	功能标识	是	只能填写“stopAllPlayback”
----------	--------	------	---	-----------------------

- Js\_RequestInterface 通用请求入参举例

```
{
    funcName: "stopAllPlayback"
}
```



■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
    code: 0          // 错误码
}
```

3.12.8销毁播放实例

■ 功能说明

销毁播放实例（**不建议使用**，使用 JS\_Disconnect 断开与插件服务的连接时，插件内部会销毁实例）。

■ 使用条件

在关闭网页时需要销毁播放实例，从 V1.1.0 开始，关闭网页时可不需销毁，视频 web 插件内部会自动销毁实例。

■ 功能标识：destroyWnd

■ 输入参数

无

■ 入参举例

funcName	string	功能标识	是	只能填写“destroyWnd”
----------	--------	------	---	------------------

■ Js\_RequestInterface 通用请求入参举例

```
{
    funcName: "destroyWnd"
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
    code: 0          // 错误码
}
```

3.12.9获取当前布局

■ 功能说明

获取当前布局。

■ 使用条件

初始化后即可获取布局。

■ 功能标识：getLayout

■ 输入参数

funcName	string	功能标识	是	只能填写“getLayout”
----------	--------	------	---	-----------------

■ 通用请求入参举例

```
{
    funcName: "getLayout"
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	
data	string	json 报文字符串	否	当 code 为 0 时该值才有意义

■ 返回值举例

```
{
    code: 0,           // 错误码
    data: "{
        layout: "2x2", // 布局样式，支持的布局详见初始化功能标识中 layout 字段
        wndNum: 4      // 当前布局的窗口数
    }"
}
```

### 3.12.10 设置当前布局

■ 功能说明

获取当前布局。

■ 使用条件

初始化后即可设置布局。

■ 功能标识：setLayout

■ 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“setLayout”
argument	string	功能参数	是	
+layout	string	布局参数	否	如未指定或指定值无效，使用默认值“2x2”布局，布局可选值有“1x1”、“2x2”、“3x3”、“4x4”、“5x5”、“1x2”、“1+2”、“1+5”、“1+7”、“1+8”、“1+9”、“1+12”、“1+16”、“4+9”、“1+1+12”、“3+4”、“1x4”、“4x6”

■ 通用请求入参举例

```
{
    funcName: " setLayout",
    argument: "{
        layout: "2x2" // 窗口布局
    }"
}
```

## ■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

## ■ 返回值举例

```
{
    code: 0          // 错误码
}
```

## 3.12.11 播放抓图

## ■ 功能说明

预览或回放抓图，图片存储在初始化中指定的文件夹（未指定时使用内置默认文件夹）。抓图后统一弹框提示。

## ■ 使用条件

预览、回放成功后。

## ■ 功能标识：snapShot

## ■ 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“snapShot”
argument	string	功能参数	是	object 格式的字符串
+name	string	图片绝对文件路径名称，如 “D:\test.jpg”	否	如未指定或无效，使用初始化中的抓图文件路径和抓图命名规则，详见 <a href="#">初始化</a> 。后缀支持 jpg 和 bmp，如指定其它后缀，则返回错误。仅包含正确的保存文件路径，不包含图片名称时，使用该文件路径保存图片，但图片名称使用默认命名规则。
+wndId	int	播放窗口序号（有效值为 0~ 获取当前布局中返回的窗口数）	否	0 或未指定-选中窗口抓图 1~获取当前布局中返回的窗口数-由 wndId 指定的窗口抓图（如返回的窗口数为 4，则 wndId 填充 1~4 表示在窗口序号从 1 开始的由 wndId 指定的窗口中抓图）

## ■ 通用请求入参举例

```
{
    funcName: "snapShot",
    argument: "{
        name: "D:\test.jpg"    // 窗口布局
    }"
}
```

## ■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败

msg	string	错误描述	否	
-----	--------	------	---	--

#### ■ 返回值举例

```
{
    code: 0          // 错误码 0-成功 1-失败
}
```

### 3.12.12 画面字符叠加

#### ■ 功能说明

预览回放画面叠加字符串。

#### ■ 使用条件

预览、回放成功后。

#### ■ 功能标识: drawOSD

#### ■ 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“drawOSD”
argument	string	功能参数	是	object 格式的字符串
+text	string	待叠加字符	是	支持“\n”换行，不超过 512 个字符
+x	int	相对播放窗口左上角的横坐标起点	是	
+y	int	相对播放窗口左上角的纵坐标起点	是	
+color	long	字体颜色	否	不指定默认为白色
+wndId	int	窗口 id	否	0 或未指定-选中窗口字符叠加 大于 0 为 wndId 指定的窗口字符叠加
+bold	int	是否加粗	否	不指定或为 0-不加粗 1-加粗，其它值-失败，默认不加粗
+alignType	int	多行字符串时的对齐方式	否	0 或未指定-左对齐 1-居中对齐 2-居右对齐，其它值-失败
+fontSize	int	字号	否	不小于 0 的字号，如果小于等于 0 或不指定则表示使用默认值 12。最大值不超过 100，超过 100 按 100 计算

#### ■ 通用请求入参举例

```
{
    funcName: "drawOSD",
    argument: "{
        text: "温度：50\n湿度：38",    // 窗口布局
        x: 5,
        y: 5
    }"
```

#### ■ 返回值

名称	类型	描述	是否必填	备注
----	----	----	------	----

code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
    code: 0          // 错误码 0-成功 1-失败
}
```

### 3.12.13 根据监控点编号批量视频预览

■ 功能说明

根据指定的监控点在指定的窗口播放。

■ 使用条件

初始化为预览模式。

■ 功能标识: startMultiPreviewByCameraIndexCode

■ 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“startMultiPreviewByCameraIndexCode”
argument	string	功能参数	是	object 格式的字符串
+list	object[]	预览参数集合	是	
++cameraIndexCode	string	监控点编号	是	
++authUuid	string	<a href="#">设置接口认证信息参数</a> 接口中设置 authUuid	否	多平台接入时需要提供此字段用于区分具体的平台，来源于 <a href="#">设置接口认证信息参数</a> 接口设置的 authUuid 参数。对于非多平台接入则忽略此参数
++streamMode	int	主子码流标识	否	0-主码流 1-子码流，如未指定则使用默认值 0，如指定非可选值，返回错误
++transMode	int	传输协议	否	0-UDP 1-TCP，如未指定则使用默认值 1，如指定非可选值则返回错误。国标协议（GB/T28181）2011 版本只支持 UDP 传输协议，2016 版本开始支持 TCP 协议，使用国标协议接入设备的视频预览时，建议传输协议参数使用 UDP。直连萤石时该字段无效
++gpuMode	int	是否启用 GPU 硬解	否	0-不启用 1-启用，如未指定则使用默认值 0，如指定非可选值则返回错误，如无特殊需求建议不启用
++wndId	int	播放窗口序号	是	取值范围 1~最大窗口数（最大窗口数由 <a href="#">获取当前布局</a> 查询得到，预览后不会跳到一个空闲窗口），由 wndId 指定的窗口上播放，若超出当前布局的窗口序号，则返回失败
++ezvizDirect	int	是否直连萤石预览	否	未指定或为 0-非直连 其它值-直连。多平

				台接入时不支持直连萤石，如指定直连，则返回失败
--	--	--	--	-------------------------

以上为单个请求的参数，示例详见入参举例。

■ 通用请求入参举例

```
{
  funcName: "startMultiPreviewByCameraIndexCode",
  argument: {
    list: [
      {
        cameraIndexCode: "c633ef048fe141e1ac6dbeb36aaf21d3",
        ezvizDirect: 0,
        gpuMode: 0,
        streamMode: 0,
        transMode: 1,
        wndId: 1
      },
      {
        cameraIndexCode: "c633ef048fe141e1ac6dbeb36aaf21d3",
        ezvizDirect: 0,
        gpuMode: 0,
        streamMode: 0,
        transMode: 1,
        wndId: 2
      }
    ]
  }
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
  code: 0,          // 错误码
}
```

3.12.14 根据监控点编号批量录像回放

■ 功能说明

根据指定的监控点编号在指定的窗口上回放。

■ 使用条件

初始化为回放模式。

■ 功能标识：startMultiPlaybackByCameraIndexCode

## ■ 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“startMultiPlaybackByCameraIndexCode”
argument	string	功能参数	是	object 格式的字符串
+list	object[]	监控点编号录像回放参数集合	是	
++cameraIndexCode	string	监控点编号	是	
++authUuid	string	<a href="#">设置接口认证信息参数</a> 接口中设置 authUuid	否	多平台接入时需要提供此字段用于区分具体的平台，来源于 <a href="#">设置接口认证信息参数</a> 接口设置的 authUuid 参数。对于非多平台接入则忽略此参数
++startTimeStamp	string	回放开始时间戳	是	单位：秒，开始播放时间，开始时间必须小于结束时间
++endTimeStamp	string	回放结束时间戳	是	单位：秒，播放结束时间，开始时间必须小于结束时间
++playTimeStamp	string	播放时间戳	否	单位：秒，如果指定了播放时间戳，则从此时间开始播放，否则从开始播放时间播放
++recordLocation	int	录像存储位置	否	0-中心存储 1-设备存储，如未指定则使用默认值 0，如指定了非可选值则返回错误
++transMode	int	传输协议	否	0-UDP 1- TCP，如未指定则使用默认值 1，如指定了非可选值则返回错误。ehome 协议接入的设备的录像取流只支持 TCP 传输协议，传输协议参数请使用 TCP。直连萤石时该字段无效
++gpuMode	int	是否启用 GPU 硬解	否	0-不启用 1-启用，如未指定则使用默认值 0，如指定非可选值则返回错误，如无特殊需求建议不启用
++wndId	int	播放窗口序号	是	取值范围 1~最大窗口数（最大窗口数由 <a href="#">获取当前布局</a> 查询得到，回放后不会跳到下一个空闲窗口），由 wndId 指定的窗口上播放，若超出当前布局的窗口序号，则返回失败
++ezvizDirect	int	是否直连萤石回放	否	未指定或为 0-非直连 其它值-直连。多平台接入时不支持直连萤石，如指定直连，则返回失败

以上为单个请求的参数，示例详见入参举例。

## ■ 通用请求入参举例

```
{
  funcName: "startMultiPlaybackByCameraIndexCode",
  argument:
  {
```

```
list:[{
  cameraIndexCode: "58e90452772a4d9da7c7ba4cef26dbf0",    // 监控点编号
  startTimeStamp: "10237898985",        // 录像查询开始时间戳，单位：秒
  endTimeStamp: "10237899985",          // 录像查询结束时间戳，单位：秒
  recordLocation: 0,    // 录像存储类型 0-中心存储 1-设备存储
  transMode: 1,         // 传输协议，0-UDP 1-TCP
  wndId: 1,             // 窗口序号
  gpuMode: 0           // 是否开启 GPU 硬解，0-不开启 1-开启
}]
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
  code: 0          // 错误码 0-成功
}
```

### 3.12.15 批量停止播放

■ 功能说明

对指定窗口停止播放。

■ 使用条件

预览回放后即可使用。

■ 功能标识：stopMultiPlay

■ 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“stopMultiPlay”
argument	string	功能参数	是	object 格式的字符串
+list	object[]	窗口序号集合	是	
++wndId	int	播放窗口序号	是	取值范围 1~最大窗口数（最大窗口数由 <a href="#">获取当前布局</a> 查询得到），由 wndId 指定的窗口上停止播放，若超出当前布局的窗口序号，则返回失败

以上为单个请求的参数，示例详见入参举例。

■ 通用请求入参举例

```
{
  funcName: "stopMultiPlay",
  argument:
  {
    list:[
```



```
        wndId: 1                // 窗口序号
    },
    {
        wndId: 2                // 窗口序号
    }
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
    code: 0,        // 错误码
}
```

### 3.12.16 设置接口认证信息参数

■ 功能说明

设置平台 APPKey、APPSecret 等信息。用于同时对接多个平台的场景，此时初始化中不能指定 appkey、secret、ip、port 这些信息。

■ 使用条件

初始化后使用。注：若已设置认证信息成功，在反初始化前，再次设置认证信息，且使用已有的 authUuid，则会失败报错。

■ 功能标识：setAuthInfo

■ 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写 “setAuthInfo”
argument	string		是	object 格式的字符串
+list	object[]	认证参数集合	是	
++appkey	string	平台提供的 APPKey	是	如何获取 APPKey 详见 <a href="#">如何获取 APPKey 和 APPSecret</a>
++secret	string	平台提供的 APPSecret	是	如何获取 APPKey 详见 <a href="#">如何获取 APPKey 和 APPSecret</a>
++ip	string	平台（Nginx）IP 地址	是	
++port	int	平台端口	是	HTTPS 协议的端口号，一般情况下是 443，需要根据具体环境来指定
++authUuid	string	接口认证 UUID，标识 appkey、secret、ip 与 port	是	使用方自行生成，保证每组 appkey、secret、ip、port 的 UUID 不一样即可。此 UUID 用于预览回放标识一组认证参数
++encryptedFields	string	加密字段	否	填充需要加密的字段，多个字段间以 “,” 分割，当前仅支持对 appkey、secret、

				ip 加密，其它字段暂不支持加密，注意加密使用同一个公钥分别对各个字段的值加密；当该字段中包含非法字段时，不报错，内部也不进行处理
--	--	--	--	---

■ 通用请求入参举例

```
{
  funcName: "setAuthInfo",
  argument: {
    list: [
      {
        appkey: "2SDKMLIS4DG567MNF",
        secret: "DFHGRDL24567UISN",
        ip: "10.10.10.10",
        port: 443,
        authUuid: "1",
        encryptedFields: "appkey,secret"
      },
      {
        appkey: "2SDKMLISSDGD4DG567MNF",
        secret: "DFHSFGGHGGRDL24567UISN",
        ip: "10.10.10.110",
        port: 443,
        authUuid: "2",
        encryptedFields: "appkey,secret"
      }
    ]
  }
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
  code: 0,           // 错误码
}
```

3.12.17 进入全屏

■ 功能说明

插件进入全屏。

■ 使用条件

已初始化。

■ 功能标识: setFullScreen

■ 输入参数

funcName	string	功能标识	是	只能填写“setFullScreen”
----------	--------	------	---	---------------------

■ 通用请求入参举例

```
{
    funcName: "setFullScreen"
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
    code: 0           // 错误码
}
```

### 3.12.18 退出全屏

■ 功能说明

插件进入全屏后，退出全屏。

■ 使用条件

已初始化，且已进入全屏状态。

■ 功能标识: exitFullScreen

■ 输入参数

funcName	string	功能标识	是	只能填写“exitFullScreen”
----------	--------	------	---	----------------------

■ 通用请求入参举例

```
{
    funcName: "exitFullScreen "
}
```

■ 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
    code: 0           // 错误码
}
```

### 3.12.19 获取版本号

■ 功能说明

获取插件版本号。

■ 使用条件

已创建插件实例（JS\_CreateWnd 成功）。

- 功能标识：getVersion
- 输入参数

funcName	string	功能标识	是	只能填写“getVersion”
----------	--------	------	---	------------------

- 通用请求入参举例

```
{  
    funcName: "getVersion"  
}
```

- 返回值

名称	类型	描述	是否必填	备注
code	int	错误码	是	0-成功，始终为 0
msg	string	错误描述	否	
data	string	版本号	是	版本号命名规则：Vx.y.z，如 V1.5.0

- 返回值举例

```
{  
    code: 0,           // 错误码  
    data: "V1.5.0"    // 版本号  
}
```

### 3.12.20 设置时间轴级别

- 功能说明  
设置时间轴级别，支持跨级别修改。
- 使用条件  
初始化为回放模式。
- 功能标识：setTimeBarLevel
- 输入参数

名称	类型	描述	是否必填	备注
funcName	string	功能标识	是	只能填写“setTimeBarLevel”
argument	string	功能参数	是	object 格式的字符串
+ levelType	string	时间轴级别	否	如未指定或指定值为字符串但是参数无效，使用默认值 24h。该字段取值范围“24h, 12h, 6h, 1h, 36m, 24m, 12m”，超出该取值范围无效。

- 通用请求入参举例

```
{  
    "funcName": " setTimeBarLevel ",  
    "argument": {  
        levelType: "6"  
    }  
}
```

- 返回值

名称	类型	描述	是否必填	备注
----	----	----	------	----

code	int	错误码	是	0-成功 -1-失败
msg	string	错误描述	否	

■ 返回值举例

```
{
  "code": 0          // 错误码 0-成功
}
```

3.13 JS\_SetWindowControlCallback 设置消息回调接口

- JS\_SetWindowControlCallback 用于设置视频 web 插件消息回调，所有视频插件的消息都通过设置的回调通知前端，前端调用示例如下：

// 设置消息回调，oWebControl 是 jsWebControl-1.0.0.min.js 中 WebControl 的一个实例

```
oWebControl.JS_SetWindowControlCallback({
  cbIntegrationCallBack: function(oData){      // oData 是封装的视频 web 插件回调消息的消息体
    console.log(JSON.stringify(oData));        // 打印消息体至控制台
  }
});
```

- 回调的消息体为 json 报文，数据格式如下：

```
{
  uuid: "xxx-xxx-xxx-xxx",      // 消息体唯一标识
  sequence: "",                 // 序号
  cmd: "window.integrationCallBack", // 命令
  responseMsg: {
    // 此处为视频 web 插件消息 json 报文字符串
  }
}
```

其中 responseMsg 对应的 value 是视频 web 插件返回的 json 封装的消息，只需解析 responseMsg 即可。目前支持的视频 web 插件消息有窗口选中消息、预览或回放播放消息、抓图结果消息和预览紧急录像或回放录像剪辑结果消息。这四类消息遵循统一的消息格式，如下：

```
{
  type: 1,    // 消息类型，取值详见 3.12.\*
  msg: {
    {
      wndId: 1,    // 窗口序号，从 1 开始
      result: 0x0100,    // 0x0100-正在播放 0x0200-空闲
      cameraIndexcode: "58e90452772a4d9da7c7ba4cef26dbf0", // 监控点编号
      expand: ""    // 扩展字段
    }
  }
}
```

3.13.1 窗口选中消息

- type 取值：1

### ■ 消息参数

名称	类型	描述	是否必填	备注
wndId	int	窗口序号	是	窗口序号从 1 开始
result	int	窗口中的播放状态	是	0x0100(256)-正在播放 0x0200(512)-空闲。
cameraIndexCode	string	监控点编号	是	result 为空闲状态时 cameraIndexCode 为空字符串，否则表示监控点编号。
+authUuid	string	接口认证 UUID	否	多平台接入时若窗口非空闲则回调此字段用于区分具体的平台，来源于 <a href="#">设置接口认证信息参数</a> 接口设置的 authUuid 参数。对于非多平台接入或窗口空闲，则无此字段
expand	string	扩展参数	否	窗口选中消息无扩展参数

### ■ 消息回调中 pRspJsonMsg 举例

```
{
  type: 1,    // 选中窗口消息
  msg:
  {
    wndId: 1,    // 窗口序号
    result: 0x0100,    // 0x0100-正在播放 0x0200-空闲
    cameraIndexCode: "58e90452772a4d9da7c7ba4cef26dbf0" // 监控点编号
  }
}
```

### ■ 备注

窗口接收到鼠标按下消息就认为是窗口选中消息，有以下几种情况会触发该事件：

- 1) 鼠标左键按下（鼠标左键点击播放窗口以及点击非选中播放窗口上、下工具条按钮都会触发该事件）
- 2) 鼠标右键按下
- 3) 鼠标左键双击

## 3.13.2 预览/回放播放消息

### ■ type 取值：2

### ■ 消息参数

名称	类型	描述	是否必填	备注
wndId	int	窗口序号	是	窗口序号从 1 开始
result	int	窗口中的播放状态	是	0x0300(768)-开始播放 0x0301(769)-播放失败（指发起请求失败） 0x0302(770)-播放异常（指取流或解码过程中的异常），0x0330(816)-播放结束
cameraIndexCode	string	监控点编号	是	四种状态下返回监控点编号
+authUuid	string	接口认证 UUID	否	多平台接入时回调此字段用于区分具体的平台，来源于 <a href="#">设置接口认证信息参数</a> 接口设置的 authUuid 参数。对于非多

				平台接入，则无此字段
expand	string	扩展参数	否	预览/回放播放消息无扩展参数

■ 消息回调中 pRspJsonMsg 举例

```
{
  type: 2, 播放消息
  msg:
  {
    wndId: 1,      // 窗口序号
    result: 0x0300, // 0x0300-开始播放 0x0301-播放失败 0x0302-播放异常, 0x0330-播放结束
    cameraIndexCode: "58e90452772a4d9da7c7ba4cef26dbf0" // 监控点编号
  }
}
```

■ 备注

无

### 3.13.3抓图结果消息

■ type 取值：3

■ 消息参数

名称	类型	描述	是否必填	备注
wndId	int	窗口序号	是	窗口序号从 1 开始
result	int	窗口中的抓图状态	是	0-成功 -1-失败。
cameraIndexCode	string	监控点编号	是	不管成功失败，都会返回监控点编号
+authUuid	string	接口认证 UUID	否	多平台接入时回调此字段用于区分具体的平台，来源于 <a href="#">设置接口认证信息参数</a> 接口设置的 authUuid 参数。对于非多平台接入，则无此字段
picData	string	base64 编码后的图片数据	否	当抓 jpg 图时通过此字段给出图片 base64 编码数据
operType	int	抓图触发方式	是	0-插件 UI 按钮触发 1-接口触发
expand	string	扩展参数	否	result 取值为 0 时，为图片绝对文件路径名称

■ 消息回调中 pRspJsonMsg 举例

```
{
  type: 3, // 抓图结果消息
  msg:
  {
    wndId: 1, // 窗口序号
    result: 0, // 0-成功 -1-失败
    operType: 0, // 抓图触发方式
    picData: "DSDJKRKLKSJFKKMLKHKMEFGKMG", // base64 后的 jpg 图片数据
    cameraIndexCode: "58e90452772a4d9da7c7ba4cef26dbf0", // 监控点编号
    expand: "D:\\test1.jpg" // 图片文件
  }
}
```

```
    }  
  }  
  ■ 备注  
    无
```

### 3.13.4预览紧急录像/回放录像剪辑结果消息

- type 取值: 4
- 消息参数

名称	类型	描述	是否必填	备注
wndId	int	窗口序号	是	窗口序号从 1 开始
result	int	窗口中的录像结果	是	0-成功 -1-失败。
cameraIndexCode	string	监控点编号	是	不管成功失败都返回监控点编号
+authUuid	string	接口认证 UUID	否	多平台接入时回调此字段用于区分具体的平台, 来源于 <a href="#">设置接口认证信息参数</a> 接口设置的 authUuid 参数。对于非多平台接入, 则无此字段
expand	string	扩展参数	否	result 取值为 0 时, 为录像存储绝对文件路径名称, 如有多个则以“;”分割

- 消息回调中 pRspJsonMsg 举例

```
{  
    type: 4,                                // 录像结果消息  
    msg:  
    {  
        wndId: 1,                            // 窗口序号  
        result: 0,                            // 0-成功 -1-失败  
        cameraIndexCode: "58e90452772a4d9da7c7ba4cef26dbf0", // 监控点编号  
        expand: "D:\\video.mp4,D:\\video-1.mp4"           // 录像绝对文件路径名称  
    }  
}
```

- 备注
- 无

### 3.13.5进入全屏/退出全屏消息

- type 取值: 5
- 消息参数

名称	类型	描述	是否必填	备注
result	int	窗口中的录像结果	是	0x0400(1024)-进入全屏 0x0401(1025)-退出全屏。

- 消息回调中 pRspJsonMsg 举例

```
{  
    type: 5,                                // 录像结果消息
```



```
msg:
{
    result: 0x400                                // 0x0400-全屏 0x0401-退出全屏
}
}
```

■ 备注  
无

### 3.13.6切换布局消息

- type 取值：6
- 消息参数

名称	类型	描述	是否必填	备注
type	int	消息类型	是	消息类型
msg	object	具体消息信息	是	
+layout	string	切换后的布局名称	是	
+wndNum	int	layout 布局下窗口数	是	

- 消息回调中 pRspJsonMsg 举例

```
{
    "type": 6,                                // 录像结果消息
    "msg":
    {
        "layout": "2x2",                    // 布局名称
        "wndNum": 4
    }
}
```

- 备注

通过监听此消息，前端可保持当前布局信息，在批量预览、回放时根据布局信息中的窗口数限定预览回放的窗口序号。

### 3.13.7播放窗口双击事件消息

- type 取值：7
- 消息参数

名称	类型	描述	是否必填	备注
type	int	消息类型	是	消息类型
msg	object	具体消息信息	是	
+wndId	int	窗口序号	是	窗口序号从 1 开始
+wndAction	int	窗口大小变化状态	是	0-还原 1-放大
+result	int	窗口中的播放状态	是	0x0100(256)-正在播放 0x0200(512)-空闲
+cameraIndexCode	string	监控点编号	否	result 为空闲状态时 cameraIndexCode 为空字符串，否则表示监控点编号。

+authUuid	string	接口认证 UUID	否	多平台接入时若窗口非空闲则回调此字段用于区分具体的平台，来源于 <a href="#">设置接口认证信息参数</a> 接口设置的 authUuid 参数。对于非多平台接入或窗口空闲，则无此字段
-----------	--------	-----------	---	--

■ 消息回调中 pRspJsonMsg 举例

```
{
  "type": 7,                                     // 播放窗口双击事件消息
  "msg": {
    " wndId ": 2,           // 窗口 id
    " wndAction ": 1,
    " result ": 512
  }
}
```

■ 备注

通过监听此消息，前端可获取双击窗口时，窗口大小发生变化的状态信息，通过此消息用户可以通过 [3.12.17](#) 和 [3.12.18](#) 中的接口实现双击窗口进入/退出全屏的功能。

3.13.8时间轴级别变化消息

■ type 取值：8

■ 消息参数

名称	类型	描述	是否必填	备注
type	int	消息类型	是	消息类型
msg	object	具体消息信息	是	
+timeBarLevel	String	时间轴级别	是	取值范围 “24h, 12h, 6h, 1h, 36m, 24m, 12m”

■ 消息回调中 pRspJsonMsg 举例

```
{
  "type": 8,                                     // 播放窗口双击事件消息
  "msg": {
    " timeBarLevel ": "6h"
  }
}
```

■ 备注

当时间轴级别发生变化时，回调该消息，该消息仅在回放模式下有效；用户通过监听该消息可实现自己的业务逻辑。

4. 视频 WEB 插件对接指南

4.1 开发流程

- 安装 VideoWebPlugin.exe 插件包（请直接安装，请勿自行解压并运行解压后的可执行文件，安装于系统盘的“Program Files (x86)\VideoWebPlugin”目录中）
  - 视频 WEB 应用开发
- 对于 WEB 开发，开发流程图如图 4.1-1。

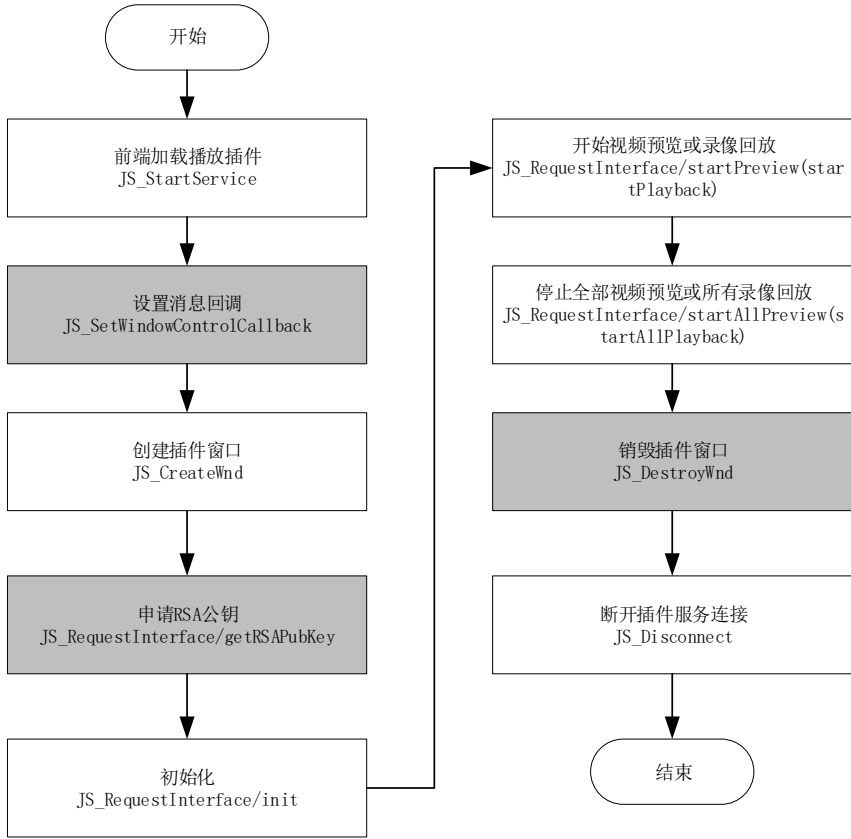


图 4.1-1 VideoWebPlugin 基本对接流程图

图 3.1-1 中灰色部分为可选步骤。其中申请 RSA 公钥可选。申请 RSA 公钥是为了加密初始化中的一些敏感参数，对安全性要求高的用户可以考虑对敏感参数加密。但需要注意的是，使用 RSA 公钥加密机制必然会导致初始化耗时。一般情况下不需要调 JS\_DestroyWnd 来销毁插件窗口（JS\_Disconnect 中会自行销毁），但一些特殊的场景如浏览器页面上需随时启用和禁用视频播放时，需调 JS\_DestroyWnd 来禁用视频播放，需要调 JS\_CreateWnd 来启用视频播放。JS\_DestroyWnd 和 JS\_Disconnect 中会反初始化插件，这里无需调 JS\_RequestInterface/uninit 反初始化。

## 4.2 网页开发指南

以视频预览为例（详细实例代码详见 demo\_window\_simple\_preview.html）。

### 4.2.1 引用 JS 文件

```
<script src="jquery-1.12.4.min.js"></script>
<script src="jsencrypt.min.js"></script> // 用于 RSA 公钥加密
<script src="jsWebControl-1.0.0.min.js"></script> // 用于前端与插件交互
```

## 4.2.2 创建 WebControl 实例

要使用视频 WEB 插件，首先需要创建 WebControl 实例，示例代码如下：

```
var oWebControl = new WebControl({ // 创建 WebControl 实例
    szPluginContainer: "playWnd", // 指定 DIV 窗口标识
    iServicePortStart: 15900, // 指定起止端口号，建议使用该值
    iServicePortEnd: 15909,

    // 用于 IE10 使用 ActiveX 的 clsid
    szClassId: "23BF3B0A-2C56-4D97-9C03-0CB103AA8F11",
    cbConnectSuccess: function () {
        // 创建 WebControl 实例成功
    },
    cbConnectError: function () {
        // 创建 WebControl 实例失败
    },
    cbConnectClose: function (bNormalClose) {
        // 插件使用过程中发生的断开与插件服务连接的回调
        // bNormalClose = false 时表示异常断开
        // bNormalClose = true 时表示正常断开
    }
});
```

## 4.2.3 启动插件

创建 WebControl 实例后即可在其创建成功回调中启动服务，在启动服务成功回调中设置消息回调并创建插件窗口。在创建 WebControl 实例失败回调中可尝试唤醒 WebControl.exe，也可以再次尝试创建 WebControl 实例。示例代码如下：

```
oWebControl = new WebControl({ // 创建 WebControl 实例
    szPluginContainer: "playWnd", // 指定 DIV 窗口标识
    iServicePortStart: 15900, // 指定起止端口号，建议使用该值
    iServicePortEnd: 15909,

    // 用于 IE10 使用 ActiveX 的 clsid
    szClassId: "23BF3B0A-2C56-4D97-9C03-0CB103AA8F11",
    cbConnectSuccess: function () { // 创建 WebControl 实例成功

        // 实例创建成功后启动服务
        oWebControl.JS_StartService("window", {
            dllPath: "./VideoPluginConnect.dll"//
        }).then(function () { // 启动插件服务成功
```

```

        // 设置消息回调
        oWebControl.JS_SetWindowControlCallback({
            cbIntegrationCallBack: cbIntegrationCallBack
        });

        //JS_CreateWnd 创建视频播放窗口，宽高可设定
        oWebControl.JS_CreateWnd("playWnd", 1000, 600).then(function () {
            init(); // 创建播放实例成功后初始化
        });
    }, function () { // 启动插件服务失败
    });
},
cbConnectError: function () { // 创建 WebControl 实例失败
    oWebControl = null;
    $("#playWnd").html("插件未启动，正在尝试启动，请稍候...");

    // 程序未启动时执行 error 函数，采用 wakeup 来启动程序
    WebControl.JS_WakeUp("VideoWebPlugin://");
    initCount++;
    if (initCount < 3) {
        setTimeout(function () {
            initPlugin();
        }, 3000)
    } else {
        $("#playWnd").html("插件启动失败，请检查插件是否安装!");
    }
},
cbConnectClose: function (bNormalClose) {
    // 异常断开: bNormalClose = false
    // JS_Disconnect 正常断开: bNormalClose = true
    console.log("cbConnectClose");
    oWebControl = null;
}
});

```

其中 iServicePortStart 和 iServicePortEnd 指定起止端口，建议使用该值。如果 VideoWebPlugin.exe 安装后，若 WebControl.exe 未启动或进程被结束，创建 WebControl 实例时 cbConnectError 会被执行，在此可以使用 WebControl.JS\_WakeUp("VideoWebPlugin://")来启动程序。上述代码片段中，如果创建 WebControl 实例失败，则尝试 3 次。initPlugin 函数封装了上述整个代码片段。

#### 4.2.4 监听事件

浏览器滚动条滚动或尺寸变化时，插件窗口需要跟着变化，需要监听滚动条滚动事件和浏览器

的 `resize` 事件，在事件中重新绘制插件窗口，并设置插件窗口超出浏览器部分显/隐。代码示例如下（代码详见 demo）：

```
$(window).resize(callback) { // 窗口 resize
    if (oWebControl != null) {
        oWebControl.JS_Resize(1000, 600); // 重绘插件窗口
        setWndCover(); // 设置超出浏览器的部分插件窗口显/隐，具体代码详见 demo
    }
});

$(window).scroll(callback) { // 滚动条滚动事件
    if (oWebControl != null) {
        oWebControl.JS_Resize(1000, 600); // 重绘插件窗口
        setWndCover(); // 设置超出浏览器的部分插件窗口显隐，具体代码详见 demo
    }
});
```

#### 4.2.5 申请 RSA 公钥

对敏感信息如 `secret` 需加密，插件提供 RSA 公钥加密机制保证敏感信息安全性，代码片段如下：

```
function getPubKey (callback) { // 申请 RSA 公钥
    oWebControl.JS_RequestInterface({
        funcName: "getRSAPubKey",
        argument: JSON.stringify({
            keyLength: 1024 // 指定秘钥长度为 1024bit
        })
    }).then(function (oData) {
        if (oData.responseMsg.data) {
            pubKey = oData.responseMsg.data; // 记录 RSA 公钥
            callback()
        }
    });
}

function setEncrypt (value) { //RSA 公钥加密
    var encrypt = new JSEncrypt();
    encrypt.setPublicKey(pubKey);
    return encrypt.encrypt(value);
}
```

#### 4.2.6 插件初始化

要显示视频预览或回放的窗口，需要对插件进行初始化，示例代码片段如下：

```
var appkey = "21349144"; //平台提供的 appkey，必填
```

版权所有©杭州海康威视数字技术股份有限公司 2020

```
// "DuyWn2H7Qb7jB9rPArth"为平台提供的 secret，必填
var secret = setEncrypt("DuyWn2H7Qb7jB9rPArth"); // 对 secret 进行 RSA 公钥加密
var ip = "10.19.132.75"; //平台 IP 地址，必填
var playMode = 0; // 初始播放模式：0-预览，1-回放，必填
var port = 443; //平台端口，若启用 HTTPS 协议，默认 443，必填
var snapDir = "D:\\SnapDir"; // 抓图存储文件路径，选填
var videoDir = "D:\\VideoDir"; // 紧急录像或录像剪辑存储文件路径，选填
var layout = "1x1"; // playMode 指定模式的布局，选填
var enableHTTPS = 1; //是否启用 HTTPS 协议与平台交互，是为 1，否为 0，选填
var encryptedFields = 'secret'; //加密字段，形如'字段 1,...,字段 n'，选填

oWebControl.JS_RequestInterface({
    funcName: "init",
    argument: JSON.stringify({
        appkey: appkey,
        secret: secret,
        ip: ip,
        playMode: playMode,
        port: port,
        snapDir: snapDir,
        videoDir: videoDir,
        layout: layout,
        enableHTTPS: enableHTTPS,
        encryptedFields: encryptedFields
    })
});
```

#### 4.2.7 视频预览业务功能

视频预览业务功能包括视频预览，停止全部预览、抓图等功能。以视频预览为例，示例如下：

```
var cameraIndexCode = $("#cameraIndexCode").val(); // 监控点编号，json 中必填
var streamMode = 0; //主子码流标识：0-主码流，1-子码流，json 中选填
var transMode = 1; // 传输协议：0-UDP，1-TCP，json 中选填
var gpuMode = 0; // 是否启用 GPU 硬解，0-不启用，1-启用，json 中选填
var wndId = -1; // 播放窗口序号（在 2x2 以上布局下可指定播放窗口），json 中选填

cameraIndexCode = cameraIndexCode.replace(/(^s*)/g, "");
cameraIndexCode = cameraIndexCode.replace(/(s*$)/g, "");

oWebControl.JS_RequestInterface({
    funcName: "startPreview",
    argument: JSON.stringify({
```

```
        cameraIndexCode:cameraIndexCode,
        streamMode: streamMode,
        transMode: transMode,
        gpuMode: gpuMode,
        wndId:wndId
    })
});
```

### 4.2.8 断开插件服务连接

前端 unload 事件或路由离开页面时需断开与插件服务的连接，示例代码如下：

```
if (oWebControl != null){
    oWebControl.JS_HideWnd();// 先隐去窗口，规避插件窗口滞后于浏览器消失问题
    oWebControl.JS_Disconnect().then(function(){
        // 断开与插件服务连接成功
    },
    function(){
        // 断开与插件服务连接失败
    });
}
```

## 4.3 iframe 对接指南

目前 iframe 对接存在插件窗口无法及时更新位置导致用户体验问题，以及内嵌页面不支持滚动条问题，比较适合页面无滚动、页面大小不经常改变的场景。请结合业务考虑是否适合使用 iframe 对接。iframe 对接 demo 请参考 demo\_for\_iframe.html 与 demo\_embedded\_for\_iframe.html 文件。

### 4.3.1 iframe demo 使用

- 1、运行 demo 目录的 webs 服务程序；
- 2、浏览器中输入 [http://127.0.0.1/demo\\_for\\_iframe.html](http://127.0.0.1/demo_for_iframe.html)

### 4.3.2 iframe 对接步骤

对于父页面，对接步骤如下（请结合 demo\_for\_iframe.html，搜索“步骤”关键字）：

- 1、demo\_for\_iframe.html 中 iframe 标签指定待嵌入页面，并禁用滚动条。详情请参考 demo\_for\_iframe.html 中步骤 1。



- 2、子页面的 onload 事件中向嵌入的页面通过消息的形式设置一些子页面的初始值。详情请参考 demo\_for\_iframe.html 中步骤 2。
- 3、监听子页面的消息。详情请参考 demo\_for\_iframe.html 中步骤 3。
- 4、监听本页面的 resize 事件，在事件响应中更新子页面一些值，使子页面触发插件窗口位置更新。详情请参考 demo\_for\_iframe.html 中步骤 4。
- 5、监听本页面的 scroll 事件，在事件响应中更新子页面一些值，使子页面触发插件窗口位置更新。详情请参考 demo\_for\_iframe.html 中步骤 5。

对于子页面，对接步骤如下（请结合 demo\_embedded\_for\_iframe，搜索“步骤”关键字）：

- 1、监听父页面的消息，并对消息作出响应。详情请参考 demo\_embedded\_for\_iframe 中步骤 1。
- 2、创建插件窗口（JS\_CreateWnd）时指定 cbSetDocTitle 回调，并在回调中通知父页面更新其标题为回调给出的 uuid）。详情请参考 demo\_embedded\_for\_iframe 中步骤 2。
- 3、创建插件成功后通知父页面将其标题更新回去。详情请参考 demo\_embedded\_for\_iframe 中步骤 3。
- 4、创建插件成功后通知父页面更新滚动条偏移量（为兼容嵌入子页面时父页面滚动条已有偏移量情况下插件窗口与 DIV 窗口无法贴合问题，父页面收到此消息需要将滚动条偏移量通知给子页面，子页面监听到此消息后更新插件窗口位置）。详情请参考 demo\_embedded\_for\_iframe 中步骤 4。

## 5. 视频 WEB 插件 UI 集成控件列表

视频 WEB 插件 UI 集成的控件包括实时预览、录像回放上工具条和下工具条中的控件以及工具栏中的控件，各个控件的详细信息详见下表：

表 5-1 集成控件详细信息

控件名称	所属模块	备注
监控点名称展示控件	实时预览、录像回放上工具条	可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
监控点类型展示	实时预览、录像回	可通过通用请求响应接口的初始化功能来自定义该控件是否显示

控件	放上工具条	
关闭播放按钮	实时预览、录像回放上工具条	可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
声音控制按钮	实时预览、录像回放下工具条	通过该按钮对播放窗口播放的视频打开和关闭声音，需要设备上配置视频类型为复合流，如果配置的视频类型为视频流，则声音控制将失败。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
抓图按钮	实时预览、录像回放下工具条	通过该按钮对当前选中播放窗口播放的视频抓取一张图片。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
电子放大按钮	实时预览、录像回放下工具条	播放库层面对播放画面细节放大，只影响当前播放的视频。小鹰眼设备不支持。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
监控点信息显示按钮	实时预览、录像回放下工具条	通过该按钮来选择是否显示监控点名称、帧率、视频分辨率、码流、视频编码格式、流数据封装格式等信息。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a> 。 录像回放单帧步退播放时，设置监控点信息显/隐对单帧状态不起效，仅对退出单帧步退后的播放起效。
小鹰眼按钮	实时预览、录像回放下工具条	通过该按钮支持在选中播放窗口中切换 8 画面和 PTZ 碗状画面两种显示模式，小鹰眼设备只支持“海康 SDK 协议”接入，小鹰眼设备实时预览和录像回放不支持即时回放、云台控制、3D 放大、电子放大。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
云台控制按钮	实时预览下工具条	通过该按钮启动云台 8 个方向的转动，需要设备支持。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
3D 放大按钮	实时预览下工具条	设备层面对播放画面细节放大，所有实时预览该设备的视频都起效。3D 放大需要设备的支持。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
主子码流切换按钮	实时预览下工具条	通过该按钮对当前选中窗口切换主子码流预览，切换主子码流是重新播放。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
语音对讲按钮	实时预览下工具条	通过该按钮实现当前窗口的语音对讲功能。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
紧急录像按钮	实时预览下工具条	通过该按钮对预览的码流存文件，如果当前预览的是大华 SDK 协议接入设备实时视频，该文件需使用大华播放器播放，否则可使用 VLC 或海康播放器播放（建议使用海康播放器播放）。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
即时回放按钮	实时预览下工具条	通过该按钮对实时预览时最近的录像回放，回放长度为 5MB 录像。支持即时回放的设备接入协议有：海康 SDK 协议、EHOME 协议、ONVIF 协议，小鹰眼设备海康 SDK 协议接入时实时预览不支持即时回放。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
录像剪辑按钮	录像回放下工具条	通过该按钮对回放的码流存文件，如果当前回放的是大华 SDK 协议接入设备的录像，该文件需使用大华播放器播放，否则可使用 VLC 或海康播放器播放（建议使用海康播放器播放）。可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
录像下载按钮	录像回放下工具条	通过该按钮发起录像下载。可通过通用请求响应接口的初始化功能来自定

		义该控件是否显示，详见 <a href="#">初始化</a>
录像锁定按钮	录像回放下工具条	通过该按钮锁定一段录像，可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
全部抓图按钮	实时预览、录像回放工具栏	对所有正在播放视频的播放窗口进行抓图
全部静音按钮	实时预览、录像回放工具栏	设置所有打开的窗口为静音，可通过通用请求响应接口的初始化功能来自定义该控件是否显示，详见 <a href="#">初始化</a>
全部关闭按钮	实时预览、录像回放工具栏	关闭所有正在播放的视频
切换画面分割按钮	实时预览、录像回放工具栏	切换当前所有播放窗口的布局，支持的布局详见 <a href="#">设置当前布局</a>
画面自适应按钮	实时预览、录像回放工具栏	对所有正在播放视频的画面切换成适应窗口或充满窗口模式
全屏按钮	实时预览、录像回放工具栏	控制整个插件窗口进入全屏或退出全屏，并回调窗口进入全屏事件或退出全屏事件。还可按“ESC”键退出全屏
播放速度控制按钮	录像回放工具栏	控制回放的播放速度
单帧前进按钮	录像回放工具栏	回放时不支持单帧前进
单帧后退按钮	录像回放工具栏	正放时大华 SDK 协议接入设备不支持单帧后退
播放与暂停按钮	录像回放工具栏	正放暂停时通过此按钮对选中播放窗口恢复播放，正放播放时通过此按钮对选中播放窗口暂停播放，回放时通过此按钮对选中播放窗口切换到正放
倒放与暂停按钮	录像回放工具栏	倒放暂停时通过此按钮对选中窗口恢复倒放，倒放播放时通过此按钮对选中播放窗口暂停倒放，正放时通过此按钮对选中播放窗口倒放
定位时间点选择按钮	录像回放工具栏	通过此按钮提供对选中播放窗口设置定位播放时间点

## 6. 错误码定义

表 6-1 错误码汇总

错误类型	错误码	错误描述
VIDEOPLUGIN_ERR_FAIL	-1	失败
VIDEOPLUGIN_ERR_SUCCESS	0	成功
VIDEOPLUGIN_ERR_JSON_FORMAT	256	非有效报文
VIDEOPLUGIN_ERR_INVALID_FUNCNAME	257	功能标识无效
VIDEOPLUGIN_ERR_MISSING_REQUIRED_FIELD	258	缺少必要字段
VIDEOPLUGIN_ERR_INVALID_FIELD_VALUE	259	字段值无效
VIDEOPLUGIN_ERR_INVALID_FIELD_VALUE_TYPE	260	字段值类型不匹配
VIDEOPLUGIN_ERR_INVALID_PLUGININSTANCE	261	参数错误，插件实例无效
VIDEOPLUGIN_ERR_INVALID_TOOL_BAR_BUTTONIDS	262	参数错误，toolBarButtonIDs 字段值无效
VIDEOPLUGIN_ERR_INVALID_BUTTONIDS	263	参数错误，buttonIDs 字段值无效
VIDEOPLUGIN_ERR_INVALID_WNDID	264	参数错误，wndId 字段值无效

VIDEOPLUGIN_ERR_GETTED_RSA_KEY	265	错误操作，重复获取公钥
VIDEOPLUGIN_ERR_INVALID_AUTHUUID	266	参数错误，接口认证 UUID 无效
VIDEOPLUGIN_ERR_REPEAT_AUTHUUID	267	参数错误，重复的 authUuid 字段，参数中 authUuid 重复或与插件中已存在的 authUuid 重复
VIDEOPLUGIN_ERR_UNINITIAL	336	未初始化
VIDEOPLUGIN_ERR_FUNCTION_NOT_SUPPORT	512	功能不支持
VIDEOPLUGIN_ERR_SUPPORT_EZVIZ_DIRECT	513	不支持直连萤石云
VIDEOPLUGIN_ERR_SUPPORT_LAYOUT	514	布局不支持
VIDEOPLUGIN_ERR_INVALID_SNAP	768	抓图失败
VIDEOPLUGIN_ERR_TALK_EXIST	1280	对讲占用中

## 7. 附录

### 7.1 如何获取 APPKey 和 APPSecret

请查看开放网站软件产品下的对接指南，网址：<https://open.hikvision.com/docs>。

### 7.2 依赖 OpenAPI 接口汇总

依赖 OpenAPI 接口以及相关功能详见下表：

表 7.2-1 视频 WEB 插件依赖 OpenAPI 接口汇总

OpenAPI 接口名称	视频 web 插件涉及功能
根据编号获取监控点详细信息	实时预览、录像回放
获取单个资源信息	实时预览、录像回放（查询设备获取设备接入协议）
获取监控点预览取流 URL	实时预览
根据监控点编号进行云台操作	实时预览
查询对讲 URL	实时预览
监控点 3D 放大	实时预览
获取监控点回放取流 URL	录像回放
查询萤石校验参数	视频预览（直连萤石预览）、录像回放（直连萤石回放）
录像锁定与解锁	录像回放

### 7.3 问题排查

插件安装在系统目录下“Program Files (x86)\VideoWebPlugin”中，日志位于此目录下的 log 文件夹以及 log 目录的 pfw64 文件夹中。一些比较重要的日志文件名称及日志信息说明如下表：

表 7.3-1 重要日志以及排查建议汇总

日志名称	日志信息说明
HttpDataManager.log	插件网络请求日志，包括四类信息：

	<ol style="list-style-type: none"> <li>1、网络状态探测日志：日志中会记录启动时的网络状态以及网络状态变更的日志，关键字为“online”和“offline”。此信息可以帮助判断网络状态。</li> <li>2、调 HTTP 接口返回的 HTTP 状态码：关键字为：“HTTP STATUS”。如日志中发现 404 的状态码，那应当怀疑此 HTTP 接口不存在（如接口大小写、是否拼错，接口以插件依赖 HTTP 接口为准）。</li> <li>3、组件返回的错误信息：格式如“{“code”:“0x01900608”,“msg”:“send message to driver fail”,“data”:null}”，此 code 为组件的错误码，可在运管中找到此错误码信息以及排查帮助（地址：<a href="http://IP:PORT/center/faq">http://IP:PORT/center/faq</a>，IP、PORT 分别填充平台 IP 与端口，端口一般为 8001）。</li> <li>4、日志中有打印错误日志，但具体错误描述没有：如“find cameraInfo fail, the result is”，此类需要在“HCNetUtilsLog_X_W.log”中排查。</li> </ol>
HCNetUtilsLog_X_W.log	网络通信库日志，这里的 X 从 1 开始。此日志中一般只需关注“FAILED”、“TIMEOUT”关键字，一般出现问题都是由于网络状况不好导致请求超时或建立 TCP 连接失败导致。
VideoPlugin.log	插件日志，记录插件的一些关键日志以及用户请求记录（用户请求记录需要配置 TRACE 基本日志）。关键日志包括萤石取流进程初始化成功与否，点位首次预览回放时的一些参数如是否直连萤石。
VideoSDK.log	<p>32 位操作系统下取流解码相关功能日志。如下：</p> <ol style="list-style-type: none"> <li>1、媒体接口的错误信息，如“HPSClient_PlayBackByTimeEx(asyn) fail, media error code is 0x0173ea87”，搜索关键词“media error”即可。日志中错误码为媒体组件错误码，日志具体信息可以在同级目录“HPSClient.log”中查看，可在运管中找到此错误码信息以及排查帮助（地址：<a href="http://IP:PORT/center/faq">http://IP:PORT/center/faq</a>，IP、PORT 分别填充平台 IP 与端口，端口一般为 8001）。</li> <li>2、直连萤石取流异常日志：如遇到关键词“exception caught”则说明直连萤石在一段时间内没有码流，需要查看 log 文件夹下“opensdk.log”做进一步分析。一般情况下是由于网络状况不佳导致。</li> </ol>
Ezviz.log	<p>萤石取流进程日志。</p> <ol style="list-style-type: none"> <li>1、如果搜索到“stream exception msg received”则说明直连萤石取流异常。需要查看 log 文件夹下“opensdk.log”做进一步分析。一般情况下是由于网络状况不佳导致。</li> <li>2、日志中“OpenSDK_XXX YYY”日志记录了直连萤石取流相关的错误信息描写以及错误码，错误码可在萤石官网 <a href="https://ys7.com/cn">https://ys7.com/cn</a> 找到。其中“XXX”、“YYY”为实际打印的日志信息。</li> </ol>
pfw64\ VideoSDK.log	64 位操作系统下取流解码相关日志。日志分析同 32 位操作系统下 VideoSDK.log 的分析过程。