

# 1. 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_preview.html	视频预览全功能 Demo，可使用此 Demo 验证功能
demo_window_integration_playback.html	录像回放全功能 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
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

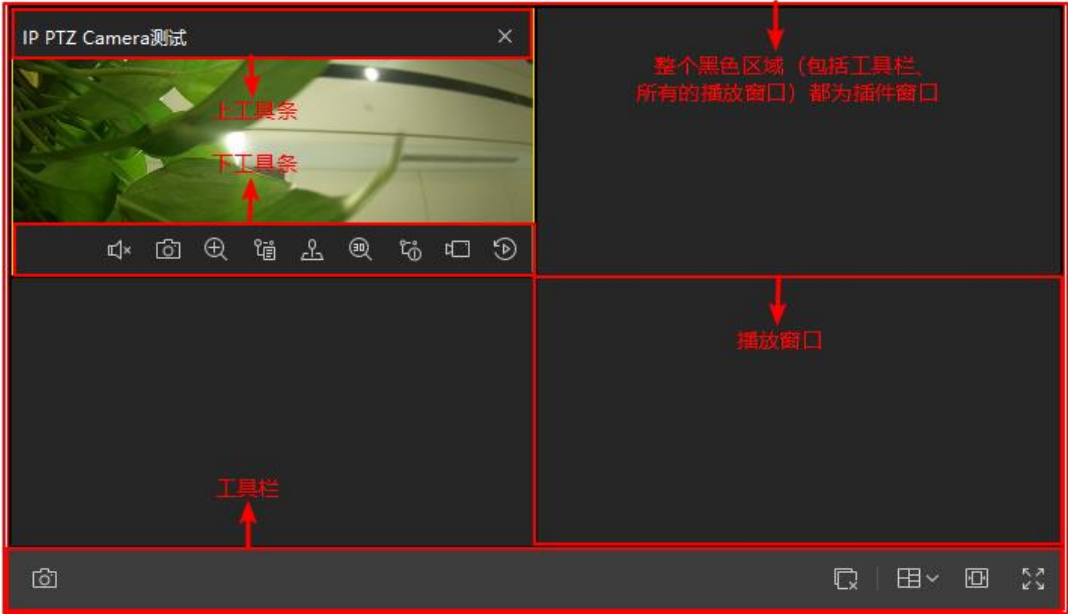


图 1.2-1 视频 WEB 插件（实时预览）UI 模块分类

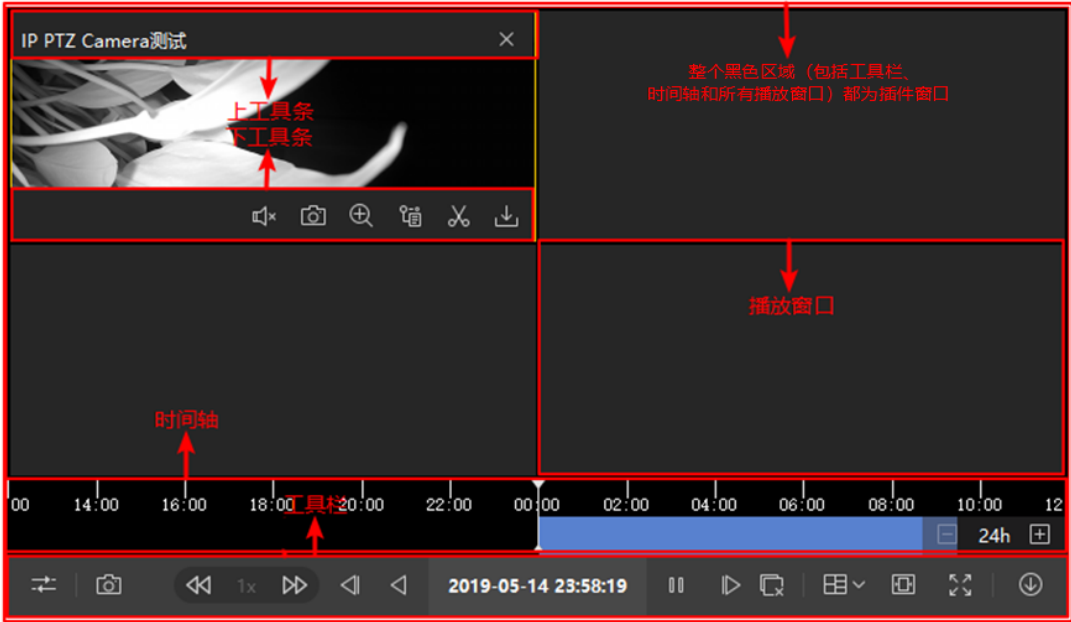


图 1.2-2 视频 WEB 插件（录像回放）UI 模块分类

### 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，海康威视版权所有

- 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
- 语音对讲只支持海康 SDK 协议、ehome 协议接入设备
- 目前仅支持配合 div 标签使用，不支持基于 iframe 方式也不支持浏览器页面弹出页面的使用方式
- IE 浏览器对接情况下至多可启动 6 个插件服务

## 2. 版本更新

### V1.3.1

- |                                 |
|---------------------------------|
| 1、解决初始化后立即执行预览或者回放导致的进程崩溃等已知缺陷； |
|---------------------------------|

### V1.3.0

- |  |
|--|
| 1、新增窗口选中、预览回放消息、抓图、录像以及进入和退出全屏消息回调功能；            |
| 2、初始化中新增配置是否显示工具栏、是否显示智能信息、工具条按钮自定义、新增初始化布局样式功能； |
| 3、预览回放新增集成至 UI 的碗状鹰眼 8 画面显示与 PTZ 碗状画面切换功能；       |
| 4、预览新增集成至 UI 的即时回放功能；                            |
| 5、回放新增集成至 UI 的单帧步进、单帧步退功能，切换布局功能新增布局样式；          |
| 6、优化解码性能：64 位操作系统上针对海康设备，支持超高分辨率解码；              |

### V1.1.0

- |  |
|--|
| 1、新增布局控制、抓图和设置全屏接口，预览、回放接口支持指定窗口、选中窗口、空闲窗口预览和回放，初始化支持设置紧急录像和录像剪辑文件的存储路径； |
| 2、预览新增集成至 UI 的主子码流切换、语音对讲、电子放大、3D 放大、紧急录像、自适应播放功能；                       |
| 3、回放新增集成至 UI 的录像下载、同步播放、异步播放、倒放、电子放大、录像剪辑和自适应播放功能；                       |
| 4、支持 HTTPS 协议与平台进行交互；  |

### V1.0.0

- |   |
|---|
| 1、新增视频预览、录像回放多窗口播放，以及窗口控制、播放控制、云台控制等功能； |
|---|

## 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 接口先将插件窗口隐藏。

### 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
- 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	无需填充

- 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).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，当需要在该窗口叠加网页弹框时或窗口随浏览器滚动到最顶端需要遮挡部分窗口时，使用该接口扣除部分窗口

■ 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

扣除插件窗口后如不再插件窗口位置弹框时需要还原扣除的窗口

■ 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 插件窗口隐藏接口

- 接口定义

JS\_HideWnd()

- 功能说明

隐藏插件窗口

- 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 插件窗口隐藏接口对除接收到全屏事件的插件窗口外的其它窗口隐藏
- ✓ 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

插件窗口隐藏后可调此接口来显示

- 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 插件窗口隐藏接口对除接收到全屏事件的插件窗口外的其它窗口隐

藏

- ✓ 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 插件窗口显示接口

- 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
- 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

```
oWebControl.JS_WakeUp("VideoWebPlugin://")    // oWebControl 为 WebControl 的对象
```
- 特殊使用场景说明



无

3.11 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
  }"
}
```

3.11.1 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
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
    }
}
```

### 3.11.2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
appkey	string	综合安防管理平台提供的 APPKey	是	如何获取 APPKey 详见 <a href="#">如何获取 APPKey 和 APPSecret</a>
secret	string	综合安防管理平台提供的 APPSecret	是	如何获取 APPSecret 详见 <a href="#">如何获取 APPKey 和 APPSecret</a>
ip	string	综合安防管理平台 IP 地址	是	
enableHTTPS	int	是否启用 HTTPS 协议与综合安防管理平台交互	否	如指定为非 0 值，则表示使用 HTTPS 协议与综合安防管理平台进行网络通信，如未指定该字段或指定为 0，表示使用 HTTP 协议与综合安防管理平台进行网络通信。具体需根据现场部署平台提供 http 协议还是 https 协议来设置，若 http 和 https 协议都支持，建议使用 https 协议

port	int	综合安防管理平台端口	是	与是否启用 HTTPS 协议（参见 enableHTTPS 字段）有关，若启用 HTTPS 协议则默认端口为 443，若不启用，默认端口 80
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	抓图存储路径	否	如未指定或指定值无效，使用默认值：{视频 Web 插件运行目录}/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路径	否	如未指定或指定值无效，使用默认值：{视频 Web 插件运行目录}/Video，命名规则为 0-{监控点名称}_{录像时间戳}.mp4（如“测试监控点_1543390276324.mp4”）
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。目前支持的智能信息有智能分析信息、移动侦测信息、POS 信息后叠加信息、图片叠加信息、热成像信息、温度信息的开关配置
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>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</p> <p>0x0104(260)-小鹰眼 0x0200(512)-预览云台控制 0x0201(513)-预览 3D 放大</p> <p>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>0x205(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>下工具条按指定按钮字符串中按钮 ID 顺序来显示</p> <p>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>
--	--	--	--	---

■ 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"
  }"
}
```

■ 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
}
```

3.11.3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  
无
- 入参举例  
无
- Js\_RequestInterface 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
}
```

3.11.4实时预览

- 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。
- 功能标识：startPreview
- 输入参数

名称	类型	描述	是否必填	备注
cameraIndexCode	string	监控点编号	是	
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
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 指定的窗口上预览（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

■ 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
}
```

3.11.5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。

■ 功能标识：startPlayback

■ 输入参数

名称	类型	描述	是否必填	备注
cameraIndexCode	string	监控点 UUID(监控点编号)	是	
startTimeStamp	string	回放开始时间戳	是	单位：秒，开始播放时间
endTimeStamp	string	回放结束时间戳	是	单位：秒，播放结束时间
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（级联情况下，若下级平台的监控点配置了多个存储位置，上级平台回放该监控点时，不支持指定存储位置回放取流，根据服务配置，使用默认存储取流）
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
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 指定的窗口上预览（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

■ 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
}
```

3.11.6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

无

■ 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
}
```

3.11.7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



无

■ 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
}
```

### 3.11.8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

无

■ 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
}
```

### 3.11.9获取当前布局

■ 功能说明

- 使用条件  
初始化后即可获取布局。
- 功能标识: `getLayout`
- 输入参数

无

- 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
    }"
}
```

3.11.10 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
layout	string	布局参数	是	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
}
```

### 3.11.11 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
name	string	图片绝对路径名称，如 “D:\test.jpg”	否	如未填充或无效，使用初始化中的抓图命名规则，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
wndId	int	播放窗口序号（有效值为 1~ 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

■ 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
}
```

3.11.12 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
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

- 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 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 4.2.*
    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
        }
    }
}
```

3.12.1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。
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

3.12.2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
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 0x0310-播放失败 0x0320-播放异常，0x0330-播放结束
        cameraIndexCode: "58e90452772a4d9da7c7ba4cef26dbf0" // 监控点编号
    }
}
```

■ 备注

无

3.12.3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
expand	string	扩展参数	否	result 取值为 0 时，为图片绝对路径文

				件名称
--	--	--	--	-----

■ 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
        cameraIndexCode: "58e90452772a4d9da7c7ba4cef26dbf0", // 监控点编号
        expand: "D:\\test1.jpg"             // 图片文件
    }
}
```

■ 备注

无

3.12.4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
expand	string	扩展参数	否	result 取值为 0 时，为录像存储绝对路径文件名称，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"             // 录像绝对路径文件名称
    }
}
```

■ 备注

无

3.12.5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

## 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 3.1-1。



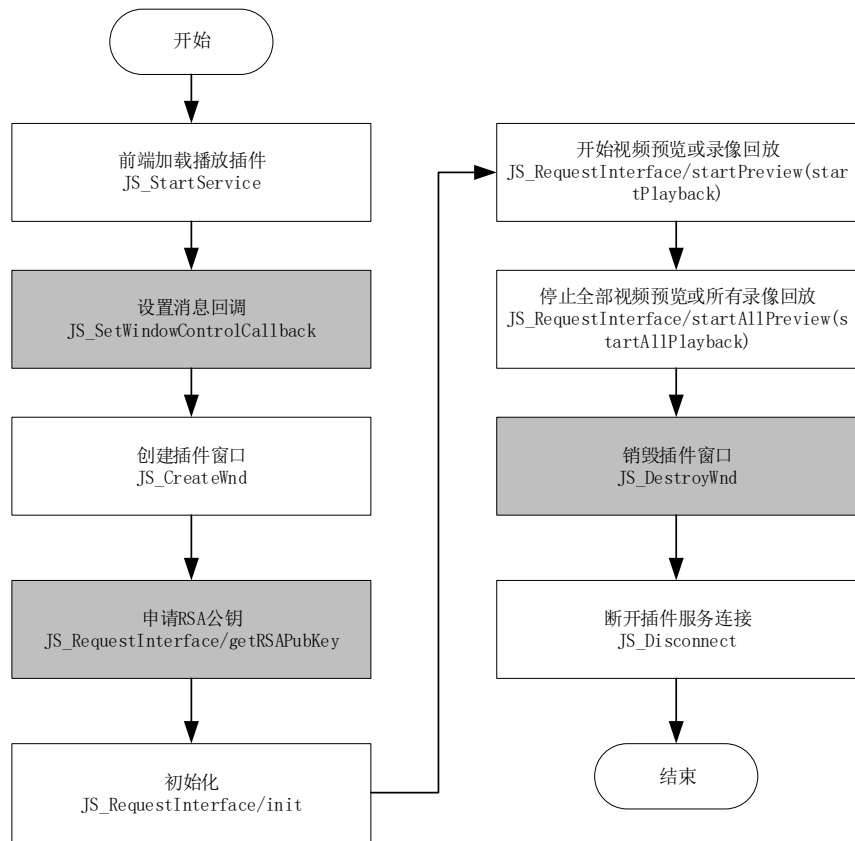


图 3.1-1 VideoWebPlugin 基本对接流程图

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
海康威视版权所有
```

```
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隐藏或显示。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隐藏或显示，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隐藏或显示，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"; // 综合安防管理平台提供的 appkey，必填

// "DuyWn2H7Qb7jB9rPArth"为综合安防管理平台提供的 secret，必填
var secret = setEncrypt("DuyWn2H7Qb7jB9rPArth"); // 对 secret 进行 RSA 公钥加密
var ip = "10.19.132.75"; // 综合安防管理平台 IP 地址，必填
var playMode = 0; // 初始播放模式：0-预览，1-回放，必填

```

海康威视版权所有 28 / 33

```

var port = 443;           // 综合安防管理平台端口，若启用 HTTPS 协议，默认 443，必填
var snapDir = "D:\\SnapDir"; // 抓图存储路径，选填
var videoDir = "D:\\VideoDir"; // 紧急录像或录像剪辑存储路径，选填
var layout = "1x1";       // playMode 指定模式的布局，选填
var enableHTTPS = 1;      // 是否启用 HTTPS 协议与综合安防管理平台交互，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();// 先让窗口隐藏，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  
    });  
}
```

## 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>
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	通过该按钮来展示或隐藏监控点名称、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>
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>
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>
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	倒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，倒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. 附录

## 6.1 如何获取 APPKey 和 APPSecret

请查看综合安防管理平台“文档中心”(“ [http\(s\)://IP:PORT/artemis-portal/document](http(s)://IP:PORT/artemis-portal/document)”，其中 IP 为综合安防管理平台 IP 地址，端口 PORT 为综合安防管理平台端口，如 <http://10.19.132.186/artemis-portal/document>) 的“开发前准备”章节。

## 6.2 依赖 OpenAPI 接口汇总

依赖 OpenAPI 接口以及相关功能详见下表：

表 6.2-1 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

## 6.3 常见问题汇总

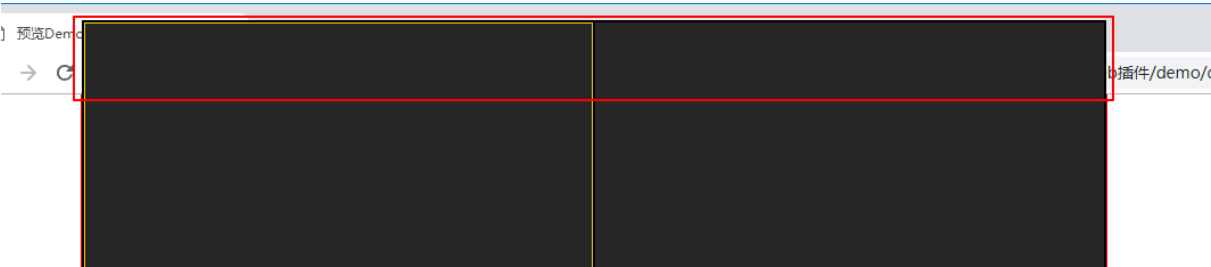
Q: 3D 放大无效果

A: 设备控制类的功能需要设备支持，请确认设备是否支持。

Q: 录像剪辑、紧急录像、录像下载下来的录像无法播放

A: 针对大华设备的录像需要使用大华播放器播放，针对海康设备可以使用 VLC 或者海康播放器播放。

Q: 插件窗口超出浏览器部分未隐藏掉，如图：



A: 请参照“网页开发指南”中的[监听事件](#)代码示例中 resize 和 scroll 事件中的 setWndCover()。

Q: 插件窗口未显示出来，同一台机器上使用 demo 可以显示插件窗口

A: 创建 WebControl 实例步骤中指定的窗口宽、高与 resize 和 scroll 中指定的宽、高不一致，导致窗口被掩盖。



Q: 插件窗口不能随浏览器滚动条滚动而移动

A: 浏览器滚动条滚动和浏览器页面尺寸变化时需要调 JS\_Resize 接口改变插件窗口的大小和位置, 请参照“网页开发指南”中的[监听事件](#)。

Q: 填入错误的 APPKey、APPSecret、综合安防管理平台 IP 地址和端口也能初始化成功

A: 插件初始化会把 APPKey、APPSecret、综合安防管理平台 IP 地址和端口保存在程序中, 不能验证这些数据的正确性, 只有真正发起 HTTP 请求时才能验证正确与否。

Q: 插件窗口提示“查询监控点信息失败”

A: 目前比较常见的原因有: 插件初始化时指定的端口不通或不对、IP 地址非综合安防管理平台 IP 地址、使用 HTTP 协议时代理不支持 HTTP 协议。

Q: 插件窗口提示“APPKey 无效”或“APPSecret 无效”

A: 比较常见的原因有: APPKey 或 APPSecret 缺失不对、初始化时 json 报文中“encryptedFields”字段写错(如写成了“encryptedField”)、初始化时对 APPKey 或 APPSecret 中一个或两个加密后但未指定加密字段或指定加密字段无效。第二种情况如对 APPKey 和 APPSecret 加密, 但通过“encryptedFields”字段只指定 APPKey 或 APPSecret 字段, 或者没有指定加密字段。

Q: 录像回放时画面上的时间早于查询录像的时间

A: 一般为存储设备、前端设备以及综合安防管理平台时间不一致问题, 请先登入综合安防管理平台进行校时。

Q: 一个浏览器页面上是否可以加载多个插件

A: 可以, 但目前插件窗口全屏后其它窗口会浮在全屏窗口上, 需要通过监听进入全屏事件和退出全屏事件来控制其它窗口的显示与隐藏状态, 如何控制窗口的显示与隐藏详见[JS\\_ShowWnd 插件窗口显示接口](#)和[JS\\_HideWnd 插件窗口隐藏接口](#)。不建议一个页面上加载多个插件窗口。

Q: 快速点击或双击插件窗口外的其它区域, 看起来插件窗口在闪烁, 可以看到蓝色的残影

A: 此是浏览器的行为, 与双击浏览器页面中文字, 该文字被选中且背景色为蓝色机制一样。可以按照规避双击浏览器页面文字后文字背景变蓝色的方法来处理。