

众云视频连接平台

(Real-time Video Services - RVS)

视频观看端 SDK 开发指南

Windows

Version 3.0 Beta 3

September 22, 2015

目录

目录	2
1. 概述	4
2. 名词解释.....	5
3. 类和接口说明.....	6
3.1 IRvsViewer 类.....	6
3.2 获取 IRvsViewer 单例.....	6
3.3 获取 Media 单例	6
3.4 获取 Command 单例	6
3.5 Viewer 的初始化	7
3.6 Viewer 模块正常销毁	7
3.7 Viewer 模块强制销毁	7
3.8 Viewer 的设置事件通知类	7
3.9 Viewer 的设置本端语言	7
3.10 写入 crash 日志.....	8
3.11 设置开发者密钥.....	8
3.12 登陆云恩服务器.....	8
3.13 获取 CID 号	9
3.14 断开与云恩服务器的连接.....	9
3.15 客户端连接采集端（获取配置信息，在线状态，通道状态灯信息）	9
3.16 客户端断开与采集端的连接.....	9
3.17 局域网内搜索采集端.....	9
3.18 IRvsViewerStreamerInfo 接口	10
3.19 设置事件通知类的.....	10
3.20 查询采集端基本信息.....	10
3.21 释放采集端基本信息查询资源.....	10
3.22 获取采集端支持的功能.....	10
3.23 释放采集端支持的功能查询资源.....	11
3.24 查询采集端的定时录制的设置信息.....	11
3.25 释放定时录制的设置信息查询资源.....	11
3.26 查询采集端运动侦测设置信息.....	11
3.27 查询采集端传感器设置信息.....	11
3.28 释放采集端传感器设置信息查询资源.....	11
3.29 设置采集端定时录制的时间计划.....	12
3.30 设置采集端运动侦测配置信息.....	12
3.31 设置采集端传感器信息.....	12
3.32 设置采集端的名称.....	12
3.33 设置采集端报警通知邮箱.....	12
3.34 设置采集端推送开关.....	12
3.35 设置采集端报警输出开关.....	13
3.36 设置采集端定时录制保存天数.....	13
3.37 设置采集端报警是否已配置.....	13
3.38 设置 3.0 之前的 IPC 采集端报警参数	13

3.39 IRvsViewerMedia 接口	13
3.40 打开采集端上的一路实时流.....	13
3.41 打开采集端上的一个录像文件.....	14
3.42 打开存储在云端的一个录像文件.....	14
3.43 关闭一路媒体流.....	14
3.44 设置媒体流的回调处理接口.....	14
3.45 获取媒体流的描述信息.....	15
3.46 获取视频数据	15
3.47 获取解码后的视频数据.....	15
3.48 获取音频数据	15
3.49 获取解码后的音频数据.....	15
3.50 暂停发送媒体流.....	16
3.51 恢复发送媒体流.....	16
3.52 将媒体流切到指定时间点播放.....	16
3.53 开始录制视频到本地.....	16
3.54 停止本地录制	16
3.55 请求 JPEG 格式的媒体流.....	16
3.56 请求一幅 JPEG 图片.....	17
3.57 请求采集端的录制文件列表.....	17
3.58 根据文件名删除采集端的录制文件.....	17
3.59 根据时间段删除采集端的录制文件.....	17
3.60 获取每一天的云视频文件数目.....	17
3.61 查询某一天的云视频文件详情.....	18
3.62 请求云视频文件的图标.....	18
3.63 通知采集端购买了云服务.....	18
3.64 设置逆向媒体流数据的格式.....	18
3.65 启动逆向媒体流.....	18
3.66 发送逆向媒体流数据.....	18
3.67 停止逆向媒体流.....	19
3.68 IRvsViewerCmd 类.....	19
3.69 设置 IRvsViewerCmd 类的回调处理接口	19
3.70 设置采集端的登录名和密码.....	19
3.71 改变采集端视频编码质量.....	19
3.72 强制采集端编码一个 I 帧.....	19
3.73 镜头翻转	20
3.74 前后摄像头切换.....	20
3.75 开关闪光灯	20
3.76 切换前后镜头	20
3.77 设置采集端时间时区信息.....	20
3.78 格式化采集端的 SD 卡	20
3.79 控制采集端 PTZ 或位移运动.....	21
3.80 获取采集端的时间时区信息.....	21
3.81 获取采集端的 sd 卡容量信息.....	21
3.82 发送自定义信令.....	21

3.83 ST_RVS_MEDIA_AVDESC 结构体.....	22
3.84 ST_RVS_STREAMER_INFO 类.....	22
3.85 ST_RVS_STREAMER_SUPPORTSERVICES 结构.....	24
3.86 ST_RVS_STREAMER_TIMERECORDINFO 类.....	24
3.87 ST_RVS_STREAMER_ALARMRECORDINFO 类.....	25
3.88 回调接口 IRvsViewerEvent	25
3.89 回调接口 IRvsViewerCmdEvent	26
3.90 回调接口 IRvsViewerMediaEvent	27
3.91 回调接口 IRvsViewerStreamerInfoEvent	28
4. 使用步骤.....	29
5. SDK 调用流程.....	29
5.1 简要流程.....	29
5.2 详细流程.....	30

1. 概述

ICHANO SDK 为网络摄像机或智能设备方案商及生产商提供了基于互联网的多媒体连接服务，使设备方案商通过简明的 API 调用，就能将所采集到的音、视频等多媒体信息通过互联网传输到用户的手机、电脑上，满足用户的监控、直播、对讲等相关的各种需求。

ICHANO SDK 分为采集端和观看端两部分，并为不同的操作系统平台提供相应的版本，本文档为 Windows 平台上观看端的 SDK 应用指南。不同操作系统的 ICHANO SDK 的 API 接口的定义相似，工作流程基本相同，可以互为参照。采集端的 SDK 和观看端的 SDK 有较多的关联性，故开发者也可以将观看端文档与采集端文档互为参照。

ICHANO SDK 观看端包含有 IRvsViewer、RvsViewerStreamerInfo、IRvsViewerMedia、IRvsViewerCmd 这 4 个主要的接口，其它类和接口均是围绕和辅助这 4 个类进行定义和使用。

观看端这 4 个主类与采集端的 Streamer、Media、Command 这 3 个主类（Linux 平台则对应为 rvs_streamer.h、rvs_streamer_media.h、rvs_streamer_cmd.h 这 3 个头文件所对应的模块）的主要功能及主要交互关系如下图所示。

2. 名词解释

本文档中所出现的容易引起歧义的名词，解释如下：

应用层	调用本 SDK 接口的代码，在本文档中称之为应用层。本文档第 5 章示例代码，即为应用层代码的示例。
采集端设备	本文档用采集端设备来表示带有视频采集功能和网络传输功能的智能设备，即上文所述“网络摄像机或智能设备”。
用户	本文档中出现的用户，并不表示使用 ICHANO SDK 的客户，而是指购买了采集端设备的使用者个人。
观看端 app	基于 ICHANO SDK 观看端所开发的，安装在用户的手机或电脑上的应用软件，用户可使用该软件来远程连接、设置采集端设备并实现各种监控相关的需求。在 ICHANO SDK 的旧版本文档里也称之为客户端。
观看端设备	也就是安装了上述观看端 app 的手机或电脑。在 ICHANO SDK 的旧版本文档里也称之为客户端设备。
Streamer	ICHANO SDK 采集端的总体处理的类名，与观看端的 Viewer 相对应。在本文档中，若没有特别强调，一个 Streamer 即表示一个采集端。
CID 号	ICHANO SDK 的云端为每个采集端设备分配的一个识别号码，由数字组成。用户第一次使用某个采集端设备时，须在观看端 app 输入该设备的 CID 号码，以区分于其它采集端设备。 当观看端设备与采集端设备处在同一个局域网内时，用户可以在观看端 app 进行“局域网内搜索”操作，来获得局域网内的采集端设备的 CID 号。 应用层若需要及时获取 CID 号，则可在会话状态[见下文]为 E_RVS_SESSION_STATE_CONNECTED 之后调用 CID 号获取接口。 相似地，ICHANO SDK 的云端也为每个观看端设备分配了一个 CID 号，采集端设备中可用此 CID 号来区分前来访问的不同观看端。
设备 ID	生产商为每个采集端设备写入的用于识别的字符串，可由任意数字字母组成。ICHANO SDK 的云端将为每一个设备 ID 产生一个独有的 CID 号供用户使用。 [设备 ID 号需要事先向南京云恩通信有限公司报备以获得授权，否则无法获得有效的 CID 号，也无法正常使用本 SDK 的接口]
镜头数	一般情况下，一个采集端设备只装有一个镜头，镜头数为 1。ICHANO SDK 考虑到支持一些特殊设备可能装有 2 个或以上的镜头，故需要应用层对所支持的镜头数量进行设置。
镜头索引	一般情况下，一个采集端设备内只有一个镜头，镜头索引[Camera Id]取值为 0。当出现多个镜头时，应用层使用该索引来区分不同的镜头。
StreamChannel	一个镜头，常有主、次两路码流[也叫主、辅码流]，两路码流的视频的分辨率、帧率、比特率等属性都不同，有时两路码流对应的音频属性也有区分，ICHANO SDK 以 StreamChannel 的概念来表示同一个镜头的不同码流。并以 StreamCount 和 Stream Id 来表示码流的数量和索引。
会话 (SESSION)	ICHANO SDK 允许多个观看端同时连接到同一个采集端进行实时观看和操作。采集端为每一个实时连接、观看和操作的观看端建立一个会话，当观看端退出实时观看页面后，该会话结束。 由于建立会话会增加采集端设备的 CPU 负荷及内存消耗，所以需要应用层根据采集端设备的实际能力来指定最大支持多少会话。（相应地，该设备最多支持多少个观看端同时进行实时连接、观看和操作） 此外，应用层也可以根据会话的状态，来管理该会话所占用的应用层资源。

PTZ	一般指 Pan/Tilt/Zoom，表示云台的全方位移动及镜头控制；。
-----	-------------------------------------

• 修订记录

时间	文档版本	修订人	备注
2015-06-15	V3.0.beta1	王定新	初稿
2015-09-22	V3.0.beta3	印体亮	修改

3. 类和接口说明

3.1 IRvsViewer 类

IRvsViewer 类是 ICHANO SDK 观看端的总体处理的类，负责登录鉴权、处理消息回调通知上层应用。该类是单例，应调用 RvsQueryInterface 来获得该类的实例。

此外，IRvsViewerStreamerInfo、IRvsViewerMedia、IRvsViewerCmd 三个类也是单例，都可以通过 RvsQueryInterface 查询获得接口单实例。其中 IRvsViewerStreamerInfo 用于对关联的采集端进行管理（包括进行信息查询和参数设置等）；IRvsViewerMedia 用于对音、视频码流相关的处理进行设置；IRvsViewerCmd 用于观看端与采集端之间的信令交互处理。

3.2 获取 IRvsViewer 单例

方法： RvsQueryInterface(IRvsViewer::IID, (RVS_VOID**) & m_pIRvsViewer);

说明： 获取 Viewer 类的唯一实例。

3.3 获取 Media 单例

方法： RvsQueryInterface(IRvsViewerMedia::IID, (RVS_VOID**) & g_pIRvsViewerCmd);

说明： 获取该 viewer 实例的成员 media。

3.4 获取 Command 单例

方法： RvsQueryInterface(IRvsViewerMedia::IID, (RVS_VOID**) & g_pIRvsViewerMedia);

说明： 获取该 viewer 实例的成员 command。

3.5 Viewer 的初始化

方法： virtual RVS_INT Init(const RVS_UC* pucWorkPath, const RVS_UC* pucCachePath, const RVS_UC* pucAppVersion);

说明： Viewer 的初始化就是整个 ICHANO SDK 的初始化工作。

参数：

pucWorkPath: 设置配置, 日志等工作路径,

pucCachePath: 媒体数据缓存目录;

pucAppVersion: 软件版本号

3.6 Viewer 模块正常销毁

方法： virtual RVS_VOID Release();

说明： 释放 Init 接口申请的资源。

参数： 无。

3.7 Viewer 模块强制销毁

方法： virtual RVS_INT Destroy();

说明： 强制释放 Init 接口申请的资源。

参数： 无。

3.8 Viewer 的设置事件通知类

方法： virtual RVS_INT SetRvsViewerEvent(IRvsViewerEvent* pIEvent);

说明： Viewer 接口异步请求处理, 需要实现对应的事件处理虚基类 IRvsViewerEvent, 然后调用此方法, 把虚基类的指针设置到 IRvsViewer 中。

参数：

IRvsViewerEvent: 每一个接口都有一个对应的回调通知虚基类, 通过实现该虚基类, 就可以对回调通知进行处理, IRvsViewer 接口对应的回调通知虚基类是 IRvsViewerEvent。

3.9 Viewer 的设置本端语言

方法： virtual RVS_INT SetLanguage(EN_RVS_VIEWER_LANGUAGE enLanguage);

说明： Viewer 端设置本端的语言类型。

参数：

enLanguage: 语言类型, 定义如下:

```
typedef enum enum_RVS_VIEWER_LANGUAGE {  
    E_RVS_VIEWER_LANGUAGE_zh_CN        = 1, // 中文简体  
    E_RVS_VIEWER_LANGUAGE_en          = 2, // 英语 (美国)  
    E_RVS_VIEWER_LANGUAGE_zh_TW       = 3, // 中文繁体
```

```

E_RVS_VIEWER_LANGUAGE_fr      = 4, // 法语
E_RVS_VIEWER_LANGUAGE_ja      = 5, // 日语
E_RVS_VIEWER_LANGUAGE_es      = 6, // 西班牙语
E_RVS_VIEWER_LANGUAGE_ko      = 7, // 韩语
E_RVS_VIEWER_LANGUAGE_it      = 8, // 意大利语
E_RVS_VIEWER_LANGUAGE_pt      = 9, // 葡萄牙语
E_RVS_VIEWER_LANGUAGE_ru      = 10, // 俄罗斯语
E_RVS_VIEWER_LANGUAGE_th      = 11, // 泰语
E_RVS_VIEWER_LANGUAGE_de      = 12, // 德语
E_RVS_VIEWER_LANGUAGE_ar      = 13, // 阿拉伯语
E_RVS_VIEWER_LANGUAGE_el      = 14 // 希腊文
}EN_RVS_VIEWER_LANGUAGE;

```

3.10 写入 crash 日志

方法: virtual RVS_VOID WriteDumpLog();

说明: 写入 crash 时日志缓存中的内容。

参数: 无。

3.11 设置开发者密钥

方法: virtual RVS_INT SetLoginInfo(const RVS_UC* pucCompanyID,
RVS_ULL ullCompanyKey,
const RVS_UC* pucAppID,
const RVS_UC* pucLicense ;

说明: 设置 ICHANO SDK 登录服务器所需权限的密钥。

参数: 所需权限的密钥需要联系南京云恩通讯有限公司得到并妥善保管。若密钥有误会导致登录服务器失败。

3.12 登陆云恩服务器

方法: void OnLoginResult()

说明: 此接口启动了采集端与服务器的连接，连接结果会通过事件通知接口 IRvsViewerEvent 的通知方法通知登录结果，

```

OnLoginResult(EN_RVS_LOGIN_STATE enLoginState,
              RVS_INT uiProgressRate,
              EN_RVS_LOGIN_ERR enErrCode)

```

OnLoginResult 通知成功以后，调用 GetCID() 可得到一个 CID 号，并具备了被 ICHANO 的采集端或者观看端访问的能力。

3.13 获取 CID 号

方法: RVS_ULL GetCID() ;

说明: 获取观看端的 CID 号, 程序第一次 Login()成功后, 会从云恩服务器获得一个唯一的 CID 号。调用此方法可以获取此号。采集端可以根据此 CID 号来区分不同的观看端。

3.14 断开与云恩服务器的连接

方法: void Logout()

说明: 断开与云恩服务器的连接。

3.15 客户端连接采集端（获取配置信息，在线状态，通道状态灯信息）

方法: RVS_INT ConnectStreamer(RVS_ULL ullStreamerCID,
 const RVS_UC* pucUserName,
 const RVS_UC* pucPassword) = 0 ;

说明: 观看端添加连接一个采集端。添加成功之后, 采集端的在线状态发生变化时, 观看端将会在 IRvsViewerEvent 接口中得到状态通知。采集端发出的推送信息, 此观看端将会收到。观看端可以通过调用 IRvsViewerStreamerInfo 的方法来获得所绑定的 Streamer 的更多信息。

绑定成功之后的 Streamer 状态发生变化时, SDK 会调用 IRvsViewerEvent 来通知本地观看端应用层。

3.16 客户端断开与采集端的连接

方法: virtual RVS_INT DisconnectStreamer(RVS_ULL ullStreamerCID);

说明: 成功返回 RVS_OK, 失败为 RVS_ERR。

3.17 局域网内搜索采集端

方法: virtual RVS_INT LANSearchStreamer();

说明: 本地[指同一个局域网内]搜索采集端, 搜索结果将通过 IRvsViewerEvent 回调方法返回给应用层。具体回调方法的定义见 OnLanSearchStreamerResult 的定义。

3.18 IRvsViewerStreamerInfo 接口

IRvsViewerStreamerInfo 用于设置、获取采集端配置信息。该类是单例，应调用来获得该类的实例。

IRvsViewerStreamerInfo 与 IRvsViewerCmd 的区别在于，前者主要是设置采集端的能力信息、描述信息、与观看端的 UI 界面相关的信息等。后者则主要是控制采集端的镜头的实时动作，与用户的操作相关。

3.19 设置事件通知类的

方法： RVS_INT SetViewerStreamerInfoEvent (IRvsViewerStreamerInfoEvent* pIStreamerInfoEvent)

说明： 设置事件通知类的，c++ 把所有的回调方法都封装的一些类里面了，其实就是 c 的结构体。比如登陆，登陆成功后通知谁，如果设置了事件通知类，那么就调用类里面登陆成功的对应的回调事件。

参数： 见各自定义。

3.20 查询采集端基本信息

方法： RVS_INT GetStreamerInfo(RVS_ULL ullStreamerCID, ST_RVS_STREAMER_INFO** ppstStreamerInfo) ;

说明： 根据采集端 cid 查询采集端基本描述信息，由于返回的接口体信息比较多，属于动态分配的内存，所以查询处理完结果以后，需要调用对应的 FreeStreamerInfo 方法进行内存空间释放。

参数： 返回值为所查询到的描述信息，具体见 IRvsViewerStreamerInfo 的定义。

3.21 释放采集端基本信息查询资源

方法： RVS_VOID FreeStreamerInfo(ST_RVS_STREAMER_INFO* ppstStreamerInfo);

说明： 释放 GetStreamerInfo 申请的内存等资源。

3.22 获取采集端支持的功能

方法： RVS_INT GetStreamerSupportServices (RVS_ULL ullStreamerCID, ST_RVS_STREAMER_SUPPORTSERVICES** ppstStreamerSupportServices) ;

说明： 获取采集端支持的能力集, 使用结束需要调用 FreeStreamerSupportServicesStruct。

参数： 返回采集端能力信息，具体见 ST_RVS_STREAMER_SUPPORTSERVICES 的定义。

3.23 释放采集端支持的功能查询资源

方法: RVS_VOID FreeStreamerSupportServicesStruct(ST_RVS_STREAMER_SUPPORTSERVICES *pstStreamerSupportServices);

说明: 释放 GetStreamerSupportServices 申请的内存等资源。

3.24 查询采集端的定时录制的设置信息

方法: RVS_INT GetStreamerRecordSchedule(RVS_ULL ullStreamerCID, RVS_INT iCamIndex, ST_RVS_STREAMER_TIMERECORDINFO** ppstRvsTimeRecordInfo)

说明: 查询采集端的定时录制的设置信息。使用结束需要释放内存

FreeStreamerRecordScheduleInfoStruct

参数: 返回采集端的定时录制信息，具体见 ST_RVS_STREAMER_TIMERECORDINFO 的定义。

3.25 释放定时录制的设置信息查询资源

方法: RVS_VOID FreeStreamerRecordScheduleInfoStruct(ST_RVS_STREAMER_TIMERECORDINFO *pstRvsTimeRecordInfo);

说明: 释放 GetStreamerRecordSchedule 申请的内存等资源

3.26 查询采集端运动侦测设置信息

方法: RVS_INT GetStreamerMotionSchedule(RVS_ULL ullStreamerCID, RVS_INT iCamIndex, ST_RVS_STREAMER_ALARMRECORDINFO** ppstRvsAlarmRecordInfo)

说明: 查询采集端报警（运动侦测）设置信息，使用结束需要释放数据空间；

参数: 具体见 ST_RVS_STREAMER_ALARMRECORDINFO 的定义。

3.27 查询采集端传感器设置信息

方法: RVS_INT GetStreamerSensors(RVS_ULL ullStreamerCID, ST_RVS_STREAMER_SENSORS** ppstRvsSensors);

说明: 查询采集端传感器设置信息。具体见 ST_RVS_STREAMER_SENSORS 的定义，使用结束需要调用 FreeStreamerSensorsInfoStruct 释放数据空间。

3.28 释放采集端传感器设置信息查询资源

方法: RVS_VOID FreeStreamerSensorsInfoStruct(ST_RVS_STREAMER_SENSORS *pstRvsSensors);

说明: 释放采集端传感器设置信息查询 GetStreamerSensors 的资源。

3.29 设置采集端定时录制的时间计划

方法: `RVS_INT SetStreamerTimeReocrdInfo(RVS_ULL ullStreamerCID,
RVS_INT iCamIndex, ST_RVS_STREAMER_TIMERECORDINFO* pstRvsTimeRecordInfo)`

说明: 设置采集端定时录制的时间计划。

3.30 设置采集端运动侦测配置信息

方法: `RVS_INT SetStreamerAlarmRecordInfo(RVS_ULL ullStreamerCID,
RVS_INT iCamIndex,
ST_RVS_STREAMER_ALARMRECORDINFO* pstRvsAlarmRecordInfo)`

说明: 设置采集端运动侦测报警的时间计划及灵敏度等配置信息。

3.31 设置采集端传感器信息

方法: `RVS_INT SetStreamerSensorSchedule(RVS_ULL ullStreamerCID, RVS_INT iSensorType,
RVS_INT iSensorIndex, ST_RVS_STREAMER_SENSORINFO* pstRvsSensorInfo)`

说明: 设置采集端传感器信息。

3.32 设置采集端的名称

方法: `RVS_INT SetStreamerName(RVS_ULL ullStreamerCID, const RVS_UC* pucStreamerName);`

说明: 设置采集端的名称。

3.33 设置采集端报警通知邮箱

方法: `RVS_INT SetStreamerEmailInfo(RVS_ULL ullStreamerCID,
RVS_BOOL bEmailEnable,
const RVS_UC* pucEmailAddr)`

说明: 设置采集端发生报警时，是否需要平台将该采集端的报警信息以邮件形式发送到指定邮箱。

3.34 设置采集端推送开关

方法: `RVS_INT SetStreamerPushEnable(RVS_ULL ullStreamerCID,
RVS_BOOL bPushEnable)`

说明: 采集端发生报警时，观看端是否希望接收来自该采集端的报警推送信息。

3.35 设置采集端报警输出开关

方法: `RVS_INT SetStreamerNoticeEnable(RVS_ULL ullStreamerCID, RVS_BOOL bNoticeEnable);`

说明: 设置采集端是否启用报警通知, 比如响铃、闪光

3.36 设置采集端定时录制保存天数

方法: `RVS_ULL SetStreamerRecordDelDays(RVS_ULL ullStreamerCID, RVS_INT iDelDays);`

说明: 设置采集端定时录制保存天数, 超过时长天数就会自动删除录制文件。

3.37 设置采集端报警是否已配置

方法: `RVS_ULL SetStreamerDetectFlag(RVS_ULL ullStreamerCID, RVS_UI iFlag);`

说明: 设置采集端报警是否已配置。

参数:

iFlag: `EN_RVS_STREAMER_DETECT_FLAG`

3.38 设置 3.0 之前的 IPC 采集端报警参数

方法: `RVS_INT SetStreamerOldIPCAlarm(RVS_ULL ullStreamerCID, RVS_INT iEmailFlag, const RVS_UC* pucEmailAddr, RVS_INT iPushFlag, RVS_INT iMotionStatus, RVS_INT iMotionSensitivity, RVS_INT iAlarmSet1, RVS_INT iAlarmSet2, RVS_INT iAlarmSet3);`

说明: 设置 3.0 之前的 IPC 采集端报警参数。

3.39 IRvsViewerMedia 接口

`Media` 是 `ICHANO SDK` 中, 对音、视频码流进行处理的类, 该类是单例, 应调用 `Viewer.getMedia()` 来获得该类的实例。主要用于和采集端建立连接, 读取视频流和音频流。提供查询采集端录像文件、云端视频文件, 播放录像和云视频等功能。

3.40 打开采集端上的一路实时流

方法: `RVS_HANDLE OpenLiveStream(RVS_ULL ullStreamerCID,`

```
RVS_INT iCamIndex,  
RVS_INT iStreamIndex,  
RVS_INT iMicIndex)
```

说明：和 streamerCID 指定的采集端建立连接，获取一路实时的音视频流。

参数：streamerCID 采集端的 cid
cameraIndex 镜头 id，从 0 开始，
streamId 镜头码流 id，从 0 开始
micId 录音设备 id，从 0 开始

返回值：streamId 流 id, 唯一标识音视频流。

3.41 打开采集端上的一个录像文件

方法：RVS_HANDLE OpenRemoteRecordFileStream(RVS_ULL ullStreamerCID,
const RVS_UC* pucFileName)

说明：打开采集端上的录像文件。

参数：streamerCID 采集端 cid
fileName 文件名

返回值：streamId 流 id, 唯一标识音视频流

3.42 打开存储在云端的一个录像文件

方法：RVS_HANDLE OpenCloudRecordFileStream(RVS_ULL ullStreamerCID, const RVS_UC*
pucEid)

说明：打开云端录像文件。

参数：streamerCID 采集端 cid
eid 云视频文件唯一标识

返回值：streamId 流 id, 唯一标识音视频流

3.43 关闭一路媒体流

方法：RVS_INT CloseStream(RVS_HANDLE hHandle)

说明：关闭一路音视频流，可关闭 4.3.1 到 4.3.3 节中方法所打开的实时视频、打开的采集端录像、云端录像等媒体流。

3.44 设置媒体流的回调处理接口

方法：RVS_INT SetViewerMediaEvent(IRvsViewerMediaEvent* pIMediaEvent)

说明：设置媒体流的回调处理接口，通知媒体流状态和错误信息。

4.3.1 到 4.3.3 节中方法打开了一路媒体流之后，ICHANO SDK 将调用此回调接口来反馈该媒体流的实时状态。

具体 IRvsViewerMediaEvent 见定义。

3.45 获取媒体流的描述信息

方法： `RVS_INT GetStreamDesc(RVS_HANDLE hHandle, _OUT ST_RVS_MEDIA_AVDESC* pstMediaDesc)`

说明： 获取媒体流的描述信息。详细见 `ST_RVS_MEDIA_AVDESC` 的定义。

3.46 获取视频数据

方法： `RVS_INT GetVideoData(RVS_HANDLE hHandle,
_OUT RVS_UC** ppucStreamBuffer,
_OUT RVS_UI* puiBufLen,
_OUT RVS_UI* puiTimeStamp)`

说明： 获取视频数据，当收到 `OnPlayStreamStatus` 中的回调状态为 `EN_RVS_MEDIASTREAM_STATE` 状态为 `EN_RVS_MEDIASTREAM_STATE_CREATED` 时调用此方法。此方法应循环调用。

注意： 此接口中的数据为未解码的视频码流数据。

3.47 获取解码后的视频数据

方法： `RVS_INT GetVideoYUVData(RVS_HANDLE hHandle,
_OUT RVS_UC** y,
_OUT RVS_UC** u,
_OUT RVS_UC** v,
_OUT RVS_UI* puiTimeStamp)`

说明： 获取解码后的视频数据，当收到 `OnPlayStreamStatus` 中的回调状态为 `EN_RVS_MEDIASTREAM_STATE` 状态为 `EN_RVS_MEDIASTREAM_STATE_CREATED` 时调用此方法。此方法应循环调用。

3.48 获取音频数据

方法： `RVS_INT GetAudioData(RVS_HANDLE hHandle,
_OUT RVS_UC** ppucStreamBuffer,
_OUT RVS_UI* puiBufLen,
_OUT RVS_UI* puiTimeStamp)`

说明： 获取音频数据，当收到 `OnPlayStreamStatus` 中的回调状态为 `EN_RVS_MEDIASTREAM_STATE_CREATED` 时调用此方法。

3.49 获取解码后的音频数据

方法： `GetAudioPcmData(RVS_HANDLE hHandle,
_OUT RVS_UC** ppucStreamBuffer,
_OUT RVS_UI* puiBufLen,`

_OUT RVS_UI* puiTimeStamp)

说明：获取音频解码后的 pcm 数据，当收到 OnPlayStreamStatus 中的回调状态为 EN_RVS_MEDIASTREAM_STATE_CREATED 时调用此方法。

3.50 暂停发送媒体流

方法：RVS_INT PauseStream(RVS_HANDLE hHandle) ;

说明：暂停发送媒体流。

3.51 恢复发送媒体流

方法：RVS_INT ResumeStream(RVS_HANDLE hHandle)

说明：恢复发送媒体流。

3.52 将媒体流切到指定时间点播放

方法：RVS_INT SeekStream(RVS_HANDLE hHandle, RVS_UI uiTimeStamp)

说明：观看采集端录制、云端录制视频时，拖动观看时间。

3.53 开始录制视频到本地

方法：RVS_INT StartLocalRecord(RVS_HANDLE hHandle, const RVS_UC* pucPathFileName)

说明：开始录制视频到观看端文件系统。

3.54 停止本地录制

方法：RVS_INT StopLocalRecord(RVS_HANDLE hHandle)

说明：停止本地录制。

3.55 请求 JPEG 格式的媒体流

方法：RVS_INT RequestJpeg(RVS_ULL ullStreamCID,
RVS_INT iCamIndex,
RVS_INT iStreamId,
EN_RVS_JPEG_TYPE enJpegType,
RVS_ULL* pullRequestID)

说明：请求 JPEG 格式的媒体流。ICHANO CLIENT SDK 除了支持 H264 的码流之外，还支持纯 JPEG 图片流。请求的码流将通过回调接口 OnRecvJpeg 获得，具体见 OnRecvJpeg 的定义。

3.56 请求一幅 JPEG 图片

方法: `RVS_INT RequestJpegFile(RVS_ULL ullStreamCID,
RVS_UC* pucFileName,
RVS_ULL* pullRequestID)`

说明: 请求一幅 JPEG 图片。此接口是在码流为 H264 的情况下，向采集端请求一幅 JPEG 图片，具体图片内容由采集端与观看端的应用层根据 filename 确定。请求的 JPEG 图片将通过回调接口 OnRecvJpeg 获得，具体见 OnRecvJpeg 的定义。

3.57 请求采集端的录制文件列表

方法: `RVS_ULL RequestStreamerRecordFiles(RVS_ULL ullStreamerCID,
RVS_INT iCamIndex,
RVS_INT iPageIndex,
RVS_INT iCountPerPage,
RVS_UC* pucBeginTime,
RVS_UC* pucEndTime,
RVS_UI uiRecordType)`

说明: 请求采集端的录制文件列表。

3.58 根据文件名删除采集端的录制文件

方法: `RVS_ULL RemoveRecordFileByName(RVS_ULL ullStreamerCID,
RVS_UC* pucFileName,
RVS_UI uiRecordType)`

说明: 根据文件名删除采集端的录制文件。

3.59 根据时间段删除采集端的录制文件

方法: `RVS_ULL RemoveRecordFilesByTime(RVS_ULL ullStreamerCID,
RVS_INT iCamIndex,
RVS_UC* pucBeginTime,
RVS_UC* pucEndTime,
RVS_UI uiRecordType)`

说明: 根据时间段删除采集端的录制文件。

3.60 获取每一天的云视频文件数目

方法: `RVS_ULL RequestFileCountPerDay(RVS_ULL ullStreamerCID)`

说明：获取每一天的视频文件数目。

3.61 查询某一天的云视频文件详情

方法：RVS_ULL RequestCloudRecordFiles(RVS_ULL ullStreamerCID,
RVS_INT iCamIndex,
RVS_INT iPageIndex,
RVS_INT iCountPerPage,
RVS_UC* pucDay, /** 2015-05-15 */
RVS_UI uiRecordType)

说明：查询某一天的云视频文件详情。

3.62 请求云视频文件的图标

方法：RVS_ULL RequestCloudFileIcon(RVS_ULL ullStreamerCID, RVS_UC* pucEid)

说明：请求云视频文件的图标。

3.63 通知采集端购买了云服务

方法：RVS_INT SetStreamerCloudBegin(RVS_ULL ullStreamerCID);

说明：通知采集端购买了云服务。

3.64 设置逆向媒体流数据的格式

方法：RVS_INT SetRevStreamProperty(ST_RVS_MEDIA_AVDESC* pstAudioFormat)

说明：设置逆向媒体流数据的格式，具体参见 ST_RVS_MEDIA_AVDESC 定义。

3.65 启动逆向媒体流

方法：RVS_HANDLE StartRevStream(RVS_ULL ullStreamerCID)

说明：开始向采集端发送逆向音频流。

3.66 发送逆向媒体流数据

方法：RVS_INT WriteRevStreamData(EN_RVS_REV_STREAM_TYPE enStreamType,
RVS_UC* pucData,
RVS_UI uiLen,
RVS_UI uiTimestamp)

说明：发送逆向媒体流数据的接口。此接口将被循环调用。

3.67 停止逆向媒体流

方法： RVS_INT StopRevStream(RVS_HANDLE hRevHandle)

说明： 停止向采集端发送逆向音频流。

3.68 IRvsViewerCmd 类

IRvsViewerCmd 类是在采集端与观看端之间进行信令交互处理的类，该类是单例，调用 RvsQueryInterface 来获得该类的实例。用于发送实时命令到采集端，控制采集端设备的编码质量、 闪关灯开关、 屏幕旋转、PTZ 动作等， 查询采集端设备的 sd 卡、时间等信息。

3.69 设置 IRvsViewerCmd 类的回调处理接口

方法： RVS_INT SetRvsCmdEvent(IRvsViewerCmdEvent* pICmdEvent) = 0

说明： 观看端通过 IRvsViewerCmd 类发出操作信令后，SDK 调用 IRvsViewerCmdEvent 中的回调方法来反馈相应的操作结果。具体见 IRvsViewerCmdEvent 的定义。

3.70 设置采集端的登录名和密码

方法： RVS_ULL ChangeStreamerLoginUserPwd(RVS_ULL ullStreamerCID,
RVS_UC* pucNewUser,
RVS_UC* pucNewPwd)

说明： 设置采集端被其他观看端连接时进行校验的用户名密码。

3.71 改变采集端视频编码质量

方法： RVS_ULL ChangeStreamerEncoderQuality(RVS_ULL ullStreamerCID,
RVS_INT iCamIndex,
RVS_INT iStreamIndex,
RVS_INT iBitrate,
RVS_INT iFramerate,
RVS_INT iFrameInterval,
RVS_INT iQuality)

说明： 改变采集端视频编码质量。

3.72 强制采集端编码一个 I 帧

方法： RVS_ULL ForceIFrame(RVS_ULL ullStreamerCID,

RVS_INT iCamIndex,
RVS_INT iStreamIndex)

说明：强制采集端编码一个 I 帧。

3.73 镜头翻转

方法： RVS_ULL CameraRotate(RVS_ULL ullStreamerCID, RVS_INT iCamIndex, RVS_UI uiRotateType)

说明：旋转镜头。

3.74 前后摄像头切换

方法： virtual RVS_ULL FrontRearCamSwitch(RVS_ULL ullStreamerCID, RVS_INT iCamIndex);

说明：对于手机采集端，前后摄像头切换。

3.75 开关闪光灯

方法： RVS_ULL TorchSwitch(RVS_ULL ullStreamerCID, RVS_INT iCamIndex)

说明：闪光灯开/关切换。

3.76 切换前后镜头

方法： RVS_ULL StreamSwitch(RVS_ULL ullStreamerCID, RVS_INT iCamIndex)

说明：切换前后镜头。

3.77 设置采集端时间时区信息

方法： RVS_ULL SetStreamerTimeZone(RVS_ULL ullStreamerCID,
RVS_UC* pucTime,
RVS_INT iTimeZone,
RVS_BOOL bSyncFlag)

说明：设置采集端时间、时区信息。

参数：

iSyncFlag: 是否同步，0 表示不同步。

iTimeZone: 时区值，单位是秒。正数表示东时区，如西区为 $-1*3600=-3600$ 。

pucTime: 时间字符串，格式是 YYYY-MM-DD-HH:mm，精确到分钟。

返回值：消息 id，唯一标识消息，从 1 开始递增。

3.78 格式化采集端的 SD 卡

方法： RVS_ULL FormatStreamerSDCard(RVS_ULL ullStreamerCID)

说明：格式化采集端的 SD 卡。

3.79 控制采集端 PTZ 或位移运动

方法：RVS_ULL PTZMoveCtrl(RVS_ULL ullStreamerCID,
RVS_INT iCamIndex,
RVS_UI uiType, /** 0x00 PTZ Ctrl; 0x01 Move Ctrl; */
RVS_INT iPRDirect, /** P or X Direct */
RVS_INT iTVDirect, /** T or Y Direct */
RVS_INT iZFDirect) = 0 ; /**Z or Z Direct */

说明：控制采集端进行云台的 PTZ 运动或者沿着 XYZ 轴运动。

参数：

uiType：控制类型值，0x01 表示设备移动；0x00 表示 PTZ 控制。

iXValue：为 P 值时，顺时针为正；为 X 值时，向右为正。

iYValue：为 T 值时，顺时针为正；为 Y 值时，向上为正。

iZValue：为 Z 值时，放大为正；为 Z 值时，向前为正。

3.80 获取采集端的时间时区信息

方法：virtual RVS_ULL RequestStreamerTimeZone(RVS_ULL ullStreamerCID)；

说明：获取采集端的时间、时区信息。

获取结果将在回调接口 IRvsViewerCmdEvent 的方法 OnCmdTimeGetResult 中反馈。

3.81 获取采集端的 sd 卡容量信息

方法：RVS_ULL RequestStreamerSDCardInfo(RVS_ULL ullStreamerCID)

说明：获取采集端的 sd 卡容量信息。获取结果将在回调接口 IRvsViewerCmdEvent.
OnCmdSdCardInfoResult 中反馈。

3.82 发送自定义信令

方法：RVS_INT SendCustomData(RVS_ULL ullStreamerCID,
RVS_UC* pucDataBuffer,
RVS_UC* pucBufLen)

说明：发送自定义的信令给采集端。用户可以通过回调接口 CustomDataRecvCallback 来收取来自采集端的自定义信令数据。

3.83 ST_RVS_MEDIA_AVDESC 结构体

此结构体用于描述逆向媒体码流的数据属性，具有以下方法：

```
typedef struct stRVS_MEDIA_AVDESC
{
    EN_RVS_VIDEO_TYPE          enVideoType; //视频格式类型
    RVS_INT                    iWidth; //视频分辨率之宽度。
    RVS_INT                    iHeight; //视频分辨率之高度。

    EN_RVS_AUDIO_TYPE         enAudioType; //音频格式类型。

    RVS_UI                     uiSampleRate; //音频采样率。

    RVS_UI                     uiChannel; //音频声道数。
    RVS_UI                     uiDepth; //音频数据位深，16位还是8位。
}ST_RVS_MEDIA_AVDESC;
```

3.84 ST_RVS_STREAMER_INFO 类

ST_RVS_STREAMER_INFO 描述了采集端的基本配置信息。有如下获取接口：

```
typedef struct stRVS_STREAMER_INFO{
    RVS_INT                    enOSType; //采集端类型，
    RVS_UC                     ucOSVersion[32]; //获取采集端操作系统版本。
    RVS_UC                     ucSDKVersion[32]; //获取采集端 SDK 版本。
    RVS_UC                     ucAppVersion[32]; //获取采集端应用版本
    RVS_INT                    iLanguage; //获取采集端语言标识。取值参见枚举 RvsLanguage 的定
    RVS_BOOL                   bPushFlag; //报警时是否发送 push 到观看端
    RVS_BOOL                   bEmailFlag; //报警时，平台是否发送邮件到指定邮箱
    RVS_BOOL                   bNoticeFlag; //报警时，是否有报警通知如响铃
    RVS_INT                    iSetFlag;
    RVS_UI                     uiRunMode;
    /** 0x00 Manual Run; 0x01 Auto Run; 0x02 Background Run; 0x04 Suspend Run; */
    RVS_UI                     uiRecordMode;
    /** 0x00 No Record; 0x01 SDCard Record; 0x02 Storage Record; */
    RVS_UI                     uiTimeZoneMode;
    /** 0x00 Not Support set timezone; 0x01 Support. */
    RVS_UI                     uiEchoCancelFlag;
    /** 0x00 Not support echo cancel, 0x01 Support. */
    RVS_UC                     ucEmailAddr[128];
    RVS_UI                     uiDelDays;
```

```

        RVS_INT                iCamCount;
        RVS_INT                iMicCount;
        RVS_UC                 ucDeviceName[32];
        ST_RVS_STREAMER_MICINFO*  pstMicInfos;//麦克风描述信息。
        ST_RVS_STREAMER_CAMERAINFO*  pstCameras;//获取镜头描述信息。

}ST_RVS_STREAMER_INFO;

typedef struct stRVS_STREAMER_CAMERAINFO{
    RVS_INT                iCamIndex;
    RVS_UC                 ucCameraName[32];
    RVS_UI                 uiPTZMoveEnable; /** 0x00 not support; 0x01 P
control; 0x02 T control; 0x04 Z control; 0x08 Move X control; 0x10 Move Y control; 0x20 Move
Z control*/
    RVS_UI                 uiTorchEnable;
    RVS_UI                 uiRotateEnable;
    RVS_UI                 uiStreamType; /** 0x00 One stream can used at one
time; 0x01 Multi stream can used at the same time. */
    RVS_INT                iStreamCount;
    ST_RVS_STREAMER_CAMERASTREAMINFO*  pstCameraStreams;
}ST_RVS_STREAMER_CAMERAINFO;

typedef struct stRVS_STREAMER_CAMERASTREAMINFO{
    RVS_INT                iStreamIndex;
    RVS_INT                iFramerate;
    RVS_INT                iBitrate;
    RVS_INT                iFrameInterval;
    RVS_INT                iQuality;
    RVS_INT                iWidth;
    RVS_INT                iHeight;
    RVS_UI                 uiEncodeType; /** EN_RVS_VIDEO_ENCODETYPE */
}ST_RVS_STREAMER_CAMERASTREAMINFO;

typedef struct stRVS_STREAMER_MICINFO{
    RVS_UI                 uiSamplerate;
    RVS_UI                 uiChannel;
    RVS_UI                 uiDepth;
    RVS_UI                 uiEncodeType; /** EN_RVS_AUDIO_ENCODETYPE */
}ST_RVS_STREAMER_MICINFO;

```

3.85 ST_RVS_STREAMER_SUPPORTSERVICES 结构

此类描述采集端支持的业务功能：

```
typedef struct stRVS_STREAMER_SUPPORTSERVICES {
    RVS_UI          uiSupportBCloud; //是否支持录制视频云存储, 1 支持, 0 不支持
    RVS_UI          uiSupportBTimeRecord; //是否支持定时录制。
    RVS_UI          uiSupportBAlarmRecord; //采集端是否支持报警录制
    RVS_UI          uiSupportBShortRecord; //是否支持短视频录制
    RVS_UI          uiSupportBSnapShort; //是否支持截图。
    RVS_UI          uiSupportBRebackTalk; //是否支持逆向语音。
    RVS_UI          uiSupportBRealIcon; //是否支持实时图标
    RVS_UI          uiSupportTorch;
    RVS_UI          uiSupportBWLAN;
    /**0x00 NOT support; 0x01 Support Command setting; 0x02 Support Sound setting; 0x04 Support
    Broadcast setting; 0x08 Support AP setting; 0x10 Support QRCode setting; */
    RVS_UI          uiSupportBSensors; //是否支持传感器
}ST_RVS_STREAMER_SUPPORTSERVICES;
```

3.86 ST_RVS_STREAMER_TIMERECORDINFO 类

此类描述采集端的定时录制设置信息，有如下获取方法：

```
typedef struct stRVS_STREAMER_TIMERECORDINFO {
    RVS_INT          iCamIndex;
    RVS_INT          iSetFlag; /** 0.Not Set; 1. Have set; 2. Have close */
    RVS_UI          uiScheduleCount;
    ST_RVS_TIMERECORD_SETTING*
    pstScheduleSettings;
}ST_RVS_STREAMER_TIMERECORDINFO;
```

```
typedef struct stRvsTimeRecordSetting
{
    ST_RVS_SCHEDULE_SETTING schedule;
    RVS_UI          uiParam;
}ST_RVS_TIMERECORD_SETTING;
```

```
typedef struct stRvsScheduleSetting
{
    RVS_UI          uiSeq;          //定时设置序号;
    RVS_BOOL        bEnable;       //是否开启
    RVS_UI          uiStartPoint;  //开始时间, 换算成秒计算;
    RVS_UI          uiEndPoint;    //结束时间, 换算成秒;
    RVS_UI          uiWeekFlag;    //周一到周日开启标志位;
}ST_RVS_SCHEDULE_SETTING;
```

3.87 ST_RVS_STREAMER_ALARMRECORDINFO 类

此类描述采集端的定时移动侦测功能的时间计划设置信息:

```
typedef struct stRVS_STREAMER_ALARMRECORDINFO {
    RVS_INT          iCamIndex;
    RVS_UI           uiScheduleCount;
    ST_RVS_MOTION_DETECT_SETTING* pstScheduleSettings;
}ST_RVS_STREAMER_ALARMRECORDINFO;
```

- ST_RVS_STREAMER_SENSORINFO 类

此类描述采集端的传感器的配置信息:

```
typedef struct stRVS_STREAMER_SENSORINFO {
    RVS_INT          iSensorIndex;
    RVS_INT          iSensorType;
    RVS_UC           ucSensorName[128];
    RVS_UI           uiScheduleCount;
    ST_RVS_SENSORS_DETECT_SETTING* pstScheduleSettings;
}ST_RVS_STREAMER_SENSORINFO;
```

3.88 回调接口 IRvsViewerEvent

定义:

```
class _RVS_API IRvsViewerEvent
{
public:

    //观看端本身连接到服务器状态通知，具体参见 EN_RVS_LOGIN_STATE 定义;
    virtual RVS_VOID OnLoginResult(EN_RVS_LOGIN_STATE enLoginState,
        RVS_INT uiProgressRate,
        EN_RVS_LOGIN_ERR enErrCode) = 0;

    virtual RVS_VOID OnUpdateCID(RVS_ULL ullLocalCID) = 0;

    //连接采集端的状态通知，具体分为三种通知，
    //1、网络连接通知 2、在线状态通知 3、配置获取状态通知;

    //1、网络连接通知
```

```

    virtual RVS_VOID OnStreamerConnectStatus (RVS_ULL ullStreamerCID,
    EN_RVS_STREAMER_CONN_STATE enConnectState) = 0;

    // 2、在线状态通知
    virtual RVS_VOID OnStreamerPresenceStatus (RVS_ULL ullStreamerCID,
    EN_RVS_STREAMER_PRESENCE_STATE enPresenceState) = 0;

    //3、配置获取状态通知;
    virtual RVS_VOID OnStreamerConfigStatus (RVS_ULL ullStreamerCID,
    EN_RVS_STREAMER_CONF_STATE enConfigState) = 0;

    //局域网搜索结果通知;
    virtual RVS_VOID OnLanSearchStreamerResult ( RVS_ULL ullStreamerCID,
    RVS_UC* pucStreamerName,
    RVS_UI uiOSType) = 0 ;
};

```

3.89 回调接口 IRvsViewerCmdEvent

定义:

```

class _RVS_API IRvsViewerCmdEvent
{
public:
    //收到自定义数据通知;
    virtual RVS_VOID OnCustomDataRecv (RVS_ULL ullPeerCID, RVS_UC* pucDataBuffer, RVS_UI
    uiBufLen) = 0 ;

    //通用操作结果通知;
    virtual RVS_VOID OnRequestStatus (RVS_ULL ullRequestID, RVS_BOOL bStatus) = 0;

    //请求采集端时间响应消息;
    virtual RVS_VOID OnCmdTimeGetResult (RVS_ULL ullRequestID, RVS_INT iErrorCode, RVS_UC*
    pucTime, RVS_INT iTimeZone, RVS_BOOL bSyncFlag) = 0;

    //请求采集端 sd 卡信息响应消息;
    virtual RVS_VOID OnCmdSdCardInfoResult (RVS_ULL ullRequestID, RVS_INT iErrorCode, RVS_ULL
    ullTotalSize, RVS_ULL ullRemainSize) = 0 ;
};

```

3.90 回调接口 IRvsViewerMediaEvent

定义:

```
class _RVS_API IRvsViewerMediaEvent
{
public:
//媒体流播放基本回调接口
//视频播放结果通知;
virtual RVS_VOID OnPlayStreamStatus(RVS_HANDLE hHandle,
                                     EN_RVS_MEDIASTREAM_STATE enStreamState,
                                     EN_RVS_MEDIASTREAM_FLAG enStreamFlag) = 0;

//本地录制视频状态的, 录制视频超过 10 分钟的时候通知;
virtual RVS_VOID OnLocalRecordStatus(RVS_HANDLE hHandle,
                                      EN_VIEWER_LOCAL_RECORD_ERRID enErrId) = 0;

//jpeg 数据回调;
virtual RVS_VOID OnRecvJpeg(RVS_ULL ullRequestID,
                             RVS_UC* pucJpegBuffer,
                             RVS_UI uiBufLen) = 0;

//录制视频列表通知;
virtual RVS_VOID OnRequestRecordFiles( RVS_ULL ullRequestID,
                                       RVS_UI uiErrorCode,
                                       RVS_UI uiTotalCount,
                                       RVS_UI uiCurrentCount,
                                       ST_RVS_RECORD_FILE_INFO stFileInfo[]) = 0;

virtual RVS_VOID OnRecordFileDeleteStatus(RVS_ULL ullRequestID, RVS_BOOL bStatus) = 0;

//获取一个月每天的云视频个数列表;
virtual RVS_VOID OnRequestCloudFilesPerDay( RVS_ULL llidStreamerCID,
                                             RVS_UI uiErrorCode,
                                             RVS_UI uiCountPerDay[],
                                             RVS_UI puiDayCount,
                                             RVS_UC* pucStartDay) = 0;

//获取每天云视频列表;
virtual RVS_VOID OnRequestCloudFiles( RVS_UI uiRequestID,
```

```

RVS_UI uiTotalCount,
RVS_UI uiCurrentCount,
ST_RVS_CLOUD_FILE_INFO stFileInfo[]) = 0;

//获取云视频图标;
virtual RVS_VOID OnRecvCloudFileIcon( RVS_ULL ullStreamerCID,
RVS_UC* pucEid,
RVS_UC* pucBuffer,
RVS_UI uiBufLen) = 0;
////////////////////////////////////
////////////////////////////////////

//逆向音视频状态通知;
virtual RVS_VOID OnReverseStreamStatus( RVS_ULL ullStreamerCID,
EN_RVS_MEDIASTREAM_STATE enStreamState,
EN_RVS_MEDIASTREAM_FLAG enStreamFlag) = 0;

//逆向音视频数据通知;
virtual RVS_VOID OnReverseStreamPutData( RVS_ULL ullStreamerCID,
EN_RVS_REV_STREAM_TYPE* penStreamType,
RVS_UC** ppucData,
RVS_UI* puiLen,
RVS_UI* puiTimestamp) = 0;
////////////////////////////////////
////////////////////////////////////
};

```

3.91 回调接口 IRvsViewerStreamerInfoEvent

定义:

```

class _RVS_API IRvsViewerStreamerInfoEvent
{

public:
//采集端信息变更通知; 具体信息变更的类型, 请参加 EN_RVS_STREAMER_INFO_TYPE 定义;
virtual RVS_VOID OnStreamerInfoChanged( RVS_ULL ullPeerCID, EN_RVS_STREAMER_INFO_TYPE
enStreamerCfgType ) = 0 ;

};

```

4. 使用步骤

- 工程配置

1.导入相关架包和库文件:

```
├─include
│   └─rvs_senv
│       └─rvs_viewer
├─lib
├─out
├─project
└─src
    └─rvs_senv
        └─rvs_viewer
```



目录结构.txt

5. SDK 调用流程

5.1 简要流程

1、对 IRvsViewer、IRvsViewerStreamerInfo、IRvsViewerMedia、IRvsViewerCmd 四个类的实例进行初始化。

2、实现 IRvsViewer、IRvsViewerStreamerInfo、IRvsViewerMedia、IRvsViewerCmd 四个接口对应的回调通知基类的方法，创建实现类的实例，并通过类的设置事件通知类方法设置事件通知类。

3、View 登录 Login() 。

4、登录成功以后调用 ConnectStreamer，连接采集端，获取采集端状态和配置信息；

5、对于在线的采集端，调用 IRvsViewerMedia 接口中的 OpenLiveStream 方法请求一路码流。

6、通过 IRvsViewerMedia 接口 GetVideoYUVData，循环获取 YUV 数据，渲染显示到窗口；

5.2 详细流程

1.初始化:

- a、调用 RvsQueryInterface 接口查询方法，查询各个接口并初始化;
- b、实现各个模块对应的事件通知类，并实例化，调用设置事件类接口进行设置.

```
RvsQueryInterface(IRvsViewer::IID, (RVS_VOID**) & m_pIRvsViewer);  
SetRvsViewerEvent
```

```
RvsQueryInterface(IRvsViewerCmd::IID, (RVS_VOID**) &m_pIRvsViewerCmd);  
SetRvsCmdEvent
```

```
RvsQueryInterface(IRvsViewerMedia::IID, (RVS_VOID**) &m_pIRvsViewerMedia);  
SetViewerMediaEvent
```

```
RvsQueryInterface(IRvsViewerStreamerInfo::IID, (RVS_VOID**)  
&m_pIRvsViewerStreamInfo);  
SetViewerStreamerInfoEvent
```

2.viewer 登录:

```
m_pIRvsViewer->Init  
m_pIRvsViewer->SetLoginInfo  
m_pIRvsViewer->Login();
```

3. 登录成功，用户添加采集端后，连接采集端[streamer]的 CID 号:

```
m_pIRvsViewer->ConnectStreamer
```

4.连接成功，获取指定采集端的能力信息，观看端可以根据此信息来动态生成 UI 操作界面[设备具有哪些能力，UI 操作界面就出现哪些项]:

```
m_pIRvsViewerStreamInfo->GetStreamerInfo
```

5.获取码流并显示:

```
m_pIRvsViewerMedia->OpenLiveStream  
m_pIRvsViewerMedia->GetStreamDesc  
m_pIRvsViewerMedia->GetVideoYUVData
```

6.用户发出控制采集端的操作:

如:

```
m_pIRvsViewerCmd->PTZMoveCtrl
```

7.退出并销毁

```
m_pIRvsViewer->Release
```

```
m_pIRvsViewerCmd->Release
```

```
m_pIRvsViewerStreamInfo->Release
```

```
m_pIRvsViewerMedia->Release
```