**NPPluginSDK开发文档**

修改记录

|  |  |  |
| --- | --- | --- |
| 时间 | 修改人 | 内容 |
| 2013-09-11 | 曾恒旭 | 创建 |
| 2013-09-27 | 曾恒旭 | 增加窗口事件信息结构，增加本地录像，本地抓拍，获取平台名称方法。修改事件信息结构 |
| 2013-10-15 | 曾恒旭 | 增加窗口事件结构，增加窗口右键自定义功能函数，添加右键菜单项，更新右键菜单项，剔除窗口右键菜单项 |
| 2013-10-22 | 胡召伟 | 更新创建窗口函数，增加窗口类型枚举等等 |
| 2013-11-06 | 胡召伟 | 增加查询中心存储录像或图片函数，中心存储文件信息结构，点播回放函数接口，存储文件类型枚举，开启停止GPS流函数，完善NC事件枚举说明 |
| 2013-11-11 | 胡召伟 | 增加前端CEFS系统存储查询相关函数接口、增加CEFS录像、图片、日志、报警、GPS等信息结构、增加CEFS录像、图片原因，日志、报警类型等枚举对象 |
| 2013-11-18 | 胡召伟 | 增加获取逻辑分组、节点、资源等接口及与之对应的信息结构，设备喊话与对讲相关接口、资源描述配置函数以及资源描述信息结构 |
| 2013-11-27 | 胡召伟 | 调整创建窗口接口说明、补增窗口类型枚举、增加平台录像图片下载接口、前端录像、前端图片下载接口以及相对应的停止下载接口 |
| 2014-01-02 | 胡召伟 | 增加配置相关接口、控制相关接口，请求响应公共接口 |
| 2014-01-07 | 胡召伟 | 增加一些独立的配置或控制功能接口，如设备管理、视频设置、报警设置等 |
| 2014-01-22 | 胡召伟 | 增加云台控制、窗口模式获取与设置、视频窗口掩码获取设置等接口 |
| 2014-02-11 | 胡召伟 | 修正获取、设置视频编码分辨率函数增加流类型参数，增加获取或设置是否使能窗口字符叠加接口，向视频窗口叠加字符接口，删除叠加的文字信息接口，叠加字符信息结构等 |
| 2014-03-04 | 胡召伟 | 增加中心GPS数据存储与查询等相关接口说明 |
| 2014-04-30 | 胡召伟 | 增加设备前端CEFS格式录像与图片查询函数 |
| 2014-05-21 | 胡召伟 | 导出获取所有正在喊话和对讲的状态信息 |
| 2014-06-03 | 胡召伟 | 外接磁盘查询下载相关的几个函数接口 |
| 2014-12-26 | 胡召伟 | 新增创建会议室、申请加入会议室、关闭或退出会议室、查询所有会议室信息、修改会议室密码、从会议室中删除一个用户、从会议室中删除一个设备、添加一个设备入会议室  修改连接对象信息结构增加CUID属性（用户连接ID）  新增获取用户连接ID（CUID）接口（只适应与当前建立的连接） |
| 2015-02-06 | 胡召伟 | 新增获取接收发送总流量接口 |
| 2016-12-14 | 胡召伟 | 修改磁盘CEFS查询、下载、点播等接口，增加盘符属性 |
| 2017-08-29 | 胡召伟 | 增加报警事件处警功能接口 |
| 2019-11-21 | 胡召伟 | 增加FAT32格式前端存储文件查询和下载接口 |
| 2019-11-25 | 胡召伟 | 增加FAT32格式前端存储录像点播功能 |

1. **目录结构**

目录

[1 目录结构 1](#_Toc19713)

[目录 1](#_Toc24272)

[2 插件 1](#_Toc12193)

[2.1 插件包 1](#_Toc6818)

[2.1.1 MediaPlugin7.exe 1](#_Toc12463)

[3 接口方法 1](#_Toc23303)

[3.1 NPPInterlayer.js脚本包含和依赖 1](#_Toc373)

[3.2 NPPInterlayer对象 1](#_Toc7416)

[3.3 NPPInterlayer自定义结构类型 1](#_Toc24694)

[3.3.1 初始化NPPILY对象参数结构NPPILY.Struct.InitParamStruct 1](#_Toc19611)

[3.3.2 调试信息结构NPPILY.Struct.DebugMessageStruct 1](#_Toc14917)

[3.3.3 返回值结构 NPPILY.Struct.ReturnValueStruct 1](#_Toc6441)

[3.3.4 连接服务器或直连设备参数结构NPPILY.Struct.ConnParamStruct 1](#_Toc2628)

[3.3.5 连接对象信息结构NPPILY.Struct.ConnectionStruct 1](#_Toc15692)

[3.3.6 窗口信息结构NPPILY.Struct.WindowStruct 1](#_Toc16924)

[3.3.7 窗口容器信息结构 NPPILY.Struct.WindowContainerStruct 1](#_Toc1823)

[3.3.8 PU资源节点结构NPPILY.Struct.PUNodeStruct 1](#_Toc30253)

[3.3.9 PU子资源节点结构NPPILY.Struct.PUResourceNodeStruct 1](#_Toc1395)

[3.3.10 事件信息结构 NPPILY.Struct.NCObjectNotifyStruct 1](#_Toc15942)

[3.3.11 窗口事件信息结构NPPILY.Struct. WindowEventStruct 1](#_Toc11276)

[6.3.11 中心存储文件信息结构NPPILY.Struct. SCIVDateFileStruct **错误!未定义书签。**](#_Toc30058)

[6.3.12 前端CEFS录像文件信息结构NPPILY.Struct.CEFSRecordStruct 1](#_Toc3824)

[6.3.13 前端CEFS图片文件信息结构NPPILY.Struct.CEFSPictureStruct 1](#_Toc6253)

[6.3.14 前端CEFS用户日志信息结构NPPILY.Struct.CEFSUserLogStruct 1](#_Toc5373)

[6.3.15 前端CEFS报警事件信息结构NPPILY.Struct.CEFSAlarmEventStruct 1](#_Toc17058)

[6.3.16 前端GPS数据信息结构NPPILY.Struct.CEFSGPSDataStruct 1](#_Toc17507)

[6.3.17 逻辑分组信息结构NPPILY.Struct.LogicGroupStruct 1](#_Toc258)

[6.3.18 逻辑分组节点信息结构NPPILY.Struct.LogicGroupNodeStruct 1](#_Toc8114)

[6.3.19 逻辑分组节点下资源信息结构NPPILY.Struct.LogicGroupResourceStruct 1](#_Toc14185)

[6.3.20 资源描述信息结构NPPILY.Struct.CommonResDescriptionStruct 1](#_Toc159)

[6.3.21 视频窗口叠加文字信息结构NPPILY.Struct.WindowTextAddStruct 1](#_Toc28895)

[6.3.22 GPS数据存储资源结构NPPILY.Struct.GPSDataStorageResourceStruct 1](#_Toc1453)

[6.3.23 中心存储GPS数据文件信息结构NPPILY.Struct.CSUGPSDataFileStruct 1](#_Toc31495)

[6.3.24 网络文件下载状态结构NPPILY.Struct.HttpDownloadStatusStruct 1](#_Toc4699)

[6.4 NPPILY自定义枚举类型 1](#_Toc28202)

[6.4.1 NPPILY视频流类型NPPILY.Enum.NrcapStreamType 1](#_Toc7561)

[6.4.2 NPPILY视频窗口类型NPPILY.Enum.WindowType 1](#_Toc3706)

[6.4.3 NPPILY云台控制命令NPPILY.Enum.PTZDirection 1](#_Toc10283)

[6.4.4 NPPILY浏览器类型 NPPILY.Enum.BrowserType 1](#_Toc6268)

[6.4.5 NPPILY支持的语言类型 NPPILY.Enum.LanguageType 1](#_Toc18061)

[6.4.6 NPPILY建立的连接类型 NPPILY.Enum.ConnectionType 1](#_Toc22974)

[6.4.7 NPPILY建立的连接状态NPPILY.Enum.ConnectionStatus 1](#_Toc14511)

[6.4.8 NPPILY获取或构建资源级别 NPPILY.Enum.ForkResourceLevel 1](#_Toc16450)

[6.4.9 NPPILY获取的设备类型NPPILY.Enum.PuModelType 1](#_Toc12929)

[6.4.10 NPPILY获取的资源类型NPPILY.Enum.PuResourceType 1](#_Toc7755)

[6.4.11 NPPILY NC事件对象NPPILY.Enum.NCObjectNotify 1](#_Toc15114)

[6.4.12 NPPILY 存储文件类型NPPILY.Enum.StorageFileType 1](#_Toc3428)

[6.4.13 NPPILY CEFS查询命令NPPILY.Enum.CEFSRequestID 1](#_Toc448)

[6.4.14 NPPILY CEFS录像原因类型NPPILY.Enum.CEFSRecordReason 1](#_Toc29408)

[6.4.15 NPPILY CEFS抓拍原因类型NPPILY.Enum.CEFSPictureReason 1](#_Toc17302)

[6.4.16 NPPILY CEFS用户日志类型NPPILY.Enum.CEFSUserLogType 1](#_Toc31013)

[6.4.17 NPPILY CEFS报警事件类型NPPILY.Enum.CEFSAlarmEventType 1](#_Toc25999)

[6.5 NPPILY全局变量 1](#_Toc11391)

[6.6 NPPILY对象类型 1](#_Toc22743)

[6.7 NPPILY方法（公开） 1](#_Toc8011)

[6.7.1 初始化NPPILY.Init(NPPILY.Struct.InitParamStruct initParam) 1](#_Toc15533)

[6.7.2 卸载 NPPILY.UnLoad() 1](#_Toc22712)

[6.7.3 创建连接 NPPILY.Connect(NPPILY.Struct.ConnParamStruct connParams) 1](#_Toc5293)

[6.7.4 断开连接 NPPILY.DisConnection(String connectId) 1](#_Toc10879)

[6.7.5 获取或构建设备（子）资源 NPPILY. ForkResource(String connectId, NPPILY.Enum.ForkResourceLevel forkLevel, uint offset, uint count, String domainRoad, object customParams) 1](#_Toc24108)

[6.7.6 创建视频窗口对象 NPPILY. CreateWindow(String connectId, DOMObject/String containerOrId, NPPILY.Enum.WindowType windowType, NPPILY.Struct.WindowEventStruct windowEvent) 1](#_Toc11563)

[6.7.7 改变视频窗口插件宽高 NPPILY. ResizeWindowDimension(NPPILY.Struct.WindowStruct objWnd, variant width, variant height) 1](#_Toc26061)

[6.7.8 播放视频 NPPILY.PlayVideo(String connectId, NPPILY.Struct.WindowStruct window, String puid, Uint idx, NPPILY.Enum.NrcapStreamType streamType, object customParams) 1](#_Toc24566)

[6.7.9 播放转码流视频 NPPILY.PlayTranscodeVideo(String connectId, NPPILY.Struct.WindowStruct window, String puid, Uint idx, uint alg,string resolution, uint bitRate, uint frameRate) 1](#_Toc1662)

[6.7.10 停止视频 NPPILY.StopVideo(NPPILY.Struct.WindowStruct window) 1](#_Toc4169)

[6.7.11 播放/停止音频 NPPILY.PlayAudio(NPPILY.Struct.WindowStruct window) 1](#_Toc1393)

[6.7.12 绑定事件 NPPILY. NCNotifyManager.Add(NPPILY.Enum.NCObjectNotify eventName, Function callback) 1](#_Toc3665)

[6.7.13 移除绑定事件NPPILY. NCNotifyManager. Remove (NPPILY.Enum.NCObjectNotify eventName) 1](#_Toc6867)

[6.7.14 更新绑定事件NPPILY. NCNotifyManager. Update (NPPILY.Enum.NCObjectNotify eventName Function callback) 1](#_Toc8026)

[6.7.15 获取平台名称 NPPILY.GetSystemName(String connectId) 1](#_Toc30031)

[6.7.16 获取或设置是否使能窗口字符叠加 NPPILY.EnableAddText(NPPILY.Struct.WindowStruct objWnd, variant bValue) 1](#_Toc25538)

[6.7.17 向视频窗口叠加字符 NPPILY.SetTextAdd(NPPILY.Struct.WindowStruct objWnd, <Array|Object>NPPILY.Struct.WindowTextAddStruct textSets) 1](#_Toc13987)

[6.7.18 删除叠加的文字信息 NPPILY.DeleteTextAdd(NPPILY.Struct.WindowStruct objWnd[, uint wndBlockNo]) 1](#_Toc21956)

[6.7.19 开启/关闭本地录像NPPILY.LocalRecord(NPPILY.Struct.WindowStruct window, String savePath, String FileName) 1](#_Toc6600)

[6.7.20 本地抓图 NPPILY.Snapshot(NPPILY.Struct.WindowStruct window, String savePath, String FileName) 1](#_Toc28529)

[6.7.21 开启窗口全屏 NPPILY. FullScreen(NPPILY.Struct.WindowStruct window) 1](#_Toc14214)

[6.7.22 退出窗口全屏 NPPILY. ExitFullScreen(NPPILY.Struct.WindowStruct window) 1](#_Toc14730)

[6.7.23 添加窗口右键菜单项 NPPILY.WindowAttachEvent.AppendMenuCommand(NPPILY.Struct.WindowStruct window，Array customMenus) 1](#_Toc21022)

[6.7.24 更新窗口右键菜单项NPPILY.WindowAttachEvent.UpdateMenuCommand (NPPILY.Struct.WindowStruct window, Array customMenus, Number bRedraw) 1](#_Toc15180)

[6.7.25 剔除窗口右键菜单项NPPILY.WindowAttachEvent.RemoveMenuCommand(NPPILY.Struct.WindowStruct window，Array customMenus) 1](#_Toc12563)

[6.7.26 云台控制NPPILY.PTZ.Control(string connectId, string puid, uint ptzIndex, NPPILY.Enum.PTZDirection direction, object options) 1](#_Toc17017)

[6.7.27 获取窗口控制模式NPPILY.WindowControlMode.get(NPPILY.Struct.WindowStruct objWnd) 1](#_Toc25095)

[6.7.28 设置窗口控制模式NPPILY.WindowControlMode.set(NPPILY.Struct.WindowStruct objWnd, uint value) 1](#_Toc17482)

[6.7.29 是否使能掩码NPPILY.EnableMask(NPPILY.Struct.WindowStruct objWnd, uint value) 1](#_Toc26157)

[6.7.30 点击掩码NPPILY.MaskClick(NPPILY.Struct.WindowStruct objWnd， uint x, uint y) 1](#_Toc17458)

[6.7.31 选择掩码区域NPPILY.MaskSelectRect(NPPILY.Struct.WindowStruct objWnd, uint bx, uint by, uint ex, uint ey) 1](#_Toc17932)

[6.7.32 获取视频窗口中的掩码NPPILY.GetMask(NPPILY.Struct.WindowStruct objWnd) 1](#_Toc30706)

[6.7.33 设置视频窗口中的掩码NPPILY.SetMask(NPPILY.Struct.WindowStruct objWnd, Array maskList) 1](#_Toc23121)

[6.7.34 开启设备的GPS信息接收通道NPPILY.StartGPSStream(String connectId, String puid, uint gpsIndex) 1](#_Toc12208)

[6.7.35 停止设备的GPS信息接收通道NPPILY.StopGPSStream(String connectId, String puid, uint gpsIndex) 1](#_Toc19473)

[6.7.36 高级查询中心存储文件NPPILY.QueryCSUFiles(String connectId, Object queryConditions) 1](#_Toc11175)

[6.7.37 获取平台存储下有存储文件的日期NPPILY.FetchCSUIVDate(String connectId, Object queryConditions) 1](#_Toc4137)

[6.7.38 按日期获取视频的平台存储文件NPPILY.FetchCSUIVDateFiles(String connectId, Object queryConditions) 1](#_Toc13449)

[6.7.39 录像回放点播NPPILY.PlayVod(String connectId, NPPILY.Struct.WidnowStruct objWnd, Object options) 1](#_Toc32420)

[6.7.40 停止点播 NPPILY.StopVod(NPPILY.Struct.WindowStruct objWnd) 1](#_Toc13263)

[6.7.41 CEFS查询有存储文件的日期NPPILY.FetchCEFSDate(String connectId, Object queryConditions) 1](#_Toc28239)

[6.7.42 查询CEFS存储文件（适用快速查询和高级查询）NPPILY.FetchCEFSFiles(String connectId, Object queryConditions) 1](#_Toc13179)

[6.7.43 获取逻辑分组NPPILY.GetLogicGroups(string connectId) 1](#_Toc15839)

[6.7.44 获取逻辑分组子节点NPPILY.GetLogicGroupNodes(string connectId, object options) 1](#_Toc10537)

[6.7.45 获取逻辑分组节点下资源NPPILY.GetLogicGroupResource(string connectId, object options) 1](#_Toc16216)

[6.7.46 资源描述配置NPPILY.CommonResConfig(string connectId, string puid, NPPILY.Enum.PuResourceType resType, uint resIdx, string action, NPPILY.Struct.CommonResDescriptionStruct paramObj) 1](#_Toc20059)

[6.7.47 开启设备喊话NPPILY.CallTalkControl.StartCall(string connectId, string puid, uint oaIndex) 1](#_Toc24464)

[6.7.48 开启设备双向对讲NPPILY.CallTalkControl.StartTalk(string connectId, string puid, uint oaIndex) 1](#_Toc6240)

[6.7.49 获取设备的喊话或对讲状态NPPILY.CallTalkControl.GetStatus(string connectId, string puid, uint oaIndex) 1](#_Toc29457)

[6.7.50 停止设备的喊话或对讲NPPILY.CallTalkControl.Stop(string connectId, string puid, uint oaIndex) 1](#_Toc13391)

[6.7.51 下载中心存储录像或图片NPPILY.Download.StartCSUFileDownload(string connectId, string csuPuid, string fileAllPath, string localSaveAsFile, unit csuIndex) 1](#_Toc6808)

[6.7.52 停止下载平台存储录像或图片NPPILY.Download.StopCSUFileDownload(string connectId, String dlStreamHandle) 1](#_Toc20449)

[6.7.53 下载前端存储录像NPPILY.Download.StartCEFSVodFileDownload(string connectId, string puid, uint sgIndex, utctime beginTime, utctime endTime, uint ivIndex, string localSaveAsFile) 1](#_Toc1288)

[6.7.54 下载前端存储图片NPPILY.Download.StartCEFSImageFileDownload(string connectId, string puid, uint sgIndex, UTCTime snapshotTime, uint noInSecond, uint ivIndex, string localSaveAsFile) 1](#_Toc32317)

[6.7.55 停止下载前端存储录像或图片NPPILY.Download.StopCEFSFileDownload(string connectId, String dlStreamHandle) 1](#_Toc16785)

[6.7.56 按连接ID清除所有下载NPPILY.Download.Clear(string connectId) 1](#_Toc16362)

[6.7.57 获取存储GPS数据的资源NPPILY.GPSData.GetGPSResFromCSU(string connectId, object options) 1](#_Toc11342)

[6.7.58 添加存储GPS数据的资源NPPILY.GPSData.AddGPSResToCSU(string connectId, object options) 1](#_Toc12163)

[6.7.59 删除存储GPS数据的资源NPPILY.GPSData.RemoveGPSResFromCSU(string connectId, object options) 1](#_Toc4276)

[6.7.60 查询设备最新GPS数据NPPILY.GPSData.QueryLastGPSDatas(string connectId, object options) 1](#_Toc8420)

[6.7.61 查询设备历史GPS数据NPPILY.GPSData.QueryHistoryGPSDatas(string connectId, object options) 1](#_Toc14460)

[6.7.62 查询某GPS保存历史数据的日期NPPILY.GPSData.QueryGPSDates(string connectId, object options) 1](#_Toc12009)

[6.7.63 查询某GPS某日期的历史数据NPPILY.GPSData.QueryGPSDateFiles(string connectId, object options) 1](#_Toc4857)

[6.7.64 快速获取配置项NPPILY.Config.GetSimple(string connectId, string puid, NPPILY.Enum.PuResourceType resType, unit resIdx, string configID, NPPILY.Enum.NrcapStreamType streamType, object customParams) 1](#_Toc21491)

[6.7.65 快速设置配置项NPPILY.Config.SetSimple(string connectId, string puid, NPPILY.Enum.PuResourceType resType, unit resIdx, string configID, NPPILY.Enum.NrcapStreamType streamType, variant value, object customParams) 1](#_Toc16372)

[6.7.66 高级获取配置项NPPILY.Config.GetComplex(string connectId, object reqParams) 1](#_Toc19411)

[6.7.67 高级设置配置项NPPILY.Config.GetComplex(string connectId, object setParams) 1](#_Toc25215)

[6.7.68 发送获取控制命令项NPPILY.Control.CommonGet(string connectId, object reqParams) 1](#_Toc12747)

[6.7.69 发送设置控制命令项NPPILY.Control.CommonSet(string connectId, object reqParams) 1](#_Toc873)

[6.7.70 多命令联合请求公共接口NPPILY.CommonRequest(string connectId, object options) 1](#_Toc9148)

[6.7.71 获取设备时区NPPILY.ST\_GetTZ(string connectId, string puid) 1](#_Toc18479)

[6.7.72 设置设备时区NPPILY.ST\_SetTZ(string connectId, string puid, timeZone) 1](#_Toc15301)

[6.7.73 手动重启设备NPPILY.ST\_Reboot(string connectId, string puid) 1](#_Toc10914)

[6.7.74 获取设备当前时间NPPILY.ST\_GetTime(string connectId, string puid) 1](#_Toc31711)

[6.7.75 设置设备当前时间NPPILY.ST\_SetTime(string connectId, string puid, string time) 1](#_Toc32413)

[6.7.76 获取设备是否在线可用NPPILY.GetResourceUsable(string connectId, string puid) 1](#_Toc25394)

[6.7.77 获取前端联动支持的事件NPPILY.ST\_GetSupportedEventSets(string connectId, string puid) 1](#_Toc15994)

[6.7.78 获取前端联动支持的动作NPPILY.ST\_GetSupportedActionSets(string connectId, string puid) 1](#_Toc25399)

[6.7.79 获取前端联动配置NPPILY.ST\_GetDeviceLinkActions(string connectId, string puid, string returnType) 1](#_Toc7214)

[6.7.80 设置前端联动配置NPPILY.ST\_SetDeviceLinkActions(string connectId, string puid, string xmlDevLinkActions) 1](#_Toc23279)

[6.7.81 摄像头状态NPPILY.IV\_GetCameraStatus(string connectId, string puid, uint ivIndex) 1](#_Toc1068)

[6.7.82 获取视频亮度NPPILY.IV\_GetBrightness(string connectId, string puid, uint ivIndex) 1](#_Toc12458)

[6.7.83 设置视频亮度NPPILY.IV\_SetBrightness(string connectId, string puid, uint ivIndex， uint value) 1](#_Toc26752)

[6.7.84 预览设置视频亮度NPPILY.IV\_PreviewBrightness(string connectId, string puid, uint ivIndex， uint value) 1](#_Toc10042)

[6.7.85 获取视频对比度NPPILY.IV\_GetContrast(string connectId, string puid, uint ivIndex) 1](#_Toc7629)

[6.7.86 设置视频对比度NPPILY.IV\_SetContrast(string connectId, string puid, uint ivIndex， uint value) 1](#_Toc12674)

[6.7.87 预览设置视频对比度NPPILY.IV\_PreviewContrast(string connectId, string puid, uint ivIndex， uint value) 1](#_Toc8454)

[6.7.88 获取视频色调NPPILY.IV\_GetHue(string connectId, string puid, uint ivIndex) 1](#_Toc24515)

[6.7.89 设置视频色调NPPILY.IV\_SetHue(string connectId, string puid, uint ivIndex， uint value) 1](#_Toc5146)

[6.7.90 预览设置视频色调NPPILY.IV\_PreviewHue(string connectId, string puid, uint ivIndex， uint value) 1](#_Toc7196)

[6.7.91 获取视频饱和度NPPILY.IV\_GetSaturation(string connectId, string puid, uint ivIndex) 1](#_Toc28915)

[6.7.92 设置视频饱和度NPPILY.IV\_SetSaturation(string connectId, string puid, uint ivIndex， uint value) 1](#_Toc22534)

[6.7.93 预览设置视频饱和度NPPILY.IV\_PreviewSaturation(string connectId, string puid, uint ivIndex， uint value) 1](#_Toc11636)

[6.7.94 获取支持的流类型NPPILY.IV\_GetSupportedStreamTypeSets(string connectId, string puid, uint ivIndex) 1](#_Toc6884)

[6.7.95 获取支持的编码分辨率NPPILY.IV\_GetSupportedResolutionSets(string connectId, string puid, uint ivIndex) 1](#_Toc8282)

[6.7.96 获取视频编码分辨率NPPILY.IV\_GetResolution(string connectId, string puid, uint ivIndex, NPPILY.Enum.NrcapStreamType streamType) 1](#_Toc8665)

[6.7.97 设置视频编码分辨率NPPILY.IV\_SetResolution(string connectId, string puid, uint ivIndex, NPPILY.Enum.NrcapStreamType streamType，string resolution) 1](#_Toc13865)

[6.7.98 获取支持的工作模式NPPILY.IDL\_GetSupportedWorkModeSets(string connectId, string puid, unit idlIndex) 1](#_Toc897)

[6.7.99 获取当前的工作模式NPPILY.IDL\_GetWorkMode(string connectId, string puid, unit idlIndex) 1](#_Toc16019)

[6.7.100 设置当前的工作模式NPPILY.IDL\_SetWorkMode(string connectId, string puid, unit idlIndex, string value) 1](#_Toc7745)

[6.7.101 支持的报警触发模式NPPILY.IDL\_GetSupportedAlertInModeSets(string connectId, string puid, unit idlIndex) 1](#_Toc7669)

[6.7.102 获取当前的报警触发模式NPPILY.IDL\_GetAlertInMode(string connectId, string puid, unit idlIndex) 1](#_Toc3660)

[6.7.103 设置当前的报警触发模式NPPILY.IDL\_SetAlertInMode(string connectId, string puid, unit idlIndex, value) 1](#_Toc19424)

[6.7.104 报警输入状态NPPILY.IDL\_GetAlertInStatus(string connectId, string puid, unit idlIndex) 1](#_Toc4234)

[6.7.105 获取开始计数时间NPPILY.IDL\_GetCountBeginTime(string connectId, string puid, unit idlIndex) 1](#_Toc3785)

[6.7.106 计数总值NPPILY.IDL\_GetCount(string connectId, string puid, unit idlIndex) 1](#_Toc8711)

[6.7.107 状态是否有效NPPILY.IDL\_GetLevelValid(string connectId, string puid, unit idlIndex) 1](#_Toc9808)

[6.7.108 读取报警布防时间表NPPILY.IDL\_GetGuardMap(string connectId, string puid, unit idlIndex) 1](#_Toc26519)

[6.7.109 设置报警布防时间表NPPILY.IDL\_SetGuardMap(string connectId, string puid, unit idlIndex, string mapValue) 1](#_Toc24778)

[6.7.110 获取报警输出默认状态NPPILY.ODL\_GetDefaultConnectStatus(string connectId, string puid, unit odlIndex) 1](#_Toc1996)

[6.7.111 获取报警输出当前状态NPPILY.ODL\_GetConnectStatus(string connectId, string puid, unit odlIndex) 1](#_Toc15844)

[6.7.112 获取接通动作别名NPPILY.ODL\_GetAliasConnect(string connectId, string puid, unit odlIndex) 1](#_Toc12662)

[6.7.113 设置接通动作别名NPPILY.ODL\_SetAliasConnect(string connectId, string puid, unit odlIndex, string aliasName) 1](#_Toc3997)

[6.7.114 获取断开动作别名NPPILY.ODL\_GetAliasBreak(string connectId, string puid, unit odlIndex) 1](#_Toc6979)

[6.7.115 设置断开动作别名NPPILY.ODL\_SetAliasBreak(string connectId, string puid, unit odlIndex, string aliasName) 1](#_Toc21489)

[6.7.116 控制报警输出状态NPPILY.ODL\_SetStatus(string connectId, string puid, unit odlIndex, uint value) 1](#_Toc27959)

[6.7.117 获取前端录像持续时间NPPILY.SG\_GetRecordTimeSpan(string connectId, string puid, unit sgIndex) 1](#_Toc5346)

[6.7.118 设置前端录像持续时间NPPILY.SG\_SetRecordTimeSpan(string connectId, string puid, unit sgIndex, uint value) 1](#_Toc19224)

[6.7.119 获取前端磁盘状态NPPILY.SG\_GetDiskStatus(string connectId, string puid, unit sgIndex) 1](#_Toc18885)

[6.7.120 获取前端磁盘空间不足时是否覆盖旧文件NPPILY.SG\_GetCoverOldRecordFile(string connectId, string puid, unit sgIndex) 1](#_Toc12031)

[6.7.121 设置前端磁盘空间不足时是否覆盖旧文件NPPILY.SG\_SetCoverOldRecordFile(string connectId, string puid, unit sgIndex, uint value) 1](#_Toc19454)

[6.7.122 获取前端磁盘空间不足时的剩余空间门限NPPILY.SG\_GetDiskInsufficientSpace(string connectId, string puid, unit sgIndex) 1](#_Toc32172)

[6.7.123 设置前端磁盘空间不足时的剩余空间门限NPPILY.SG\_SetDiskInsufficientSpace(string connectId, string puid, unit sgIndex, uint value) 1](#_Toc17037)

[6.7.124 获取前端录像文件保留天数NPPILY.SG\_GetRecordFileReserveDays(string connectId, string puid, unit sgIndex) 1](#_Toc19032)

[6.7.125 设置前端录像文件保留天数NPPILY.SG\_SetRecordFileReserveDays(string connectId, string puid, unit sgIndex, uint value) 1](#_Toc17182)

[6.7.126 获取前端文件系统类型NPPILY.SG\_GetFileSystemType(string connectId, string puid, unit sgIndex) 1](#_Toc11787)

[6.7.127 获取前端磁盘信息NPPILY.SG\_GetDiskInfo(string connectId, string puid, unit sgIndex) 1](#_Toc7287)

[6.7.128 开始前端初始化文件系统NPPILY.SG\_StartInitFileSystem(string connectId, string puid, unit sgIndex, string sgDiskLetter) 1](#_Toc30411)

[6.7.129 查询前端初始化文件系统进度NPPILY.SG\_QueryInitFileSystemProgress(string connectId, string puid, unit sgIndex, string sgDiskLetter) 1](#_Toc7545)

[6.7.130 手动启动前端存储NPPILY.SG\_ManualStart(string connectId, string puid, unit sgIndex, NPPILY.Enum.PuResourceType ivType, uint ivIndex, ivStreamType, uint duration) 1](#_Toc21791)

[6.7.131 手动停止前端存储NPPILY.SG\_ManualStop(string connectId, string puid, unit sgIndex, NPPILY.Enum.PuResourceType ivType, uint ivIndex, ivStreamType) 1](#_Toc1299)

[6.7.132 查询CEFS录像或图片NPPILY.SG\_CEFSQueryFiles(string connectId, string puid, unit sgIndex, unit ivIndex, UTC beginTime, UTC endTime, NPPILY.Enum.NrcapStreamType ivStreamType) 1](#_Toc8881)

[6.7.133 获取平台录像持续时间NPPILY.SC\_GetRecordTimeSpan(string connectId, string csuPuid, unit csuIndex) 1](#_Toc27349)

[6.7.134 设置平台录像持续时间NPPILY.SC\_SetRecordTimeSpan(string connectId, string csuPuid, unit csuIndex, uint value) 1](#_Toc29475)

[6.7.135 获取平台磁盘状态NPPILY.SC\_GetDiskStatus(string connectId, string csuPuid, unit csuIndex) 1](#_Toc11519)

[6.7.136 获取平台存储磁盘满时是否覆盖旧文件NPPILY.SC\_GetCoverOldRecordFile(string connectId, string csuPuid, unit csuIndex) 1](#_Toc26694)

[6.7.137 设置平台存储磁盘满时是否覆盖旧文件NPPILY.SC\_SetCoverOldRecordFile(string connectId, string csuPuid, unit csuIndex， uint value) 1](#_Toc6220)

[6.7.138 获取平台存储录像文件保存天数NPPILY.SC\_GetRecordFileReserveDays(string connectId, string csuPuid, unit csuIndex) 1](#_Toc19801)

[6.7.139 设置平台存储录像文件保存天数NPPILY.SC\_SetRecordFileReserveDays(string connectId, string csuPuid, unit csuIndex， uint value) 1](#_Toc24259)

[6.7.140 获取平台存储抓拍文件保存天数NPPILY.SC\_GetSnapshotReserveDays(string connectId, string csuPuid, unit csuIndex) 1](#_Toc22783)

[6.7.141 设置平台存储抓拍文件保存天数NPPILY.SC\_SetSnapshotReserveDays(string connectId, string csuPuid, unit csuIndex， uint value) 1](#_Toc24144)

[6.7.142 获取平台存储GPS数据保存天数NPPILY.SC\_GetGPSReserveDays(string connectId, string csuPuid, unit csuIndex) 1](#_Toc13312)

[6.7.143 设置平台存储GPS数据保存天数NPPILY.SC\_SetGPSReserveDays(string connectId, string csuPuid, unit csuIndex， uint value) 1](#_Toc31994)

[6.7.144 获取平台存储是否使能GPS存储NPPILY.SC\_GetEnableGPSDataStorage(string connectId, string csuPuid, unit csuIndex) 1](#_Toc4894)

[6.7.145 设置平台存储是否使能GPS存储NPPILY.SC\_SetEnableGPSDataStorage(string connectId, string csuPuid, unit csuIndex， uint value) 1](#_Toc20461)

[6.7.146 获取平台存储磁盘信息NPPILY.SC\_GetDiskInfo(string connectId, string csuPuid, unit csuIndex) 1](#_Toc25729)

[6.7.147 手动启动平台存储NPPILY.SC\_ManualStart(string connectId, string csuPuid, unit csuIndex, string rec\_puid, uint ivIndex, ivStreamType, string rec\_reason, uint rec\_duration) 1](#_Toc11187)

[6.7.148 手动停止平台存储NPPILY.SC\_ManualStop(string connectId, string csuPuid, unit csuIndex, string rec\_puid, uint ivIndex, ivStreamType) 1](#_Toc21891)

[6.8 调用说明 1](#_Toc28494)

[7 附录 1](#_Toc23529)

[7.1 错误码 1](#_Toc10518)

[7.2 辅助函数对象NPPUtils 1](#_Toc1477)

[7.2.1 NPPUtils.Hash哈希方法体结构 1](#_Toc27523)

[7.2.2 NPPUtils.Array数组辅助对象 1](#_Toc11675)

[7.2.3 NPPUtils.Timer定时器对象 1](#_Toc6475)

[7.2.4 NPPUtils.Log对象 1](#_Toc23681)

[7.2.5 NPPUtils.DateFormat(mask,date)格式化返回当前客户端系统时间 1](#_Toc27521)

[7.2.6 NPPUtils.GetDateTimeUTCSeconds(date)返回UTC时间戳 1](#_Toc12871)

[7.2.7 NPPUtils.DTStrToTimestamp(dateStr)标准的时间字符串转为时间戳 1](#_Toc18839)

[7.2.8 NPPUtils.Zeroize(value,length)根据长度左补零 1](#_Toc11250)

[7.2.9 NPPUtils.CheckByteLength(value, minlen, maxlen)验证字串长度 1](#_Toc2666)

[7.2.10 NPPUtils.GetStringRealLength(source) 中英文混合字符串真实长度 1](#_Toc8982)

[7.2.11 NPPUtils.Regexs 1](#_Toc11976)

[7.2.12 NPPUtils.UTF8toUnicode utf8编码字符转为unicode编码 1](#_Toc29890)

[7.2.13 NPPUtils.UnicodetoUTF8 unicode编码字符转为utf8编码 1](#_Toc26895)

[7.2.14 NPPUtils.UnicodetoGB2312 unicode编码字符转为GB2312编码 1](#_Toc29991)

[7.2.15 NPPUtils.GB2312toUnciode GB2312编码字符转为unicode编码 1](#_Toc8916)

[7.3 文件操作对象NPPILY.Folder 1](#_Toc15648)

[7.3.1 NPPILY.Folder.GetSystemRoot获取操作系统根目录 1](#_Toc30900)

[7.3.2 NPPILY.Folder.DebugSwitch(boolean bValue)切换插件调试 1](#_Toc2367)

[7.3.3 NPPILY.Folder.GetFileFolder()打开目录选择对话框 1](#_Toc9661)

[7.3.4 NPPILY.Folder.OpenFolder(string folderPath)打开文件夹 1](#_Toc24083)

[7.3.5 NPPILY.Folder.CreateDirectory(string folderPath)创建文件夹 1](#_Toc25363)

[7.3.6 NPPILY.Folder.DeleteDirectory(string folderPath)删除文件夹 1](#_Toc956)

[7.3.7 NPPILY.Folder.DeleteFile(string fileName)删除文件 1](#_Toc9655)

[7.3.8 NPPILY.Folder.FileExist(string fileName)检测文件是否存在 1](#_Toc15118)

[7.3.9 NPPILY.Folder.ReadFile(string fileName)读文件内容 1](#_Toc13219)

[7.3.10 NPPILY.Folder.WriteFile(string fileName, string content, uint contentLength, boolean bValue)写文件内容 1](#_Toc24779)

[7.3.11 NPPILY.Folder.GetFolderFiles(string folderPath, fileType, bValue, bReturnFullPath)按类型获取目录下的文件列表 1](#_Toc4749)

[7.3.12 NPPILY.Folder.GetFolders(string folderPath, bValue, bReturnFullPath)获取目录下的文件夹列表 1](#_Toc6084)

[7.3.13 NPPILY.Folder.Base64Enc(string segment)进行Base64编码 1](#_Toc9753)

[7.3.14 NPPILY.Folder.Base64Dec(string base64EnStr, boolean bUsingUTF8)进行Base64解码 1](#_Toc6363)

[7.3.15 NPPILY.Folder.DesEnc(string base64Raw, string base64Key)Des编码 1](#_Toc16449)

[7.3.16 NPPILY.Folder.DesDec(string base64EnStr, string base64KeyStr)Des解码 1](#_Toc61)

[7.3.17 NPPILY.Folder.GetRandTokenForUser(string base64EnStr, string keyStr) 1](#_Toc15919)

[7.3.18 NPPILY.Folder.SetRandTokenForUser(string rawData, string keyStr) 1](#_Toc26962)

[7.3.19 NPPILY.Folder.ReadFileEx(string fileName)读取指定文件内容，经Base64编码后返回 1](#_Toc23795)

[7.3.20 NPPILY.Folder.CopyFile(string srcFileName, string dstFileName)将文件复制到指定目录下 1](#_Toc30506)

[7.3.21 NPPILY.Folder.StartHttpDownload(string url, string saveDir, boolean autoRun)开始下载解码库插件 1](#_Toc3237)

[7.3.22 NPPILY.Folder.CloseHttpDownload()停止下载解码库插件 1](#_Toc5502)

[7.3.23 NPPILY.Folder.GetStatus()获取解码库插件下载状态 1](#_Toc32259)

[7.3.24 NPPILY.Folder.GetPluginVersion()获取解码库插件版本号 1](#_Toc24182)

[7.3.25 NPPILY.Folder.PUDetectStartup()开启设备搜索功能 1](#_Toc17798)

[7.3.26 NPPILY.Folder.PUDetectGetPUList()搜索设备列表 1](#_Toc13262)

[7.3.27 NPPILY.Folder.PUDetectCleanup()关闭设备搜索功能 1](#_Toc2170)

1. **插件**
   1. **插件包**
      1. **MediaPlugin7.exe**

网页插件。需要在第一次加载网页时，手动下载安装插件。支持并兼容市面主流浏览器（IE, Chrome, Firefox）

1. **接口方法**
   1. **NPPInterlayer.js脚本包含和依赖**

NPPInterlayer.js依赖于NPPInterface.js脚本库，必须先调用NPPInterface.js脚本库后再调用NPPInterlayer.js

* 1. **NPPInterlayer对象**

网页脚本对象不可以和NPPILY以及NPPIF重名

* 1. **NPPInterlayer自定义结构类型**
     1. **初始化NPPILY对象参数结构NPPILY.Struct.InitParamStruct**

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| debug | Bool NULL | 是否开始调试状态  true开启  false不开启（默认） |
| cb | Function NULL | 调试信息输出回调函数，回调时参数的结构参考NPPILY.Struct.DebugMessageStruct |
| extraParams | Object NULL | 其他参数，格式为  {  language(NPPILY.Enum.LanguageType) 语言风格  warmTip(object):  {  active(boolean) 是否使能未加载插件时内部温馨提示，默认false不提示 pluginFile(string) 插件绝对地址或网络地址，默认"MediaPlugin7.exe"  html(html code) 自定义网页提示代码  }  }，可以省略 |

* + 1. 调试信息结构NPPILY.Struct.DebugMessageStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| time | String NOT NULL | 事件发生时间 |
| fn | FUNCTION null | 事件发生的函数 |
| msg | String NOT NULL | 事件描述 |

* + 1. 返回值结构 NPPILY.Struct.ReturnValueStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| rv | Number | 错误码（查看附录定义） |
| response | Object NULL | 当rv为0（常量NrcapError.NRCAP\_SUCCESS）时，返回正确，有效的值 |

* + 1. 连接服务器或直连设备参数结构NPPILY.Struct.ConnParamStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| path | String NOT NULL | 连接地址，结构为ip:port（支持域名）  例如127.0.0.1:8866、www.crearo.com:8866等 |
| epId | String NOT NULL | 企业ID，例如system |
| username | String NOT NULL | 登录用户名，例如admin |
| password | String NULL | 登录密码 |
| bFixCUIAddress | String NULL | 是否透过网闸模式(是1/true，否0/false)，默认为否 |

* + 1. 连接对象信息结构NPPILY.Struct.ConnectionStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| key | String NOT NULL | Connect ID创建一个连接后，系统生成的Conenct ID |
| connParam | NPPILY.Struct.ConnParamStruct | 连接登录参数 |
| ncName | Object NOT NULL | NC插件名称 |
| nc | Object NOT NULL | NC插件实例 |
| connType | NPPILY.Enum.ConnectionType | 连接类型，判断是连接C7平台.Server还是直连设备.Device |
| status | NPPILY.Enum.ConnectionStatus | 连接登陆状态 |
| session | String NOT NULL | NC连接句柄 |
| CUID | String NOT NULL | 用户连接ＩＤ（ＣＵＩＤ），连接C7平台时有效 |
| userPriority | String NOT NULL | 用户优先级 |
| systemName | String NOT NULL | 登陆C7平台后获取来的C7平台系统名 |
| domainRoad | Array NULL | 登陆的是C7平台，才有可能有子域信息（即C7平台为级联的根平台，其他为级联的子域平台），domainRoad为空，表示没有级联的子域平台；例如domainRoad为[“A”, “C”, “B.A”, “D.B.A”]，其中的英文点号“.”表示隶属关系，如“B.A”代表B平台为A的子域平台，A又隶属于此处登录的平台 |
| resource | Array(NPPILY.Struct.PUNodeStruct) NULL | 资源信息（保留未启用） |

* + 1. 窗口信息结构NPPILY.Struct.WindowStruct

|  |  |  |  |
| --- | --- | --- | --- |
| 变量 | | 类型 | 描述 |
| container | | Object NOT NULL | 插件容器DOM对象 |
| containerId | | String NOT NULL | 插件容器ID |
| wnd | | Object NOT NULL | 窗口插件DOM对象 |
| wndName | | String NOT NULL | 窗口插件ID名称 |
| connectId | | String NOT NULL | NPPILY.Connect创建的连接ID |
| wndHandle | | String NOT NULL | 窗口句柄 |
| type | | NPPILY.Enum.WindowType NOT NULL | 窗口类型 |
| params | | Object | 相关视频参数 |
|  | puid | String NOT NULL | 设备PUID |
| idx | String NULL | 视频资源索引，默认为0 |
| streamType | NPPILY.Enum.NrcapStreamType  NOT NULL | 流类型 |
| ivStreamHandle | String NULL | 视频流句柄 |
| iaStreamHandle | String NULL | 音频流句柄 |
| alg | String NULL | 平台转码流编码算法 |
| resolution | String NULL | 平台转码流分辨率 |
| bitRate | String NULL | 平台转码流码率 |
| frameRate | String NULL | 平台转码流帧率 |
| speed | String NULL | 点播速度 |
| startTime | String NULL | 点播开始时间 |
| endTime | String NULL | 点播结束时间 |
| fileFullPath | String NULL | 点播文件全路径 |
| fileTimeLength | String NULL | 点播文件时长 |
| status | | Object | 视频窗口状态 |
|  | isddrawing | Bool NULL FALSE | 是否使用了ddraw模式 |
| isplaybyP2P | Bool NULL FALSE | 是否使用了P2P（点对点）方式观看视频（未启用） |
| recording | Bool NULL FALSE | 是否在本地录像 |
| playaudioing | Bool NULL FALSE | 是否在播放音频 |
| playvideoing | Bool NULL FALSE | 是否在播放视频 |
| playvoding | Bool NULL FALSE | 是否在点播录像 |
| isfullscreening | Bool NULL FALSE | 视频窗口是否在全屏 |
| customParams | | Object NULL | 自定义参数 |
| style | | Object | 窗口样式（未启用） |

* + 1. 窗口容器信息结构 NPPILY.Struct.WindowContainerStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| container | Object NOT NULL | 窗口插件的Dom容器对象 |
| type | NPPILY.Enum.WindowType  NOT NULL | 窗口类型 |
| active | Bool NULL | 窗口是否被激活 |
| window | NPPILY.Struct.WindowStruct NULL | 窗口对象，保存NPPILY.CreateWindow创建的窗口对象 |
| description | Object | 正在播放的视频资源描述 |

* + 1. PU资源节点结构NPPILY.Struct.PUNodeStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| puid | String NOT NULL | 设备PUID |
| resType | String NOT NULL | 资源类型，一般为  NPPILY.Enum.PuResourceType.SELF |
| resIdx | String NOT NULL | 资源索引，一般为”0” |
| name | String NOT NULL | 设备名称 |
| description | String NULL | 描述 |
| enable | String NOT NULL | 使能（”1”是”0”否） |
| online | String NOT NULL | 在线状态（”1”是”0”否） |
| immitted | String NOT NULL | 是否允许接入（”0”不允许，”1”允许） |
| modelName | String NOT NULL | 站点型号 |
| modelType | String NOT NULL | 站点类型，参考NPPILY.Enum.PUModelType |
| manufactrueID | String NOT NULL | 厂商ID |
| hardwareVersion | String NOT NULL | 硬件版本号 |
| softwareVersion | String NOT NULL | 软件版本号 |
| deviceID | String NOT NULL | 设备ID |
| latitude | String NOT NULL | 固定点纬度 |
| longitude | String NOT NULL | 固定点经度 |
| \_HANDLE | String NOT NULL | 设备资源句柄 |
| childResource | Array NULL | 子资源Array |

* + 1. PU子资源节点结构NPPILY.Struct.PUResourceNodeStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| type | String NOT NULL | 资源类型，参考NPPILY.Enum.PUResourceType |
| idx | String NOT NULL | 索引号 |
| name | String NOT NULL | 名称 |
| description | String NOT NULL | 描述 |
| enable | String NOT NULL | 使能 |
| \_HANDLE | String NOT NULL | 设备子资源句柄 |

* + 1. 事件信息结构 NPPILY.Struct.NCObjectNotifyStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| eventName | String NOT NULL | 事件名称  对应NC事件类型枚举NPPILY.Enum.NCObjectNotify |
| \_HANDLE | String NOT NULL | 事件对应的句柄 |
| status | String NULL | 事件所对应的状态，根据eventName判断有无 |
| statusDesc | String NULL | 事件所对应的状态描述，根据eventName判断有无 |
| errorCode | String NULL | 事件所对应的错误码 |
| keyData | Object NULL | 事件所对应的重要信息对象，某些状态下可能为空对象{}，请注意甄别 |

* + 1. 窗口事件信息结构NPPILY.Struct. WindowEventStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| lbtn\_click | Object NULL | 鼠标左键点击  {  status(boolean) 是否启用，默认false  callback(function) 回调函数  }，回调时callback(x,y)，(x,y)代表窗口中点击的坐标 |
| select\_rect | Object NULL | 矩形框选择  {  status(boolean) 是否启用，默认false  callback(function) 回调函数  }，注意在窗口模式为“框选”时有效，回调时callback(x1, y1, x2, y2)，其中(x1,y1)、(x2,y2)分别代表窗口中框选的左上与右下坐标 |
| ptz\_control | Object NULL | 云台控制  {  status(boolean) 是否启用，默认false  callback(function) 回调函数  }，注意在窗口模式为“拖拽”时有效 |
| fsw\_show | Object NULL | 显示全屏  {  status(boolean) 是否启用，默认true（建议不进行设置，因为窗口插件双击总会全屏或退出全屏，设置为false后全屏不会显示画面）  callback(function) 回调函数  }，注意在窗口模式为“拖拽”时有效，回调时callback(newWndHandle)，newWndHandle为全屏窗口句柄 |
| fsw\_hide | Object NULL | 关闭全屏  {  status(boolean) 是否启用，默认true（建议不进行设置，因为窗口插件双击总会全屏或退出全屏）  callback(function) 回调函数  }，注意在窗口模式为“拖拽”时有效，回调时callback(oldWndHandle)，oldWndHandle为原来窗口句柄 |
| menu\_command | Object NULL | 自定义右键菜单项  {  status(boolean) 是否启用，默认false，必须设置成true右键菜单才有效  menu(Array) 菜单项  callback(function) 回调函数  }，注意在窗口模式为“拖拽”时有效，  menu属性取值结构为:new Array（{key: “?”, text: “?”},...]）  如：[{key: “-”, text: “split\_1”}, {key: “localSnapshot”, text: “本地抓拍”}, {key: “-”, text: “split\_2”}]  其中key为某个菜单项的标识，不可以重复（如果key为“-”、“separator”、“split”之一，表示分割条，可以有若干个，text取值可以有所区别，为后续菜单项操作有用），回调时callback(key);menu节点可以为空的数组，那么后续添加右键菜单项需要使用追加菜单项接口，参考3.7.17 |

* + 1. 中心存储文件信息结构NPPILY.Struct.SCIVDateFileStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| fileName | String NOT NULL | 文件名称 |
| filePath | String NOT NULL | 文件平台保存根目录 |
| fileSize | String NOT NULL | 文件大小 |
| beginTime | UTC timestamp NOT NULL | 文件开始时间 |
| endTime | UTC timestamp NOT NULL | 文件结束时间 |
| reason | String NOT NULL | 文件原因（最终以英文竖线|连接多个原因字符串） |
| csuPuid | String NOT NULL | 中心存储器PUID |
| csuIndex | Uint NOT NULL | 中心存储单元资源索引，缺省为0 |
| puid | String NOT NULL | 所属设备PUID |
| type | String NOT NULL | 所属资源类型，一般固定为NPPILY.Enum.PuResourceType.VideoIn |
| idx | Uint NOT NULL | 所属资源索引，缺省为0 |

* + 1. 前端CEFS录像文件信息结构NPPILY.Struct.CEFSRecordStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| needNextQuery | Boolean NOT NULL | 是否需要下次查询 |
| nextQueryBeginTime | UTC timestamp NULL | 下次查询时间（needNextQuery=true时有此节点） |
| beginTime | UTC timestamp NOT NULL | 文件开始时间 |
| endTime | UTC timestamp NOT NULL | 文件结束时间 |
| bitRate | Uint NOT NULL | 码率（单位kbps），可以根据此进行估计文件大小  bitRate\*(endTime-beginTime) |
| reason | NPPILY.Enum.CEFSRecordReason NOT NULL | 文件原因  原因描述，可以根据NPPILY.Enum.CEFSRecordReasonDesc[reason][NPPILY.language]得到 |
| puid | String NOT NULL | 所属设备PUID |
| resType | String NOT NULL | 所属资源类型，一般固定为NPPILY.Enum.PuResourceType.VideoIn |
| resIdx | Uint NOT NULL | 所属资源索引，缺省为0 |

* + 1. 前端CEFS图片文件信息结构NPPILY.Struct.CEFSPictureStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| needNextQuery | Boolean NOT NULL | 是否需要下次查询 |
| nextQueryBeginTime | UTC timestamp NULL | 下次查询时间（needNextQuery=true时有此节点） |
| time | UTC timestamp NOT NULL | 抓拍时间 |
| noInSecond | Uint NOT NULL | 秒内编号，一秒内可能有多个图片，从0开始 |
| reason | NPPILY.Enum.CEFSPictureReason NOT NULL | 文件原因  原因描述，可以根据NPPILY.Enum.CEFSPictureReason[reason][NPPILY.language]得到 |
| puid | String NOT NULL | 所属设备PUID |
| resType | String NOT NULL | 所属资源类型，一般固定为NPPILY.Enum.PuResourceType.VideoIn |
| resIdx | Uint NOT NULL | 所属资源索引，缺省为0 |

1. * 1. 前端CEFS用户日志信息结构NPPILY.Struct.CEFSUserLogStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| needNextQuery | Boolean NOT NULL | 是否需要下次查询 |
| nextQueryBeginTime | UTC timestamp NULL | 下次查询时间（needNextQuery=true时有此节点） |
| time | UTC timestamp NOT NULL | 产生时间 |
| type | NPPILY.Enum.CEFSUserLogType NOT NULL | 用户日志类型  类型描述，可以根据NPPILY.Enum.CEFSUserLogType[type][NPPILY.language]得到 |
| data | String NOT NULL | 用户日志数据（Base64编码的） |
| puid | String NOT NULL | 所属设备PUID |

* + 1. 前端CEFS报警事件信息结构NPPILY.Struct.CEFSAlarmEventStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| needNextQuery | Boolean NOT NULL | 是否需要下次查询 |
| nextQueryBeginTime | UTC timestamp NULL | 下次查询时间（needNextQuery=true时有此节点） |
| time | UTC timestamp NOT NULL | 产生时间 |
| type | NPPILY.Enum.CEFSAlarmEventType NOT NULL | 报警事件类型  类型描述，可以根据NPPILY.Enum.CEFSAlarmEventType[type][NPPILY.language]得到 |
| data | String NOT NULL | 报警事件数据（Base64编码的） |
| puid | String NOT NULL | 所属设备PUID |
| resType | String NOT NULL | 所属资源类型，一般固定为NPPILY.Enum.PuResourceType.AlertIn |
| resIdx | Uint NOT NULL | 所属资源索引，缺省为0 |

* + 1. 前端GPS数据信息结构NPPILY.Struct.CEFSGPSDataStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| needNextQuery | Boolean NOT NULL | 是否需要下次查询 |
| nextQueryBeginTime | UTC timestamp NULL | 下次查询时间（needNextQuery=true时有此节点） |
| time | UTC timestamp NOT NULL | 产生时间 |
| latitude | String NOT NULL | 纬度,浮点型字符串,取值范围[-90,90],北纬为正,南纬为负 |
| longitute | String NOT NULL | 经度,浮点型字符串,取值范围[-180,180),东经为正,西经为负 |
| bearing | String NOT NULL | 方向,浮点型字符串,取值范围[0,360),正北为0,正东为90,依次类推. |
| speed | Uint NOT NULL | 速度,浮点型字符串,单位km/h |
| altitude | Uint NOT NULL | 海拔,浮点型字符串,单位m,小于等于-50000表示无效. |
| offlineFlag | Uint NOT NULL | 离线标识,1表示下线是存的数据,0表示实时数据. |
| state | Uint NOT NULL | GPS状态,0表示正常,1表示无信号,2表示无模块. |
| maxSpeed | Uint NOT NULL | 最高限速,整型字符串,单位km/h |
| minSpeed | Uint NOT NULL | 最低限速,整型字符串,单位km/h |
| puid | String NOT NULL | 所属设备PUID |

* + 1. 逻辑分组信息结构NPPILY.Struct.LogicGroupStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| index | Unit NOT NULL | 逻辑分组索引 |
| name | String NOT NULL | 逻辑分组名称 |
| lastRefreshTime | string | 最新刷新时间 |
| refreshInterval | string | 刷新时间间隔 |
| childResource | Array<NPPILY.Struct.LogicGroupNodeStruct> NULL | 子节点或资源  （保留属性） |

* + 1. 逻辑分组节点信息结构NPPILY.Struct.LogicGroupNodeStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| index | Unit NOT NULL | 逻辑分组节点索引 |
| name | String NOT NULL | 逻辑分组节点名称 |
| parentNode\_Index | Unit NOT NULL | 逻辑分组节点上级索引，根节点上级为0 |
| childResource | Array<NPPILY.Struct.LogicGroupNodeStruct|NPPILY.Struct.LogicGroupResourceStruct> NULL | 子节点或资源  （保留属性） |

* + 1. 逻辑分组节点下资源信息结构NPPILY.Struct.LogicGroupResourceStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| puid | String NOT NULL | 设备PUID |
| type | String NOT NULL | （视频）资源类型，一般为NPPILY.Enum.PuResourceType.VideoIn |
| idx | Unit NOT NULL | （视频）资源索引 |
| name | String NOT NULL | （视频）资源名称 |
| description | String NOT NULL | （视频）资源描述 |
| enable | Unit NOT NULL | （视频）资源是否使能 |
| parentNode\_Index | Unit NOT NULL | 逻辑分组节点资源上级节点索引 |

* + 1. 资源描述信息结构NPPILY.Struct.CommonResDescriptionStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| puid | String NOT NULL | 设备PUID |
| resType | String NOT NULL | 资源类型 |
| resIdx | Unit NOT NULL | 资源索引 |
| name | String NOT NULL | 资源名称 |
| description | String NOT NULL | 资源描述 |
| enable | Unit NOT NULL | 资源是否使能 |

* + 1. 视频窗口叠加文字信息结构NPPILY.Struct.WindowTextAddStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| wndBlockNo | Number NOT NULL | 字符叠加的序号，从0开始，0,1,2，...15（默认为0） |
| left | Number NOT NULL | 叠加字幕的横向坐标,取值范围不限 |
| top | Number NOT NULL | 叠加字幕的纵向坐标,取值范围不限 |
| width | Number NOT NULL | 每个字符的宽度,没有范围,视情况而定 |
| height | Number NOT NULL | 每个字符的高度,没有范围,视情况而定 |
| color | Number NOT NULL | RGB(r,g,b)计算出来的值r(red), g(green), b(blue) 取值范围 0~255  公式为(65536 \* b + 256 \* g + r) |
| enableTilt | Number NOT NULL | 0表示不倾斜,1表示倾斜,默认不倾斜 |
| enableUnderline | Number NOT NULL | 0表示不带下划线,1表示带下划线,默认不带下划线 |
| fontFamily | String NOT NULL | 字体的字样,宋体,华文行楷,隶书等等,此参数的最大长度为64 |
| content | String NOT NULL | 叠加的字符,最大长度128 |

* + 1. GPS数据存储资源结构NPPILY.Struct.GPSDataStorageResourceStruct

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| puid | String NOT NULL | 所属设备PUID |
| resType | NPPILY.Enum.PuResourceType NOT NULL | 资源类型，一般为NPPILY.Enum.PuResourceType.GPS |
| resIdx | Number NOT NULL | 资源索引 |
| optID | String NOT NULL | 操作ID |
| value | NPPILY.Enum.NrcapStreamType NOT NULL | 流类型 |

* + 1. 中心存储GPS数据文件信息结构NPPILY.Struct.CSUGPSDataFileStruct

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量 | 类型 | | 描述 | |
| time | | UTC timestamp NOT NULL | 产生时间 |
| latitude | | String NOT NULL | 纬度,浮点型字符串,取值范围[-90,90],北纬为正,南纬为负 |
| longitute | | String NOT NULL | 经度,浮点型字符串,取值范围[-180,180),东经为正,西经为负 |
| bearing | | String NOT NULL | 方向,浮点型字符串,取值范围[0,360),正北为0,正东为90,依次类推. |
| speed | | Uint NOT NULL | 速度,浮点型字符串,单位km/h |
| altitude | | Uint NOT NULL | 海拔,浮点型字符串,单位m,小于等于-50000表示无效. |
| state | | Uint NOT NULL | GPS状态,0表示正常,1表示无信号,2表示无模块. |
| maxSpeed | | Uint NOT NULL | 最高限速,整型字符串,单位km/h |
| minSpeed | | Uint NOT NULL | 最低限速,整型字符串,单位km/h |
| timestemp | | Uint NOT NULL |  |
| puid | String NOT NULL | | 所属设备PUID | |
| resType | NPPILY.Enum.PuResourceType NOT NULL | | 资源类型，一般为NPPILY.Enum.PuResourceType.GPS | |
| resIdx | Number NOT NULL | | 资源索引 | |
| value | NPPILY.Enum.NrcapStreamType NOT NULL | | 流类型 | |

* + 1. 网络文件下载状态结构NPPILY.Struct.HttpDownloadStatusStruct

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 变量 | 类型 | | 描述 | |
| flag | | Number NOT NULL | 标记  -1 下载失败  0 下载完成  1 正在下载 |
| description | | String NOT NULL | 状态描述 |
| speed | | String NOT NULL | 下载速率，单位bps |
| totalLength | | String NOT NULL | 文件总长度，单位字节 |
| currentLength | | Uint NOT NULL | 文件已下载长度，单位字节 |

* 1. NPPILY自定义枚举类型
     1. NPPILY视频流类型NPPILY.Enum.NrcapStreamType

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| REALTIME | REALTIME | 实时流 |
| STORAGE | STORAGE | 存储流 |
| PICTURE | PICTURE | 图片流 |
| MOBILE2G | MOBILE2G | 2G手机观看码流 |
| MOBILE3G | MOBILE3G | 3G手机观看码流 |
| TRANSCODE | TRANSCODE | 平台转码流 |
| HD | HD | 高清流 |

* + 1. NPPILY视频窗口类型NPPILY.Enum.WindowType

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| VIDEO | VIDEO | 实时视频窗口 |
| VOD | VOD | 录像回放窗口 |

* + 1. NPPILY云台控制命令NPPILY.Enum.PTZDirection

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| turnup | PTZ\_StartTurnUp | 向上转 |
| turndown | PTZ\_StartTurnDown | 向下转 |
| turnleft | PTZ\_StartTurnLeft | 向左转（动） |
| turnright | PTZ\_StartTurnRight | 向右转 |
| stopturn | PTZ\_StopTurn | 停止转动 |
| aperturea | PTZ\_AugmentAperture | 增大光圈 |
| aperturem | PTZ\_MinishAperture | 减小光圈 |
| stopaperture | PTZ\_StopApertureZoom | 停止光圈缩放 |
| focusfar | PTZ\_MakeFocusFar | 推远焦点 |
| focusnear | PTZ\_MakeFocusNear | 拉近焦点 |
| stopfocus | PTZ\_StopFocusMove | 停止焦点调整 |
| zoomin | PTZ\_ZoomInPicture | 放大图像 |
| zoomout | PTZ\_ZoomOutPicture | 缩小图像 |
| stopzoom | PTZ\_StopPictureZoom | 停止图像缩放 |
| startsecondarydev | PTZ\_StartSecondaryDev | 开启辅助设备 |
| stopsecondarydev | PTZ\_StopSecondaryDev | 关闭辅助设备 |
| movetopresetpos | PTZ\_MoveToPresetPos | 前往预置位 |
| setpresetpos | PTZ\_SetPresetPos | 设置预置位 |

* + 1. NPPILY浏览器类型 NPPILY.Enum.BrowserType

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| IE | IE | IE浏览器 |
| Chrome | Chrome | 谷歌浏览器 |
| FireFox | FireFox | 火狐浏览器 |
| Other | Other | 其他浏览器 |

* + 1. NPPILY支持的语言类型 NPPILY.Enum.LanguageType

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| zh\_CN | zh\_CN | 中文 |
| en | en | 英文 |

* + 1. NPPILY建立的连接类型 NPPILY.Enum.ConnectionType

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| Server | Server | 连接C7平台服务器 |
| Device | Device | 直连设备 |

* + 1. NPPILY建立的连接状态NPPILY.Enum.ConnectionStatus

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| Idle | Idle | 初始状态 |
| Connecting | Connecting | 正在建立连接 |
| Connected | Connected | 连接已建立 |
| Failed | Failed | 建立连接失败 |

* + 1. NPPILY获取或构建资源级别 NPPILY.Enum.ForkResourceLevel

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| nppForkPUInfo | nppForkPUInfo | 获取PU节点信息 |
| nppForkPUResourceInfo | nppForkPUResourceInfo | 获取PU节点资源信息 |
| nppForkOnePUInfo | nppForkOnePUInfo | 获取一个PU节点信息 |

* + 1. NPPILY获取的设备类型NPPILY.Enum.PuModelType

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| ENC | ENC | 有线编码器 |
| WENC | WENC | 无线编码器 |
| DEC | DEC | 有线解码器 |
| WDEC | WDEC | 无线解码器 |
| CSU | CSU | 中心存储单元 |
| ESU | ESU | 企业自建存储单元 |

* + 1. NPPILY获取的资源类型NPPILY.Enum.PuResourceType

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| GPS | GPS | GPS模块 |
| WIFI | WIFI | WIFI模块 |
| AudioIn | IA | 输入音频 |
| AlertIn | IDL | 输入报警 |
| AudioOut | OA | 输出音频 |
| AlertOut | ODL | 输出报警 |
| VideoIn | IV | 输入视频 |
| VideoOut | OV | 输出视频 |
| PTZ | PTZ | 云台 |
| SELF | SELF | 站点 |
| Storager | SG | 存储器(前端存储) |
| SC | SC | 存储器(中心存储) |
| SerialPort | SP | 设备串口 |
| Wireless | WM | 无线模块 |

* + 1. NPPILY NC事件对象NPPILY.Enum.NCObjectNotify

|  |  |  |
| --- | --- | --- |
| 定义 | 值 | 描述 |
| event\_notify | event\_notify | 接收平台（设备）事件通知  注：事件信息结构 NPPILY.Struct.NCObjectNotifyStruct中（下同）\_HANDLE句柄对应连接session，可以根据connectId（连接ID）到NPPILY.Connections中去匹配；如果页面中只存在一个连接信息，那么这些事件肯定都属于此连接周期。（当通知对象的errorCode非0时，视为连接已断开） |
| stream\_status\_notify | stream\_status\_notify | 接收流状态通知  \_HANDLE句柄包含以下几种：  1、视频流（实时视频与点播回放）的句柄，视频播放成功后即为窗口信息结构NPPILY.Struct.WindowStruct中params.ivStreamHandle   1. 开启喊话与双向对讲返回的句柄 2. 开启存储文件下载返回的句柄，那么下载状态等信息可以根据此回调事件接收到 3. 会议室流句柄的通知 |
| gps\_data\_notify | gps\_data\_notify | 接收GPS信息通知  注：\_HANDLE句柄对应开启接收设备的GPS流时response节点值 |
| copy\_file\_notify | copy\_file\_notify | 外接磁盘存储文件下载信息通知  注：\_HANDLE、errorCode无意义，keyData结构为{  localFilePath(string) 本地保存全路径（根据这个区别不同的下载信息）  currentBytes(uint)已经下载的字节数  currentTime(uint) 下载已经用去的时间秒数  status(uint) 下载状态量（-1：下载异常出错，0：正在下载，1：下载完成）  statusDesc(string) 下载状态描述  } |

* + 1. NPPILY 存储文件类型NPPILY.Enum.StorageFileType

|  |  |  |
| --- | --- | --- |
| 定义（常量） | 值 | 描述 |
| Platform | Platform | 中心存储 |
| CEFS | CEFS | 前端CEFS存储 |
| FAT32 | FAT32 | 前端FAT32存储 |

* + 1. NPPILY CEFS查询命令NPPILY.Enum.CEFSRequestID

|  |  |  |
| --- | --- | --- |
| 定义（常量） | 值 | 描述 |
| QueryRecordDate | QueryRecordDate | 查询录像日期命令 |
| QueryRecord | QueryRecord | 查询录像命令 |
| QueryLinkActionRecord | QueryLinkActionRecord | 查询联动录像命令 |
| QueryPictureDate | QueryPictureDate | 查询图片日期命令 |
| QueryLinkActionPicture | QueryLinkActionPicture | 查询联动抓图命令 |
| QueryPictureHalfHour | QueryPictureHalfHour | 查询半小时图片命令 |
| QueryUserLog | QueryUserLog | 查询用户日志命令 |
| QueryUserLogDate | QueryUserLogDate | 查询用户日志日期命令 |
| QueryUserLogHalfHour | QueryUserLogHalfHour | 查询半小时用户日志命令 |
| QueryAlarmEvent | QueryAlarmEvent | 查询报警事件命令 |
| QueryAlarmEventDate | QueryAlarmEventDate | 查询报警事件日期命令 |
| QueryAlarmEventHalfHour | QueryAlarmEventHalfHour | 查询半小时报警事件命令 |
| QueryGPSData | QueryGPSData | GPS数据查询命令 |
| QueryGPSDataDate | QueryGPSDataDate | 查询GPS数据日期命令 |

* + 1. NPPILY CEFS录像原因类型NPPILY.Enum.CEFSRecordReason

|  |  |  |
| --- | --- | --- |
| 定义（常量） | 值 | 描述 |
| Schedule | Schedule | 前端计划录像 |
| Manual | Manual | 手动录像 |
| AlertIn | LinkAction(AlertIn) | 联动录像（发生报警） |
| MotionDetected | LinkAction(MotionDetected) | 联动录像（侦测到移动） |
| SignalLost | LinkAction(SignalLost) | 联动录像（视频信号丢失） |

* + 1. NPPILY CEFS抓拍原因类型NPPILY.Enum.CEFSPictureReason

|  |  |  |
| --- | --- | --- |
| 定义（常量） | 值 | 描述 |
| Schedule | Schedule | 前端抓拍录像 |
| Manual | Manual | 手动抓拍 |
| AlertIn | LinkAction(AlertIn) | 联动抓拍（发生报警） |
| MotionDetected | LinkAction(MotionDetected) | 联动抓拍（侦测到移动） |
| SignalLost | LinkAction(SignalLost) | 联动抓拍（视频信号丢失） |

* + 1. NPPILY CEFS用户日志类型NPPILY.Enum.CEFSUserLogType

|  |  |  |
| --- | --- | --- |
| 定义（常量） | 值 | 描述 |
| UserLogin | UserLogin | 用户登录 |
| UserLogout | UserLogout | 用户退出 |
| Booted | Booted | 系统启动 |
| Reboot | Reboot | 系统重启 |
| AdjustTime | AdjustTime | 修改系统时间 |
| Formatted | Formatted | 完成格式化 |
| OpenDoor | OpenDoor | 打开挡板 |
| CloseDoor | CloseDoor | 关闭挡板 |
| PowerOff | PowerOff | 系统关机 |

* + 1. NPPILY CEFS报警事件类型NPPILY.Enum.CEFSAlarmEventType

|  |  |  |
| --- | --- | --- |
| 定义（常量） | 值 | 描述 |
| AlertIn | AlertIn | 报警输入发生报警 |
| MotionDetected | MotionDetected | 视频侦测到移动 |
| SignalLost | SignalLost | 视频信号丢失 |

* 1. NPPILY全局变量

|  |  |  |
| --- | --- | --- |
| 定义 | 类型 | 描述 |
| version | String | NPPILY版本号 |
| debug | Bool | 是否在调试状态下 |
| language | NPPILY.Enum.LanguageType | 语言版本 |
| Connections | NPPUtils.Hash（详见附录中说明） | 保存与所有服务器的连接及获取到的所有信息，详细说明见3.3.5服务器连接对象NPPILY.Struct.ConnectionStruct说明 |
| WindowContainers | NPPUtils.Hash | 保存所有视频窗口容器对象NPPILY.Struct.WindowContainerStruct |

* 1. **NPPILY对象类型**
  2. NPPILY方法（公开）
     1. 初始化NPPILY.Init(NPPILY.Struct.InitParamStruct initParam)

参数：NPPILY.Struct.InitParamStruct initParam [NULL]

调用：NPPILY.Init(initParam)

返回值：Object(NPPILY.Struct.ReturnValueStruct)

【注意我们协定NPPILY的功能接口最终返回值结构均为“返回值结构NPPILY.Struct.ReturnValueStruct”的实例对象】

说明：初始化NPPILY对象，设置调试状态。并检查插件是否安装成功。如果没有安装成功，进行页面提示。此方法必须在所有NPPILY的方法调用之前调用。

* + 1. 卸载 NPPILY.UnLoad()

参数：无

调用：NPPILY.UnLoad()

返回值：NPPILY.Struct.ReturnValueStruct

说明：卸载NPPILY对象，和NPPILY.Init方法相对应，如果初始化之后，关闭网页不调用卸载方法，会导致页面出错。Init和UnLoad必须匹配调用。

* + 1. 创建连接 NPPILY.Connect(NPPILY.Struct.ConnParamStruct connParams,Boolean allowSameDomain)

参数：NPPILY.Struct.ConnParamStruct connParams [NOT NULL]

Boolean allowSameDomain[NULL] 是否允许同平台不同用户多个连接，默认否（同平台最多建立一个连接）

调用：NPPILY.Connect(connParams, allowSameDomain)

返回值：NPPILY.Struct.ReturnValueStruct

说明：创建连接,并记录在NPPILY.Connections中。当返回值object.rv等于0时，object.response为connectId（连接ID）。否则object.rv为相应的错误码，连接失败时，会断开连接并从NPPILY.Connections中清除连接信息

* + 1. 断开连接 NPPILY.DisConnection(String connectId)

参数：String connectId [NOT NULL] 连接ID

调用：NPPILY.DisConnection(connectId)

返回值：NPPILY.Struct.ReturnValueStruct

说明：断开一个连接，与建立连接NPPILY.Connect是互斥的；

* + 1. 获取或构建设备（子）资源 NPPILY. ForkResource(String connectId, NPPILY.Enum.ForkResourceLevel forkLevel, uint offset, uint count, String domainRoad, object customParams)

参数：

String connectId [NOT NULL] 连接ID

NPPILY.Enum.ForkResourceLevel forkLevel [NOT NULL] 构建类型

uint offset [NULL] 分页查询开始偏移值

uint count [NULL] 分页查询每次返回的最大条数

String domainRoad [NULL] 根域（缺省为空）或子域名称（级联平台的名称）

object customParams [NULL]

调用：NPPILY.ForkResource(connectId, forkLevel, offset, count, domainRoad, customParams)

返回值：NPPILY.Struct.ReturnValueStruct

说明：获取设备资源或者构建子资源信息。若此次连接类型为平台连接，那么获取资源时domainRoad传入空即返回的是根域平台（此次建立连接的平台）的资源，如果domainRoad为此根域下级联平台的名称，那么就返回相应子域的资源信息；在forkLevel 等于NPPILY.Enum.ForkResourceLevel.nppForkOnePUInfo（及检测并获取特定PUID对应的资源信息）时，domainRoad可以省略；offset和count参数在forkLevel 等于NPPILY.Enum.ForkResourceLevel.nppForkPUInfo时有效。获取直连设备时，可以只传connectId即可。获取平台设备时，connectId和forkLevel参数必须传入，可分页查询资源信息。获取子资源信息时，customParams结构如下

{

PUID: <puid(string) 设备PUID>

}

返回值object.rv等于0时，即代表获取成功。object.response等于resource。Resource结构类型说明见3.3.8（NPPILY.Struct.PuNodeStruct）3.3.9（NPPILY.Struct.PuResourceNodeStruct）。否则，object.rv等于相应的错误码。

* + 1. 创建视频窗口对象 NPPILY. CreateWindow(String connectId, DOMObject/String containerOrId, NPPILY.Enum.WindowType windowType, NPPILY.Struct.WindowEventStruct windowEvent)

参数：

String connectId [NOT NULL] 连接ID

DOMObject/String containerOrId [NOT NULL] 窗口插件容器或ID

NPPILY.Enum.WindowType windowType [NULL] 窗口类型，默认VIDEO NPPILY.Struct.WindowEventStruct windowEvent [NULL] 绑定窗事件，可以省略

调用：NPPILY.CreateWindow(connectId, containerOrId, windowType, windowEvent)

返回值：NPPILY.Struct.ReturnValueStruct

说明：当返回值rv节点为常量NrcapError.NRCAP\_SUCCESS，response为创建的视频窗口对象（结构为NPPILY.Struct.WindowStruct），上层调用后此窗口对象原则上需要记录到全局窗口容器对象NPPILY.WindowContainers中。

* + 1. 改变视频窗口插件宽高 NPPILY. ResizeWindowDimension(NPPILY.Struct.WindowStruct objWnd, variant width, variant height)

参数：

NPPILY.Struct.WindowStruct objWnd [NOT NULL] 窗口信息结构对象

uint|string|percent width [NULL] 宽度，数值或字符串或百分比如0 | 50px | 50em | 100%，缺省为0

uint|string|percent height [NULL] 高度，数值或字符串或百分比如0 | 50px | 50em | 100%，缺省为0

调用：NPPILY.ResizeWindowDimension(objWnd, width, height)

说明：视频窗口创建成功后，可调用此接口进行改变视频窗口插件大小

* + 1. 播放视频 NPPILY.PlayVideo(String connectId, NPPILY.Struct.WindowStruct window, String puid, Uint idx, NPPILY.Enum.NrcapStreamType streamType, object customParams)

参数：

String connectId [NOT NULL]

NPPILY.Struct.WindowStruct window [NOT NULL] 视频窗口对象

String puid [NOT NULL]

Uint idx [NOT NULL]

NPPILY.Enum.NrcapStreamType streamType [NULL]

Object customParams [NULL]

调用：NPPILY.PlayVideo(connectId, window, puid, idx, streamType)

返回值：NPPILY.Struct.ReturnValueStruct

说明：播放一路视频。返回值object.rv等于0时，播放成功。否则object.rv返回相应的错误码

【注意】获取视频的状态，请注册流状态回调（回调函数也可以是匿名的，如下）NPPILY.NCNotifyManager.Add(NPPILY.Enum.NCObjectNotify.stream\_status\_notify, function (notify) {

// notify结构符合NPPILY.Struct.NCObjectNotifyStruct如果此条通知对应的是视频状态，那么具体数据格式形如{"eventName":"stream\_status\_notify","\_HANDLE":"1584F508","status":1,"statusDesc":"正在播放","errorCode":"0","keyData":{"bit\_rate":"21784","frame\_rate":"2"}}其中notify.\_HANDLE句柄（这里对应视频流的句柄，window.params.ivStreamHandle，通过这个来区别视频），notify.status为状态码，notify.keyData为关键数据（有时可能为空对象{}），notify.keyData.bit\_rate码率（单位bps），notify.keyData.frame\_rate帧率

});

* + 1. 播放转码流视频 NPPILY.PlayTranscodeVideo(String connectId, NPPILY.Struct.WindowStruct window, String puid, Uint idx, uint alg,string resolution, uint bitRate, uint frameRate)

参数：String connectId [NOT NULL]

NPPILY.Struct.WindowStruct window [NOT NULL] 视频窗口对象

String puid [NOT NULL]

Uint idx [NOT NULL] 视频资源索引

Uint alg [NOT NULL] 视频算法类型，如H264、MPEG4等

String resolution [NOT NULL] 分辨率，如CIF、D1等，根据支持图像分辨率决定

Uint bitRate [NOT NULL] 码率，如1->1000

Uint frameRate [NOT NULL] 帧率，如1->30

调用：NPPILY.PlayTranscodeVideo(connectId, objWnd, puid, idx, alg, resolution, bitRate, frameRate);

* + 1. 停止视频 NPPILY.StopVideo(NPPILY.Struct.WindowStruct window)

参数：NPPILY.Struct.WindowStruct window [NOT NULL]

调用：NPPILY.StopVideo(window)

返回值：NPPILY.Struct.ReturnValueStruct

说明：停止一路视频。返回值object.rv等于0时，播放成功。否则object.rv返回相应的错误码

* + 1. 播放/停止音频 NPPILY.PlayAudio(NPPILY.Struct.WindowStruct window)

参数：NPPILY.Struct.WindowStruct window [NOT NULL] 视频窗口对象

调用：NPPILY. PlayAudio (window)

返回值：NPPILY.Struct.ReturnValueStruct

说明：开启或者关闭一路音频。返回值object.rv等于0时，播放成功。否则object.rv返回相应的错误码

* + 1. 绑定事件 NPPILY. NCNotifyManager.Add(NPPILY.Enum.NCObjectNotify eventName, Function callback)

参数：NPPILY.Enum.NCObjectNotify eventName

Function callback(NPPILY.Struct.NCObjectNotifyStruct object)

调用：NPPILY. NCNotifyManager.Add(eventName, callback)

返回值：NPPILY.Struct.ReturnValueStruct

说明：注册是一个事件，通过回调函数来通知你。回调参数见3.3.10

* + 1. 移除绑定事件NPPILY. NCNotifyManager. Remove (NPPILY.Enum.NCObjectNotify eventName)

参数：NPPILY.Enum.NCObjectNotify eventName

调用：NPPILY. NCNotifyManager.Remove(eventName)

返回值：NPPILY.Struct.ReturnValueStruct

说明：移除一个事件。

* + 1. 更新绑定事件NPPILY. NCNotifyManager. Update (NPPILY.Enum.NCObjectNotify eventName Function callback)

参数：NPPILY.Enum.NCObjectNotify eventName

Function callback()

调用：NPPILY. NCNotifyManager. Update(eventName,callback)

返回值：NPPILY.Struct.ReturnValueStruct

说明：更新一个事件

* + 1. 获取平台名称 NPPILY.GetSystemName(String connectId)

参数：String connectId

调用：NPPILY.GetSystemName(counnectId)

返回值：NPPILY.Struct.ReturnValueStruct

说明：获取平台名称

* + 1. 获取或设置是否使能窗口字符叠加 NPPILY.EnableAddText(NPPILY.Struct.WindowStruct objWnd, variant bValue)

参数：NPPILY.Struct.WindowStruct objWnd [NOT NULL] 窗口对象

Boolean/uint bValue [NULL] 是（true|1）否（false|0）使能，省略或bValue=null视为获取使能状态

调用：NPPILY.EnableAddText(objWnd, bValue)

返回值：NPPILY.Struct.ReturnValueStruct

说明：在叠加之前请调用此接口使能，否则叠加的内容不予展示

* + 1. 向视频窗口叠加字符 NPPILY.SetTextAdd(NPPILY.Struct.WindowStruct objWnd, <Array|Object>NPPILY.Struct.WindowTextAddStruct textSets)

参数：NPPILY.Struct.WindowStruct objWnd [NOT NULL] 窗口对象

<Array|Object>NPPILY.Struct.WindowTextAddStruct textSets[NOT NULL] 叠加字符信息参数

调用：NPPILY.SetTextAdd(objWnd, textSets)

返回值：NPPILY.Struct.ReturnValueStruct

当只一个设置节点时，不成功返回非0，多于一个节点时，任一 个成功设置就返回0，全部不成功返回非0

说明： 向视频窗口中叠加一些描述字符串。NPPILY.Struct.WindowTextAddStruct中wndBlockNo表示叠加字符串的编号，textSets对象中如果有重复的wndBlockNo将做覆盖处理。特别注意，需先调用NPPILY.EnableAddText(objWnd, true)接口使叠加文字生效。

* + 1. 删除叠加的文字信息 NPPILY.DeleteTextAdd(NPPILY.Struct.WindowStruct objWnd[, uint wndBlockNo])

参数：NPPILY.Struct.WindowStruct objWnd [NOT NULL] 窗口对象

uint wndBlockNo[NULL] 叠加的编号(参考NPPILY.Struct.WindowTextAddStruct中定义)，省略时删除所有的信息

调用：NPPILY.DeleteTextAdd(objWnd, wndBlockNo)

返回值：NPPILY.Struct.ReturnValueStruct

* + 1. 开启/关闭本地录像NPPILY.LocalRecord(NPPILY.Struct.WindowStruct window, String savePath, String FileName)

参数：NPPILY.Struct.WindowStruct window [NOT NULL] 窗口对象

String savePath [NULL] 存储路径目录，缺省为C:/

String fileName [NULL] 本地存储文件名，缺省格式为yyyyMMdd\_HHmmss.avi

调用：NPPILY.LocalRecord(window, savePath, fileName)

返回值：NPPILY.Struct.ReturnValueStruct

说明：开启或者关闭本地录像。

* + 1. 本地抓图 NPPILY.Snapshot(NPPILY.Struct.WindowStruct window, String savePath, String FileName)

参数：NPPILY.Struct.WindowStruct window [NOT NULL] 窗口对象

String savePath [NULL] 存储路径目录，缺省为C:/

String fileName [NULL] 本地存储文件名，可自定义，但必须是bmp类型的，缺省格式为yyyyMMdd\_HHmmss\_l.bmp（如20140624\_153010\_560.bmp）

调用：NPPILY.Snapshot(window, savePath, fileName)

返回值：NPPILY.Struct.ReturnValueStruct

说明：无视频画面时，调用接口成功，但需要等到有视频画面时插件才能真实执行抓拍操作；本地可能不会及时创建相应抓图，故需要稍候检测本地图片是否存在！

* + 1. 开启窗口全屏 NPPILY. FullScreen(NPPILY.Struct.WindowStruct window)

参数：NPPILY.Struct.WindowStruct window [NOT NULL]

调用：NPPILY.FullScreen(window)

返回值：NPPILY.Struct.ReturnValueStruct

说明：开启窗口全屏

* + 1. 退出窗口全屏 NPPILY. ExitFullScreen(NPPILY.Struct.WindowStruct window)

参数：NPPILY.Struct.WindowStruct window [NOT NULL]

调用：NPPILY.ExitFullScreen(window)

返回值：NPPILY.Struct.ReturnValueStruct

* + 1. 添加窗口右键菜单项 NPPILY.WindowAttachEvent.AppendMenuCommand(NPPILY.Struct.WindowStruct window，Array customMenus)

参数：NPPILY.Struct.WindowStruct window [NOT NULL] 视频窗口对象

Array customMenus [NULL]

调用：NPPILY.WindowAttachEvent.AppendMenuCommand(window, customMenus)

返回值：NPPILY.Struct.ReturnValueStruct

说明：添加窗口自定义菜单项。customMenus参数结构为:new Array（{key: “?”, text: “?”},...]），如：[{key: “-”, text: “split\_1”}, {key: “localSnapshot”, text: “本地抓拍”}, {key: “-”, text: “split\_2”}]其中key为某个菜单项的标识，不可以重复（如果key为“-”、“separator”、“split”之一，表示分割条，可以有若干个，text取值可以有所区别，为后续菜单项操作有用），回调时menu\_command.callback(key)；如果menu\_command.status = false时，将不会显示右键菜单项（menu\_command为NPPILY.Struct.WindowEventStruct结构的节点）

* + 1. 更新窗口右键菜单项NPPILY.WindowAttachEvent.UpdateMenuCommand (NPPILY.Struct.WindowStruct window, Array customMenus, Number bRedraw)

参数：NPPILY.Struct.WindowStruct window [NOT NULL] 视频窗口对象

Array customMenus [NULL]

Number bRedraw [NULL] 是否重绘右键菜单项，默认不重绘

调用：NPPILY.WindowAttachEvent.UpdateMenuCommand(window, customMenus, bRedraw)

返回值：NPPILY.Struct.ReturnValueStruct

说明：更新右键菜单项。bRedraw为1时，重绘右键菜单项。为0或省略时，更新菜单项。customMenus参考NPPILY.WindowAttachEvent.AppendMenuCommand

* + 1. 剔除窗口右键菜单项NPPILY.WindowAttachEvent.RemoveMenuCommand(NPPILY.Struct.WindowStruct window，Array customMenus)

参数：NPPILY.Struct.WindowStruct window [NOT NULL] 视频窗口对象

Array customMenus [NULL]

调用：NPPILY.WindowAttachEvent. RemoveMenuCommand (window, customMenus)

返回值：NPPILY.Struct.ReturnValueStruct

说明：删除窗口右键菜单项。当不传customMenus参数时，删除所有的菜单项。否则删除customMenus中key指定的菜单项。对于菜单项为[“-”，“separator”，“split”]的删除，将匹配text删除。customMenus参考NPPILY.WindowAttachEvent.AppendMenuCommand

* + 1. 云台控制NPPILY.PTZ.Control(string connectId, string puid, uint ptzIndex, NPPILY.Enum.PTZDirection direction, object options)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint ptzIndex [NOT NULL] 云台资源索引

NPPILY.Enum.PTZDirection direction [NOT NULL] 控制命令

Object options [NULL] 其他可选参数

{

degree(unit) 转动角度，可不传，默认为0（选传）

secondaryDevNo(uint) 辅助设备编号（选传）

presetPosNo(unit) 预置位编号（选传）

presetPosName(string) 预置位名称（选传）

}

调用：NPPILY.PTZ.Control(connectId, puid, ptzIndex, direction, options)

返回值：Object(NPPILY.Struct.ReturnValueStruct)

说明：1、options.degree(unit) 云台上下左右转动时选择传入

2、options.secondaryDevNo(uint) 操作辅助设备时选择传入，不可为空

3、options.presetPosNo(unit) 操作预置位时选择传入，不可为空

4、options.presetPosName(string) 操作设置预置位时选择传入，不可为空

* + 1. 获取窗口控制模式NPPILY.WindowControlMode.get(NPPILY.Struct.WindowStruct objWnd)

参数：NPPILY.Struct.WindowStruct objWnd [NOT NULL] 连接ID

调用：NPPILY.WindowControlMode.get(objWnd)

返回值：operator.response = 0（表示拖拽模式）或1（表示框选模式）

* + 1. 设置窗口控制模式NPPILY.WindowControlMode.set(NPPILY.Struct.WindowStruct objWnd, uint value)

参数：NPPILY.Struct.WindowStruct objWnd [NOT NULL] 连接ID

Uint value [NULL] 0（表示拖拽模式）或1（表示框选模式），缺省为0

调用：NPPILY.WindowControlMode.set(objWnd, value)

说明：在操作设置视频掩码时，请设置为框选模式

* + 1. 是否使能掩码NPPILY.EnableMask(NPPILY.Struct.WindowStruct objWnd, uint value)

参数：NPPILY.Struct.WindowStruct objWnd [NOT NULL] 连接ID

Uint value [NULL] 0/false非或1/true使能，缺省为非

调用：NPPILY.EnableMask(objWnd, value)

说明：对视频掩码进制设置时，请先进行使能控制，同时窗口控制模式也请设置为框选模式

* + 1. 点击掩码NPPILY.MaskClick(NPPILY.Struct.WindowStruct objWnd， uint x, uint y)

参数：NPPILY.Struct.WindowStruct objWnd [NOT NULL] 连接ID

Uint x [NULL] 视频窗口鼠标点击的x坐标

Uint y [NULL] 视频窗口鼠标点击的y坐标

调用：NPPILY.MaskClick(objWnd, x, y)

说明：窗口拖拽模式下，在窗口点击事件回调中需要调用此接口，然后视频掩码网格中才会对选中的某个小块着色

* + 1. 选择掩码区域NPPILY.MaskSelectRect(NPPILY.Struct.WindowStruct objWnd, uint bx, uint by, uint ex, uint ey)

参数：NPPILY.Struct.WindowStruct objWnd [NOT NULL] 连接ID

Uint bx [NULL] 视频窗口鼠标拖拽选择开始点的x坐标

Uint by [NULL] 视频窗口鼠标拖拽选择开始点的y坐标

Uint ex [NULL] 视频窗口鼠标拖拽选择结束点的x坐标

Uint ey [NULL] 视频窗口鼠标拖拽选择结束点的y坐标

调用：NPPILY.MaskSelectRect(objWnd, bx, by, ex, ey)

说明：窗口拖拽模式下，在窗口拖拽选择区域事件回调中需要调用此接口，然后视频掩码网格中才会对选中的某个区域着色

* + 1. 获取视频窗口中的掩码NPPILY.GetMask(NPPILY.Struct.WindowStruct objWnd)

参数：NPPILY.Struct.WindowStruct objWnd [NOT NULL] 连接ID

调用：NPPILY.GetMask(objWnd)

返回值：operator.response(Array<198个十进制数值>

说明：视频掩码控制时，获取得到窗口中的掩码

* + 1. 设置视频窗口中的掩码NPPILY.SetMask(NPPILY.Struct.WindowStruct objWnd, Array maskList)

参数：NPPILY.Struct.WindowStruct objWnd [NOT NULL] 连接ID

Array maskList [NOT NULL] 掩码数组，198个十进制数值

调用：NPPILY.SetMask(objWnd, maskList)

说明：视频掩码控制时，设置在窗口中显示掩码

* + 1. 开启设备的GPS信息接收通道NPPILY.StartGPSStream(String connectId, String puid, uint gpsIndex)

参数：String connectId [NOT NULL] 连接ID

String puid [NOT NULL] 含GPS模块的设备PUID

Uint gpsIndex [NOT NULL] GPS模块资源索引

调用：NPPILY.StartGPSStream(connectId, puid, gpsIndex)

返回值：NPPILY.Struct.ReturnValueStruct

说明：返回值节点rv=0时，那么response为GPS资源句柄（与gps\_data\_notigy事件回调中接收到的\_HANDLE一致，进行匹配划归）

* + 1. 停止设备的GPS信息接收通道NPPILY.StopGPSStream(String connectId, String puid, uint gpsIndex)

参数：String connectId [NOT NULL] 连接ID

String puid [NOT NULL] 含GPS模块的设备PUID

Uint gpsIndex [NOT NULL] GPS模块资源索引

调用：NPPILY.StopGPSStream(connectId, puid, gpsIndex)

返回值：NPPILY.Struct.ReturnValueStruct

* + 1. 高级查询中心存储文件NPPILY.QueryCSUFiles(String connectId, Object queryConditions)

参数：String connectId [NOT NULL] 连接ID

Object queryConditions[NOT NULL] 查询条件参数

调用：NPPILY.QueryCSUFiles(connectId, queryConditions)

返回值：NPPILY.Struct.ReturnValueStruct

说明：高级查询中心存储录像或图片；queryConditions结构如下（\*为必传参数）

{

csuPuid(string) \* 中心存储器PUID

csuIndex(uint) 中心存储单元资源索引，缺省为0

puid(string) 查询对象（设备）PUID

ivIndex(uint) 查询对象（摄像头）资源索引，缺省为0

objSets(Object|Array) 若干查询对象集合，在queryConditions.puid存在情况下，此节点无效

beginTime(UTC time) 查询开始时间（UTC秒，如1385380100），缺省为0

endTime(UTC time) 查询结束时间（UTC秒，如1385380100），缺省为当前时间

offset(uint) 查询开始索引，缺省从0开始

count(uint) 查询条数，缺省为200

streamType(NPPILY.Enum.NrcapStreamType) 限定查询存储的流类型，多个流类型之间使用英文竖线（|）分割，如（REALTIME|STORAGE），查图片使用PICTURE，缺省为STORAGE查录像

reason(string) 存储原因，多个原因之间使用英文竖线（|）分割，如（Manual|Plan），缺省为空字符串将查询所有符合条件录像

logicMode(string) 查询逻辑（AND与，OR或），缺省OR

domainRoad(string) 子域平台名（缺省空字符串，代表根平台）

}

其中queryConditions.objSets

-> 单个查询对象 如 = {puid: "151...", ivIndex: "0"}

-> 多个查询数组 如 = [{puid: "151...", ivIndex: "0"}, {puid: "151...", ivIndex: "0"}, ...]

返回值节点rv=NrcapError.NRCAP\_SUCCESS时，那么response为数组或结构，(Object|Array)(NPPILY.Struct.SCIVDateFileStruct)

* + 1. 获取平台存储下有存储文件的日期NPPILY.FetchCSUIVDate(String connectId, Object queryConditions)

参数：String connectId [NOT NULL] 连接ID

Object queryConditions[NOT NULL] 查询条件参数

调用：NPPILY.FetchCSUIVDate(connectId, queryConditions)

返回值：NPPILY.Struct.ReturnValueStruct

说明：快速索引查询中心存储录像或图片日期；queryConditions结构如下（\*为必传参数）

{

csuPuid(string) \* 中心存储器PUID

csuIndex(uint) 中心存储单元资源索引，缺省为0

puid(string) \* 查询对象（设备）PUID

ivIndex(uint) 查询对象（摄像头）资源索引，缺省为0

offset(uint) 查询开始索引，缺省从0开始

count(uint) 查询条数，缺省为200

streamType(NPPILY.Enum.NrcapStreamType) 限定查询存储的流类型，多个流类型之间使用英文竖线（|）分割，如（REALTIME|STORAGE），查图片使用PICTURE，缺省为STORAGE查录像

domainRoad(string) 子域平台名（缺省空字符串，代表根平台）

}

返回值节点rv=NrcapError.NRCAP\_SUCCESS时，那么response为数组Array(UTC timestamp)

* + 1. 按日期获取视频的平台存储文件NPPILY.FetchCSUIVDateFiles(String connectId, Object queryConditions)

参数：String connectId [NOT NULL] 连接ID

Object queryConditions[NOT NULL] 查询条件参数

调用：NPPILY.FetchCSUIVDateFiles(connectId, queryConditions)

返回值：NPPILY.Struct.ReturnValueStruct

说明：快速索引查询中心存储录像或图片日期；queryConditions结构如下（\*为必传参数）

{

csuPuid(string) \* 中心存储器PUID

csuIndex(uint) 中心存储单元资源索引，缺省为0

puid(string) \* 查询对象（设备）PUID

ivIndex(uint) 查询对象（摄像头）资源索引，缺省为0

datetime(UTC timestamp) \* 以UTC时间（如1385380100）查询，必选，应该为通过NPPILY.FetchCSUIVDate获取到的UTC日期时间

offset(uint) 查询开始索引，缺省从0开始

count(uint) 查询条数，缺省为200

streamType(NPPILY.Enum.NrcapStreamType) 限定查询存储的流类型，多个流类型之间使用英文竖线（|）分割，如（REALTIME|STORAGE），查图片使用PICTURE，缺省为STORAGE查录像

domainRoad(string) 子域平台名（缺省空字符串，代表根平台）

}

返回值节点rv=NrcapError.NRCAP\_SUCCESS时，那么response为数组或结构，(Object|Array)(NPPILY.Struct.SCIVDateFileStruct)

* + 1. 录像回放点播NPPILY.PlayVod(String connectId, NPPILY.Struct.WidnowStruct objWnd, Object options)

参数：String connectId [NOT NULL] 连接ID

NPPILY.Struct.WidnowStruct objWnd [NOT NULL] 窗口信息结构

Object options [NOT NULL] 点播相关参数

调用：NPPILY.PlayVod(connectId, objWnd, options)

返回值：NPPILY.Struct.ReturnValueStruct

说明：快速索引查询中心存储录像或图片日期；options结构如下（\*为必传参数）

{

type(NPPILY.Enum.StorageFileType) \* 存储文件类型

csuPuid(string) 中心存储器PUID

csuIndex(uint) 中心存储单元资源索引，缺省为0

puid(string) 点播对象（设备）PUID

ivIndex(uint) 点播对象（摄像头）资源索引，缺省为0

sgIndex(uint) 前端点播时存储器资源索引，缺省为0

diskLetter(string) 前端存储器磁盘盘符，比如A | B …， 磁盘盘符通过CEFS查询存储文件时，返回值属性对应DiskLetter

fileFullPath(string) 点播的录像文件全路径，如D:/StoragePath/123.avi、/disk/storage/0/123.avi等

speed(uint) 播放速度，缺省0为正常速度

direction(uint) 播放时间轴方向，1正向（缺省），0倒向

relativeStartTime(UTC timestamp) 相对播放开始时间（缺省为0秒）

beginTime(UTC timestamp) 录像开始时间

endTime(UTC timestamp) 录像结束时间

}

1. 点播平台存储录像，type=NPPILY.Enum.StorageFileType.Platform时，可选下列参数传入csuPuid、csuIndex、puid、ivIndex、fileFullPath、speed、direction、relativeStartTime
2. 点播前端CEFS录像，type=NPPILY.Enum.StorageFileType.CEFS时，可选下列参数传入puid、sgIndex、ivIndex、diskLetter 、speed、direction、beginTime、endTime

3、点播前端FAT32（EXT4等）录像，type=NPPILY.Enum.StorageFileType.FAT32时，可选下列参数传入puid、sgIndex、fileFullPath、speed、direction、relativeStartTime

* + 1. 停止点播 NPPILY.StopVod(NPPILY.Struct.WindowStruct objWnd)

参数：NPPILY.Struct.WindowStruct objWnd[NOT NULL]

调用：NPPILY.StopVideo(objWnd)

返回值：NPPILY.Struct.ReturnValueStruct

说明：停止一路回放视频

* + 1. CEFS查询有存储文件的日期NPPILY.FetchCEFSDate(String connectId, Object queryConditions)

参数：String connectId [NOT NULL] 连接ID

Object queryConditions[NOT NULL] 查询条件参数

调用：NPPILY.FetchCEFSDate(connectId, queryConditions)

返回值：NPPILY.Struct.ReturnValueStruct

说明：快速索引查询CEFS存储数据日期；queryConditions结构如下（\*为必传参数）

{

puid(string) \* 查询对象设备PUID

requestID(NPPILY.Enum.CEFSRequestID) \* 请求命令

sgIndex(uint) 前端存储器资源索引，一般为0（缺省）

diskLetter(string) 查询磁盘的盘符，缺省为空字符串

channelSets(Uint|Array) 查询通道号，从0开始，依次表示第一路...视频，缺省为0，多个通道号使用数组，如new Array(0, 1, ..)

}

- queryConditions.channelSets 单个数值，如0或[0]

requestID = NPPILY.Enum.CEFSRequestID.QueryRecordDate | QueryPictureDate

- queryConditions.channelSets不传

requestID = NPPILY.Enum.CEFSRequestID.QueryUserLogDate

- queryConditions.channelSets 单个数值或数组，如0或[0, 1, 2, ...]

requestID = NPPILY.Enum.CEFSRequestID.QueryAlarmEventDate

返回值节点rv=NrcapError.NRCAP\_SUCCESS时，那么response为数组Array(UTC timestamp)

* + 1. 查询CEFS存储文件（适用快速查询和高级查询）NPPILY.FetchCEFSFiles(String connectId, Object queryConditions)

参数：String connectId [NOT NULL] 连接ID

Object queryConditions[NOT NULL] 查询条件参数

调用：NPPILY.FetchCEFSFiles(connectId, queryConditions)

返回值：NPPILY.Struct.ReturnValueStruct

说明：快速索引查询CEFS存储数据日期；queryConditions结构如下（\*为必传参数）

{

puid(string) \* 查询对象设备PUID

requestID(NPPILY.Enum.CEFSRequestID) \* 请求命令

beginTime(UTC timstamp) \* 起始查询时间，缺省为0，如1325376000

endTime(UTC timstamp) \* 结束查询时间，缺省为0，如1325386000

sgIndex(uint) 前端存储器资源索引，一般为0（缺省）

diskLetter(string) 查询磁盘的盘符，缺省为空字符串

channelSets(Uint|Array) 查询通道号，从0开始，依次表示第一路...视频，缺省为0，多个通道号使用数组，如new Array(0, 1, ..)

reasonSets(String|Array) 存储原因

typeSets(String|Array) 数据类型

}

- queryConditions.diskLetter磁盘盘符，可以根据NPPILY.SG\_GetDiskInfo(connectId, puid, sgIndex)查询出磁盘信息，然后找到状态为OK的磁盘盘符查询，如果查询不到磁盘信息，那么缺省queryConditions.diskLetter查前端存储文件，很可能没有数据。

- requestID = NPPILY.Enum.CEFSRequestID.QueryRecord 快速索引查询

=> queryConditions.channelSets一个数值，如0或[0]

=> queryConditions.reasonSets传值也无效，将查所有原因的

- requestID = NPPILY.Enum.CEFSRequestID.QueryLinkActionRecord 高级查询

=> queryConditions.channelSets一个数值，如0或[0]

=> queryConditions.reasonSets<(String|Array)(NPPILY.Enum.CEFSRecordReason)>，如Schedule或[Schedule, Manual]，缺省将查所有原因的

- requestID = NPPILY.Enum.CEFSRequestID.QueryPicture 快速索引查询

=> queryConditions.channelSets一个数值，如0或[0]

=> queryConditions.reasonSets传值也无效，将查所有原因的

- requestID = NPPILY.Enum.CEFSRequestID.QueryLinkActionPicture 高级查询

=> queryConditions.channelSets单个数值或数组，如0或1或[0, 1, ...]

=> queryConditions.reasonSets<(String|Array)(NPPILY.Enum.CEFSPictureReason)>，如Schedule或[Schedule, Manual]，缺省将查所有原因的

- requestID = NPPILY.Enum.CEFSRequestID.QueryUserLog

=> 高级查询时: queryConditions.typeSets<(String|Array)(NPPILY.Enum.CEFSUserLogType)>，如UserLogin或UserLogin, UserLogout]

=> 快速索引查询时：queryConditions.typeSets不必传值，将查所有类型的

- requestID = NPPILY.Enum.CEFSRequestID.QueryAlarmEvent

queryConditions.channelSets单个数值或数组，如0或1或[0, 1, ...]

=> 高级查询时: queryConditions.typeSets<(String|Array)(NPPILY.Enum.CEFSAlarmEventType)>，如AlertIn或[AlertIn, MotionDetected]

=> 快速索引查询时：queryConditions.typeSets不必传值，将查所有类型的

- requestID = NPPILY.Enum.CEFSRequestID.QueryGPSData

返回值节点rv=NrcapError.NRCAP\_SUCCESS时，那么response为数组Array(XXX)，XXX结构如下：

- requestID = NPPILY.Enum.CEFSRequestID.QueryRecord

- requestID = NPPILY.Enum.CEFSRequestID.QueryLinkActionRecord

=> NPPILY.Struct.CEFSRecordStruct

- requestID = NPPILY.Enum.CEFSRequestID.QueryPicture

- requestID = NPPILY.Enum.CEFSRequestID.QueryLinkActionPicture

=> NPPILY.Struct.CEFSPictureStruct

- requestID = NPPILY.Enum.CEFSRequestID.QueryUserLog

=> NPPILY.Struct.CEFSUserLogStruct

- requestID = NPPILY.Enum.CEFSRequestID.QueryAlarmEvent

=> NPPILY.Struct.CEFSAlarmEventStruct

- requestID = NPPILY.Enum.CEFSRequestID.QueryGPSData

=> NPPILY.Struct.CEFSGPSDataStruct

* + 1. 获取逻辑分组NPPILY.GetLogicGroups(string connectId)

参数：String connectId [NOT NULL] 连接ID

调用：NPPILY.GetLogicGroups(connectId)

返回值：NPPILY.Struct.ReturnValueStruct

说明：成功返回时，response节点为Array(NPPILY.Struct.LogicGroupStruct)

* + 1. 获取逻辑分组子节点NPPILY.GetLogicGroupNodes(string connectId, object options)

参数：String connectId [NOT NULL] 连接ID

Object options[NOT NULL] 其他可选参数

调用：NPPILY.GetLogicGroupNodes(connectId, options)

返回值：NPPILY.Struct.ReturnValueStruct

说明：返回值rv节点为常量NrcapError.NRCAP\_SUCCESS时，response节点对应Array(NPPILY.Struct.LogicGroupNodeStruct)；options结构如下：

{

logicGroupIndex(uint<NPPILY.Struct.LogicGroupStruct:index>) \* 逻辑分组索引

logicGroupNodeIndex(uint) 逻辑分组节点索引，缺省为0

offset(uint) 分页查询开始条数，缺省为0

count(uint) 分页查询最大条数，缺省为4294967295（一个较大的数值）

fetchMode(string) 获取模式，缺省为step（递归分部获取） | once（一次性获取全部节点）

}

注意：

- 获取逻辑分组节点或节点下节点

- logicGroupNodeIndex表示当前节点（父节点），根节点的父节点索引为0，即缺省获取逻辑分组直接子节点

- fetchMode = 'step'时，如果logicGroupNodeIndex属性存在，就获取这个节点下面的节点，不递归。如果不存在此属性，则获取所有节点

- fetchMode = 'once'时，无论logicGroupNodeIndex存不存在，则获取logicGroupIndex下面的所有节点

- count 默认为一个足够大的数，但是如果节点过大的话，建议配合offset进行分页获取

* + 1. 获取逻辑分组节点下资源NPPILY.GetLogicGroupResource(string connectId, object options)

参数：String connectId [NOT NULL] 连接ID

Object options[NOT NULL] 其他可选参数

调用：NPPILY.GetLogicGroupResource(connectId, options)

返回值：NPPILY.Struct.ReturnValueStruct

说明：返回值rv节点为常量NrcapError.NRCAP\_SUCCESS时，response节点对应Array(NPPILY.Struct.LogicGroupResourceStruct)；options结构如下：

{

logicGroupIndex(uint<NPPILY.Struct.LogicGroupStruct:index>) \* 逻辑分组索引

logicGroupNodeIndex(uint) 逻辑分组节点索引，缺省为0

offset(uint) 分页查询开始条数，缺省为0

count(uint) 分页查询最大条数，缺省为4294967295（一个较大的数值）

fetchMode(string) 获取模式，缺省为step（递归分部获取） | once（一次性获取全部节点）

}

注意：

- 获取逻辑分组节点或节点下节点资源

- 约定节点下的资源只有摄像头，没有PU以及其他资源，但为了扩然依然保留Res.Type属性

- logicGroupNodeIndex表示当前节点（父节点），根节点的父节点索引为0，即缺省获取逻辑分组直接子节点资源

- fetchMode = 'step'时，如果logicGroupNodeIndex属性存在，就获取这个节点下面的节点资源，不递归。如果不存在此属性，则获取所有节点资源

- fetchMode = 'once'时，无论logicGroupNodeIndex存不存在，则获取logicGroupIndex下面的所有节点资源

- count 默认为一个足够大的数，但是如果节点过大的话，建议配合offset进行分页获取

* + 1. 资源描述配置NPPILY.CommonResConfig(string connectId, string puid, NPPILY.Enum.PuResourceType resType, uint resIdx, string action, NPPILY.Struct.CommonResDescriptionStruct paramObj)

参数：String connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

NPPILY.Enum.PuResourceType resType [NOT NULL] 资源类型

Uint resIdx [NOT NULL] 资源索引

String action [NULL] get为获取资源描述，set为设备资源描述

NPPILY.Struct.CommonResDescriptionStructparamObj 资源描述结构信息，action为set时使用

调用：NPPILY.CommonResConfig(connectId, puid, resType, resIdx, action, paramObj)

返回值：NPPILY.Struct.ReturnValueStruct

说明：参数action=“get”，返回值rv节点为常量NrcapError.NRCAP\_SUCCESS时，response节点为Array(NPPILY.Struct.CommonResDescriptionStruct)；设置资源描述信息时，puid、resType、resIdx等应该与paramObj中相关属性是一致的。

* + 1. 开启设备喊话NPPILY.CallTalkControl.StartCall(string connectId, string puid, uint oaIndex)

参数：String connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint oaIndex [NOT NULL] 音频输出（OA）资源索引

调用：NPPILY.CallTalkControl.StartCall(connectId, puid, oaIndex)

返回值：NPPILY.Struct.ReturnValueStruct

说明：返回值rv节点为常量NrcapError.NRCAP\_SUCCESS时，表示喊话成功（下面开启对讲也是一样的），response节点为通道流句柄；其他错误码：

NRCAP\_FAILED操作失败、NRCAP\_ERROR\_UNSUPPORT属性不支持、NRCAP\_ERROR\_PU\_OFFLINE设备不在线，肯定喊话会失败、NRCAP\_ERROR\_HANDLE\_ERROR设备句柄出错，可能是平台没有此设备信息、

NRCAP\_ERROR\_CALLTALK\_OCCUPIED音频输出资源被（可能为其他用户）占用、NRCAP\_ERROR\_CALLTALK\_INCOMPATIBLE开启对讲了，喊话与对讲应互斥使用、NRCAP\_ERROR\_CALLTALK\_EXISTED已经开启喊话了

* + 1. 开启设备双向对讲NPPILY.CallTalkControl.StartTalk(string connectId, string puid, uint oaIndex)

参数：String connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint oaIndex [NOT NULL] 音频输出（OA）资源索引

调用：NPPILY.CallTalkControl.StartTalk(connectId, puid, oaIndex)

返回值：NPPILY.Struct.ReturnValueStruct

说明：参考开启喊话说明；另外应做到开启音频或喊话与对讲互斥！

* + 1. 获取设备的喊话或对讲状态NPPILY.CallTalkControl.GetStatus(string connectId, string puid, uint oaIndex)

参数：String connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint oaIndex [NOT NULL] 音频输出（OA）资源索引

调用：NPPILY.CallTalkControl.GetStatus(connectId, puid, oaIndex)

返回值：NPPILY.Struct.ReturnValueStruct

说明：开启喊话或对讲前后，可以根据此函数检测某个设备是否在喊话或对讲状态，返回值response结构为

{

puid: ?,

oaIndex: ?,

call: ?, // - 是否在喊话

talk: ?, // - 是否在双向对讲

oaStreamHandle: ? // - 代表喊话或对讲通道的流句柄，为空时，表示初始或开启不成功状态

}

* + 1. 获取所有正在喊话对讲的设备状态NPPILY.CallTalkControl.GetAllRunningStatus(string connectId)

参数：String connectId [NOT NULL] 连接ID

调用：NPPILY.CallTalkControl.GetAllRunningStatus(connectId, puid, oaIndex)

返回值：NPPILY.Struct.ReturnValueStruct

说明：开启喊话或对讲前后，可以根据此函数获取到所有在喊话或对讲的状态，返回值response为数组结构：

[{puid: ?, oaIndex: ?, call: ?, talk: ?, oaStreamHandle: ?}, ...];具体属性的定义请参考上个接口中说明

* + 1. 停止设备的喊话或对讲NPPILY.CallTalkControl.Stop(string connectId, string puid, uint oaIndex)

参数：String connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint oaIndex [NOT NULL] 音频输出（OA）资源索引

调用：NPPILY.CallTalkControl.Stop(connectId, puid, oaIndex)

返回值：NPPILY.Struct.ReturnValueStruct

说明：开启喊话或双向对讲通道后，共用此函数来关闭通道

* + 1. 下载中心存储录像或图片NPPILY.Download.StartCSUFileDownload(string connectId, string csuPuid, string fileAllPath, string localSaveAsFile, unit csuIndex)

参数：String connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

String fileAllPath [NOT NULL] 中心存储文件的全路径名

String localSaveAsFile [NOT NULL] 下载到本地保存的全路径名，如C:/vod.avi、C:/image.bmp等

Uint csuIndex [NULL] 中心存储单元资源索引，一般为0，可省略

调用：NPPILY.Download.StartCSUFileDownload(connectId, csuPuid, fileAllPath, localSaveAsFile, csuIndex)

返回值：NPPILY.Struct.ReturnValueStruct

说明：返回值rv节点为常量NrcapError.NRCAP\_SUCCESS时，response节点代表下载通道（流）句柄，这个标识很重要，要注意保存下来，在NPPILY.Enum.NCObjectNotify.stream\_status\_notify的回调函数中，匹配此句柄可以得到对应的下载状态信息，keyData结构为

{

download\_length: ?, // - 已经下载的字节数

download\_time: ?, // - 已经下载的累计时间

bit\_rate: ? // - 下载速率，单位KBps

}

* + 1. 停止下载平台存储录像或图片NPPILY.Download.StopCSUFileDownload(string connectId, String dlStreamHandle)

参数：String connectId [NOT NULL] 连接ID

String dlStreamHandle [NOT NULL] 下载通道（流）句柄

调用：NPPILY.Download.StopCSUFileDownload(connectId, dlStreamHandle)

* + 1. 下载前端存储录像NPPILY.Download.StartCEFSVodFileDownload(string connectId, string puid, uint sgIndex, string disk Letter, utctime beginTime, utctime endTime, uint ivIndex, string localSaveAsFile)

参数：String connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器（SG）资源索引

String diskLetter [NOT NULL] 前端存储器磁盘盘符，比如A | B …， 磁盘盘符通过CEFS查询存储文件时，返回值属性对应DiskLetter

UTCTime beginTime [NOT NULL] 录像开始时间，UTC秒

UTCTime endTime [NOT NULL] 录像结束时间，UTC秒

Uint ivIndex [NOT NULL] 视频（IV）资源索引 String localSaveAsFile [NOT NULL] 下载到本地保存的全路径名，如C:/vod.avi等

调用：NPPILY.Download.StartCEFSVodFileDownload(connectId, puid, sgIndex, disk Letter, beginTime, endTime, ivIndex, localSaveAsFile)

返回值：NPPILY.Struct.ReturnValueStruct

说明：返回值rv节点为常量NrcapError.NRCAP\_SUCCESS时，response节点代表下载通道（流）句柄，这个标识很重要，要注意保存下来，在NPPILY.Enum.NCObjectNotify.stream\_status\_notify的回调函数中，匹配此句柄可以得到对应的下载状态信息，keyData结构为

{

download\_length: ?, // - 已经下载的字节数

download\_time: ?, // - 已经下载的累计时间

bit\_rate: ? // - 下载速率，单位KBps

}

* + 1. 下载前端存储图片NPPILY.Download.StartCEFSImageFileDownload(string connectId, string puid, uint sgIndex, string diskLetter, UTCTime snapshotTime, uint noInSecond, uint ivIndex, string localSaveAsFile)

参数：String connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器（SG）资源索引

String diskLetter [NOT NULL] 前端存储器磁盘盘符，比如A | B …， 磁盘盘符通过CEFS查询存储文件时，返回值属性对应DiskLetter

UTCTime snapshotTime [NOT NULL] 抓拍时间，UTC秒

Uint noInSecond [NOT NULL] 抓拍图片秒内编号，默认为0，同一秒内可以有多个抓图

Uint ivIndex [NOT NULL] 视频（IV）资源索引 String localSaveAsFile [NOT NULL] 下载到本地保存的全路径名，如C:/vod.avi等

调用：NPPILY.Download.StartCEFSImageFileDownload(connectId, puid, sgIndex, snapshotTime, noInSecond, ivIndex, localSaveAsFile)

返回值：NPPILY.Struct.ReturnValueStruct

说明：返回值rv节点为常量NrcapError.NRCAP\_SUCCESS时，response节点代表下载通道（流）句柄，这个标识很重要，要注意保存下来，在NPPILY.Enum.NCObjectNotify.stream\_status\_notify的回调函数中，匹配此句柄可以得到对应的下载状态信息，keyData结构为

{

download\_length: ?, // - 已经下载的字节数

download\_time: ?, // - 已经下载的累计时间

bit\_rate: ? // - 下载速率，单位KBps

}

* + 1. 停止下载前端存储录像或图片NPPILY.Download.StopCEFSFileDownload(string connectId, String dlStreamHandle)

参数：String connectId [NOT NULL] 连接ID

String dlStreamHandle [NOT NULL] 下载通道（流）句柄

调用：NPPILY.Download.StopCEFSFileDownload(connectId, dlStreamHandle)

* + 1. 按连接ID清除所有下载NPPILY.Download.Clear(string connectId)

参数：String connectId [NOT NULL] 连接ID

调用：NPPILY.Download.Clear(connectId)

* + 1. 获取存储GPS数据的资源NPPILY.GPSData.GetGPSResFromCSU(string connectId, object options)

参数：String connectId [NOT NULL] 连接ID

Object options [NOT NULL] 结构如下

{

csuPuid(string) \* 中心存储器PUID

csuIndex(uint) 中心存储器资源索引，缺省为0

offset(uint) 分页偏移量，缺省为0

count(uint) 分页查询最大条数，缺省为200

}

调用：NPPILY.GPSData.GetGPSResFromCSU(connectId, options);

返回值：成功时，operator.response节点结构为Array<NPPILY.Struct.GPSDataStorageResourceStruct>

* + 1. 添加存储GPS数据的资源NPPILY.GPSData.AddGPSResToCSU(string connectId, object options)

参数：String connectId [NOT NULL] 连接ID

Object options [NOT NULL] 结构如下

{

csuPuid(string) \* 中心存储器PUID

csuIndex(uint) 中心存储器资源索引，缺省为0

serialNo(string) 缺省为当前登录平台用户名

serialToken(string) 缺省为当前登录平台用户密码

puid(string) 设备PUID

resType(NPPILY.Enum.PuResourceType) 资源类型，一般为GPS

resIdx(uint) 资源索引，缺省为0

optID(string) 操作ID，缺省为“CTL\_SCHEDULER\_StartStream”

value(NPPILY.Enum.NrcapStreamType) 流类型

resourceSets(Array(NPPILY.Struct.GPSDataStorageResourceStruct)) 资源集合，当上面puid不存在时才会使用resourceSets，puid存在视为单项操作

}

调用：NPPILY.GPSData.AddGPSResToCSU(connectId, options);

* + 1. 删除存储GPS数据的资源NPPILY.GPSData.RemoveGPSResFromCSU(string connectId, object options)

说明：参考“添加存储GPS数据的资源”

* + 1. 查询设备最新GPS数据NPPILY.GPSData.QueryLastGPSDatas(string connectId, object options)

参数：String connectId [NOT NULL] 连接ID

Object options [NOT NULL] 结构如下

{

csuPuid(string) \* 中心存储器PUID

csuIndex(uint) 中心存储器资源索引，缺省为0

puid(string) 设备PUID

resType(NPPILY.Enum.PuResourceType) 资源类型，一般为GPS

resIdx(uint) 资源索引，缺省为0

resourceSets(Array) 资源集合，每项结构为{puid:?,resType:?,resIdx:?}。当上面puid不存在时才会使用resourceSets，puid存在视为单项操作

}

调用：NPPILY.GPSData.QueryLastGPSDatas(connectId, options);

返回值：成功时，operator.response节点结构为Array<NPPILY.Struct.CSUGPSDataFileStruct>

* + 1. 查询设备历史GPS数据NPPILY.GPSData.QueryHistoryGPSDatas(string connectId, object options)

参数：String connectId [NOT NULL] 连接ID

Object options [NOT NULL] 结构如下

{

csuPuid(string) \* 中心存储器PUID

csuIndex(uint) 中心存储器资源索引，缺省为0

beginTime(UTC time) 查询开始时间，UTC时间秒，缺省为0

endTime(UTC time) 查询结束时间，UTC时间秒，缺省为当前时间

offset(uint) 分页偏移量，缺省为0

count(uint) 分页查询最大条数，缺省为200

puid(string) 设备PUID

resType(NPPILY.Enum.PuResourceType) 资源类型，一般为GPS

resIdx(uint) 资源索引，缺省为0

resourceSets(Array) 资源集合，每项结构为{puid:?,resType:?,resIdx:?}。当上面puid不存在时才会使用resourceSets，puid存在视为单项操作

}

调用：NPPILY.GPSData.QueryHistoryGPSDatas(connectId, options);

返回值：成功时，operator.response节点结构为Array<NPPILY.Struct.CSUGPSDataFileStruct>

* + 1. 查询某GPS保存历史数据的日期NPPILY.GPSData.QueryGPSDates(string connectId, object options)

参数：String connectId [NOT NULL] 连接ID

Object options [NOT NULL] 结构如下

{

csuPuid(string) \* 中心存储器PUID

csuIndex(uint) 中心存储器资源索引，缺省为0

offset(uint) 分页偏移量，缺省为0

count(uint) 分页查询最大条数，缺省为200

puid(string) 设备PUID

resType(NPPILY.Enum.PuResourceType) 资源类型，一般为GPS

resIdx(uint) 资源索引，缺省为0

}

调用：NPPILY.GPSData.QueryGPSDates(connectId, options);

返回值：成功时operator.response结构为Array<UTC Time>，如["1393776000", "1393862400", ...]

* + 1. 查询某GPS某日期的历史数据NPPILY.GPSData.QueryGPSDateFiles(string connectId, object options)

参数：String connectId [NOT NULL] 连接ID

Object options [NOT NULL] 结构如下

{

csuPuid(string) \* 中心存储器PUID

csuIndex(uint) 中心存储器资源索引，缺省为0

offset(uint) 分页偏移量，缺省为0

count(uint) 分页查询最大条数，缺省为200

date(UTC Time) UTC时间秒，一般为NPPILY.GPSData.QueryGPSDates查询返回的时间

puid(string) 设备PUID

resType(NPPILY.Enum.PuResourceType) 资源类型，一般为GPS

resIdx(uint) 资源索引，缺省为0

}

调用：NPPILY.GPSData.QueryGPSDateFiles(connectId, options);

返回值：成功时operator.response结构为Array<NPPILY.Struct.CSUGPSDataFileStruct>

* + 1. 快速获取配置项NPPILY.Config.GetSimple(string connectId, string puid, NPPILY.Enum.PuResourceType resType, unit resIdx, string configID, NPPILY.Enum.NrcapStreamType streamType, object customParams)

参数：String connectId [NOT NULL] 连接ID

String puid[NOT NULL] 设备PUID

NPPILY.Enum.PuResourceType resType[NOT NULL] 资源类型

Unit resIdx[NOT NULL] 资源类型

String configID[NOT NULL] 设备配置项命令名称

NPPILY.Enum.NrcapStreamType streamType[NULL] 流类型

Object customParams[NULL] 其他可选参数

调用：NPPILY.Config.GetSimple(connectId, puid, resType, resIdx, configID, streamType, customParams)

返回值: operator::Object(NPPILY.Struct.ReturnValue) 下同

说明：

1、返回值operator.rv==NrcapError.NRCAP\_SUCCESS，那么可认为获取成功，operator.response有效

2、configID可参考《ConfigID7.doc》中配置命令，下同

3、customParams结构如下，

{

returnFlag(uint) 是否自行解析（0内部解析，1留给上层自行解析），缺省为0

}

4、customParams.returnFlag == 0，operator.response一般为配置项的具体取值；customParams.returnFlag == 1，operator.response为JSON格式对象，对应响应的全部信息，上层需自行处理

示例：

1、获取视频亮度var operator = NPPILY.Config.GetSimple(connectId, “151000003727830108”, NPPILY.Enum.PuResourceType.VideoIn, 0, “CFG\_IV\_Brightness”); 其中connectId?处应根据实际具体取值（下同），那么operator可能的结构为{“rv”: 0, “response”: “55”}

2、获取视频亮度var operator = NPPILY.Config.GetSimple(connectId, “151000003727830108”, NPPILY.Enum.PuResourceType.VideoIn, 0, “CFG\_IV\_Brightness”, “”, {returnFlag: 1}); 那么operator可能的结构为{"rv":0,"response":{"Msg":{"Name":"CUCommonMsgRsp","Cmd":{"NUErrorCode":"0","Prio":"1","Type":"GET","DstRes":{"Type":"IV","ErrorCode":"0","Setable":"1","OptID":"CFG\_IV\_Brightness","Idx":"0","Param":{"Value":"55"}}}}}}

* + 1. 快速设置配置项NPPILY.Config.SetSimple(string connectId, string puid, NPPILY.Enum.PuResourceType resType, unit resIdx, string configID, NPPILY.Enum.NrcapStreamType streamType, variant value, object customParams)

参数：String connectId [NOT NULL] 连接ID

String puid[NOT NULL] 设备PUID

NPPILY.Enum.PuResourceType resType[NOT NULL] 资源类型

Unit resIdx[NOT NULL] 资源类型

String configID[NOT NULL] 设备配置项命令名称

NPPILY.Enum.NrcapStreamType streamType[NOT NULL] 流类型,必须传值没有就为空字符串或null

Variant value[NOT NULL] 取值类型可以为string|uint|object等

Object customParams[NULL] 其他可选参数

调用：NPPILY.Config.SetSimple(connectId, puid, resType, resIdx, configID, streamType, value， customParams)

说明：返回值operator.rv==NrcapError.NRCAP\_SUCCESS，那么可认为设置成功，否则是失败的

示例：设置视频亮度var operator = NPPILY.Config.SetSimple(connectId, “151000003727830108”, NPPILY.Enum.PuResourceType.VideoIn, 0, “CFG\_IV\_Brightness”， null, 77); 那么operator可能的结构为{"rv":0,"response":{"Type":"IV","ErrorCode":"0","OptID":"CFG\_IV\_Brightness","Idx":"0"}}

* + 1. 高级获取配置项NPPILY.Config.GetComplex(string connectId, object reqParams)

参数：String connectId [NOT NULL] 连接ID

Object reqParams [NOT NULL] 可选请求参数

{

puid(string) \* 设备PUID

resType(NPPILY.Enum.PuResourceType) \* 资源类型

resIdx(uint) \* 资源索引

configID(string) \* 配置命令

streamType(NPPILY.Enum.NrcapStreamType) 流类型

returnType(string) 返回类型 xml | json(default)

returnFlag(uint) 0(default)返回有用的节点 | 1自行解析，当returnType='xml'时，视returnFlag===1处理

cmdType(NPPILY.Enum.CmdType) 默认GET

domainRoad(string) 级联子域名称，根平台默认为空字符串

}

调用：NPPILY.Config.GetComplex(connectId, reqParams);

说明：1、returnType为operator.response返回类型，如果为xml，那 么将返回未经处理xml响应报文，上层自行处理

2、对于响应协议中具体值属于Param属性或子节点的，那么一般会返回Param节点

示例：1、获取视频亮度var operator = NPPILY.Config.GetComplex

(

connectId,

{

puid: "151038400117512284",

resType: NPPILY.Enum.PuResourceType.VideoIn,

resIdx: 0,

configID: "CFG\_IV\_Brightness",

returnFlag: 0

}

); operator可能的取值结构为{“rv”: 0, “response”: {“Value”：“55”}}

2、支持的流类型var operator = NPPILY.Config.GetComplex

(

connectId,

{

puid: "151038400117512284",

resType: NPPILY.Enum.PuResourceType.VideoIn,

resIdx: 0,

configID: "CFG\_IV\_SupportedStreamTypeSets",

returnFlag: 0

}

); operator可能的取值结构为{"rv":0,"response":{"StreamType":["REALTIME","STORAGE","MOBILE2G"]}}

* + 1. 高级设置配置项NPPILY.Config.GetComplex(string connectId, object setParams)

参数：String connectId [NOT NULL] 连接ID

Object setParams [NOT NULL] 可选设置参数

{

puid(string) \* 设备PUID

resType(NPPILY.Enum.PuResourceType) \* 资源类型

resIdx(uint) \* 资源索引

configID(string) \* 配置命令

param(xml string) \* 含<Param ...>...</Param>

streamType(NPPILY.Enum.NrcapStreamType) 流类型

returnType(string) xml | json(default)

returnFlag(uint) 0(default)返回有用的节点 | 1自行解析，当returnType='xml'时，视returnFlag===1处理

cmdType(NPPILY.Enum.CmdType) 默认SET

domainRoad(string) 级联子域名称，根平台默认为空字符串

}

调用：NPPILY.Config.SetComplex(connectId, setParams);

说明：setParams.param为设置参数xml报文格式的，一般含Param节点

示例：设置云台转动速度var operator = NPPILY.Config.SetComplex

(

connectId,

{

puid: "151038400117512284",

resType: NPPILY.Enum.PuResourceType.PTZ,

resIdx: 0,

configID: "CFG\_PTZ\_Speed",

param: "<Param Value=\"55\" />"

}

); operator可能的取值结构为{"rv":0,"response":{"Type":"PTZ","ErrorCode":"0","OptID":"CFG\_PTZ\_Speed","Idx":"0"}}

* + 1. 发送获取控制命令项NPPILY.Control.CommonGet(string connectId, object reqParams)

参数：String connectId [NOT NULL] 连接ID

Object reqParams [NOT NULL] 可选请求参数

{

puid(string) \* 设备PUID

resType(NPPILY.Enum.PuResourceType) \* 资源类型

resIdx(uint) \* 资源索引

controlID(string) \* 控制命令

param(xml string) \* 含<Param ...>...</Param>

streamType(NPPILY.Enum.NrcapStreamType) 流类型

returnType(string) xml | json(default)

returnFlag(uint) 0(default)返回有用的节点 | 1自行解析，当returnType='xml'时，视returnFlag===1处理

cmdType(NPPILY.Enum.CmdType) 默认CTL

xmlObjSets(xml string) 其他可能对象参数（如查询中心存储文件时需要，含<ObjSets>...</ObjSets>）

domainRoad(string) 级联子域名称，根平台默认为空字符串

}

调用：NPPILY.Control.CommonGet(connectId, reqParams);

说明：reqParams.controlID为控制命令名称，具体参考《ControlID7.doc》中定义的协议

* + 1. 发送设置控制命令项NPPILY.Control.CommonSet(string connectId, object reqParams)

参数：String connectId [NOT NULL] 连接ID

Object setParams [NOT NULL] 可选请求参数

{

puid(string) \* 设备PUID

resType(NPPILY.Enum.PuResourceType) \* 资源类型

resIdx(uint) \* 资源索引

controlID(string) \* 控制命令

param(xml string) \* 含<Param ...>...</Param>

streamType(NPPILY.Enum.NrcapStreamType) 流类型

returnType(string) xml | json(default)

returnFlag(uint) 0(default)返回有用的节点 | 1自行解析，当returnType='xml'时，视returnFlag===1处理

cmdType(NPPILY.Enum.CmdType) 默认CTL

xmlObjSets(xml string) 其他可能对象参数（如查询中心存储文件时需要，含<ObjSets>...</ObjSets>）

domainRoad(string) 级联子域名称，根平台默认为空字符串

}

调用：NPPILY.Control.CommonSet(connectId, setParams);

说明：1、reqParams.controlID为控制命令名称，具体参考《ControlID7.doc》中定义的协议

2、注意NPPILY.Control.CommonGet、NPPILY.Control.CommonSet实现控制功能上都是一样的，只是这里人为的封装区别出来了，获取的控制命令使用NPPILY.Control.CommonGet，设置的控制命令使用NPPILY.Control.CommonSet，当然具体开发时看需要可以特定某个接口进行调用

* + 1. 多命令联合请求公共接口NPPILY.CommonRequest(string connectId, object options)

参数：string connectId [NOT NULL] 连接ID

Object options [NOT NULL] 可选参数

{

cmdType(NPPILF.Enum.CmdType) \* 请求命令类型，所有请求项符合此类型

puid(string) \* 请求对象设备PUID，所有请求项针对此设备

xmlDstRes(xml string) \* 联合请求命令

xmlObjSets(xml string) 其他可能对象参数（如查询中心存储文件时需要，含<ObjSets>...</ObjSets>）

domainRoad(string) 级联子域名称，根平台默认为空字符串

returnType(string) 响应字符串格式xml | json(default)

returnFlag(uint) 解析信号量（0内部解析，1留给上层自行解析，当returnType='xml'时，视returnFlag===1处理）

}

调用：NPPILY.CommonRequest(connectId, options);

说明：1、以上配置命令、控制命令操作均可以使用此接口替换（也即是上面的接口是在此基础上又进一步封装而成的），单命令操作时建议使用对应的配置或控制接口

2、要特别注意options.cmdType一定不可省略传值

3、options.xmlDstRes结构如<DstRes Type=\"?\" Idx=\"?\" OptID=\"?\" StreamType=\"?\"... ><Param ... /></DstRes>...<DstRes ...>...</DstRes>

4、options.xmlObjSets结构如<ObjSets><Res ObjType=\"?\" ObjID=\"?\" Type=\"?\" Idx=\"?\" ></Res>...</ObjSets>

5、组合的xml报文中，所有命令都应该是符合options.cmdType的同一种操作，查询都是查询，设置都是设置；某个查询项的Idx属性值一定不要超出设备对应资源索引范围，否则整体都会响应出错，比如视频索引范围是0、1、2、3,那么Idx为5就是需要避免的错误操作

示例：1、查询视频亮度对比度等var operator = NPPILY.CommonRequest

(

connectId,

{

cmdType: NPPILY.Enum.CmdType.GET,

puid: "151038400117512284",

xmlDstRes: "<DstRes Type=\"IV\" Idx=\"0\" OptID=\"CFG\_IV\_Brightness\"></DstRes><DstRes Type=\"IV\" Idx=\"0\" OptID=\"CFG\_IV\_Contrast\"></DstRes><DstRes Type=\"IV\" Idx=\"0\" OptID=\"CFG\_IV\_Hue\"></DstRes><DstRes Type=\"IV\" Idx=\"0\" OptID=\"CFG\_IV\_Saturation\"></DstRes>"

}

); operator返回值可能的结构为{"rv":0,"response":[{"Type":"IV","ErrorCode":"0","Setable":"1","OptID":"CFG\_IV\_Brightness","Idx":"0","Param":{"Value":"43"}},{"Type":"IV","ErrorCode":"0","Setable":"1","OptID":"CFG\_IV\_Contrast","Idx":"0","Param":{"Value":"52"}},{"Type":"IV","ErrorCode":"0","Setable":"1","OptID":"CFG\_IV\_Hue","Idx":"0","Param":{"Value":"0"}},{"Type":"IV","ErrorCode":"0","Setable":"1","OptID":"CFG\_IV\_Saturation","Idx":"0","Param":{"Value":"50"}}]}

2、设置视频相关参数var operator = NPPILY.CommonRequest

　　(

　　 connectId,

　　 {

　　 cmdType: NPPILY.Enum.CmdType.SET,

　　 puid: "151000002383163484",

　　 xmlDstRes: "<DstRes Type=\"IV\" Idx=\"0\" OptID=\"CFG\_IV\_Brightness\"><Param Value=\"43\" /></DstRes><DstRes Type=\"IV\" Idx=\"0\" OptID=\"CFG\_IV\_Contrast\"><Param Value=\"52\" /></DstRes><DstRes Type=\"IV\" Idx=\"0\" OptID=\"CFG\_IV\_Hue\"><Param Value=\"0\" /></DstRes><DstRes Type=\"IV\" Idx=\"0\" OptID=\"CFG\_IV\_Saturation\"><Param Value=\"65\" /></DstRes>"

　　 }

　　); operator返回值可能的结构为{"rv":0,"response":[{"Type":"IV","ErrorCode":"0","OptID":"CFG\_IV\_Brightness","Idx":"0"},{"Type":"IV","ErrorCode":"0","OptID":"CFG\_IV\_Contrast","Idx":"0"},{"Type":"IV","ErrorCode":"0","OptID":"CFG\_IV\_Hue","Idx":"0"},{"Type":"IV","ErrorCode":"0","OptID":"CFG\_IV\_Saturation","Idx":"0"}]}

* + 1. 获取设备时区NPPILY.ST\_GetTZ(string connectId, string puid)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

调用：NPPILY.ST\_GetTZ(connectId, puid);

说明：返回值operator（下同）成功时operator.response格式如下：GMT-08:00,这个就是北京时间的设置；

表示当前的时区加/减多少时间就是格林威治时间.比如北京的时间减去8个小时就是格林威治时间.这个跟Windows系统的时区设置中的时间是反的,只要将Windows显示的”GMT+08:00”的加号变成减号就可以了

* + 1. 设置设备时区NPPILY.ST\_SetTZ(string connectId, string puid, timeZone)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

String timeZone [NOT NULL] 表示时区的字符串，如“GMT-08:00”

调用：NPPILY.ST\_SetTZ(connectId, puid, timeZone);

* + 1. 手动重启设备NPPILY.ST\_Reboot(string connectId, string puid)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

调用：NPPILY.ST\_Reboot(connectId, puid);

* + 1. 获取设备当前时间NPPILY.ST\_GetTime(string connectId, string puid)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

调用：NPPILY.ST\_GetTime(connectId, puid);

说明：返回UTC时间秒，示例：operator.response = “1389078573”;

* + 1. 设置设备当前时间NPPILY.ST\_SetTime(string connectId, string puid, string time)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

String time [NOT NULL] UTC时间秒，如1389078573

调用：NPPILY.ST\_SetTime(connectId, puid, time);

* + 1. 获取设备是否在线可用NPPILY.GetResourceUsable(string connectId, string puid)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

调用：NPPILY.GetResourceUsable(connectId, puid);

* + 1. 获取前端联动支持的事件NPPILY.ST\_GetSupportedEventSets(string connectId, string puid)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

调用：NPPILY.ST\_GetSupportedEventSets(connectId, puid);

返回值：成功时，response结构为Array（{"Type": ?, "Event": ?}, ...)

* + 1. 获取前端联动支持的动作NPPILY.ST\_GetSupportedActionSets(string connectId, string puid)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

调用：NPPILY.ST\_GetSupportedActionSets(connectId, puid);

返回值：成功时，response结构为Array（String, String, ...)

String可能取值定义如下：

alertOut 报警输出

sendEmail 发送电子邮件,并附抓图

ftpUpload 抓拍上传到FTP服务器

record 开始前端录像

snapShot 开始前端抓拍

moveToPresetPosition 前往预置位

online 上线

sendSMS 发送报警短信

preTransmitVideo 预传视频

preSnapshot 预抓图

preTransmitAudio 预传音频

preTransmitGPS 预传定位信息

uploadAlert 上传报警信息

uploadRecord 上传录像,实时上传视频和音频

playTipVoice 播放提示声音

* + 1. 获取前端联动配置NPPILY.ST\_GetDeviceLinkActions(string connectId, string puid, string returnType)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

String returnType [NULL] 返回字符串格式，xml或json（缺省值）

调用：NPPILY.ST\_GetDeviceLinkActions(connectId, puid);

返回值：成功时，response结构如下（如果返回xml格式）

<deviceLinkActions>

<deviceLinkAction>

<eventType>EVT\_IDL\_AlertIn</eventType>

<srcResType>IDL</srcResType>

<eventSrcNo>0</eventSrcNo>

<guardMap>0x……</guardMap>

<action>alertOut</action>

<actionParams>

</actionParams>

</deviceLinkAction>

...

</deviceLinkActions>

* + 1. 设置前端联动配置NPPILY.ST\_SetDeviceLinkActions(string connectId, string puid, string xmlDevLinkActions)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

String xmlDevLinkActions[NULL] 联动配置（xml格式字符串，<deviceLinkActions><deviceLinkAction>...</deviceLinkAction>...<deviceLinkAction>...</deviceLinkAction></deviceLinkActions>）

调用：NPPILY.ST\_SetDeviceLinkActions(connectId, puid);

* + 1. 摄像头状态NPPILY.IV\_GetCameraStatus(string connectId, string puid, uint ivIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint ivIndex [NOT NULL] 视频输入（IV）资源索引

调用：NPPILY.IV\_GetCameraStatus(connectId, puid, ivIndex);

返回值：成功时operator.response为0表示无视频，为1表示有视频

* + 1. 获取视频亮度NPPILY.IV\_GetBrightness(string connectId, string puid, uint ivIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint ivIndex [NOT NULL] 视频输入（IV）资源索引

调用：NPPILY.IV\_GetBrightness(connectId, puid, ivIndex);

返回值：成功时operator.response取值范围0-100

* + 1. 设置视频亮度NPPILY.IV\_SetBrightness(string connectId, string puid, uint ivIndex， uint value)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint ivIndex [NOT NULL] 视频输入（IV）资源索引

Uint value [NOT NULL] 取值范围0-100

调用：NPPILY.IV\_SetBrightness(connectId, puid, ivIndex，value);

* + 1. 预览设置视频亮度NPPILY.IV\_PreviewBrightness(string connectId, string puid, uint ivIndex， uint value)

说明：参考NPPILY.IV\_SetBrightness；设置亮度前可以调用此命令进行预览，然后再调用设置

* + 1. 获取视频对比度NPPILY.IV\_GetContrast(string connectId, string puid, uint ivIndex)
    2. 设置视频对比度NPPILY.IV\_SetContrast(string connectId, string puid, uint ivIndex， uint value)
    3. 预览设置视频对比度NPPILY.IV\_PreviewContrast(string connectId, string puid, uint ivIndex， uint value)
    4. 获取视频色调NPPILY.IV\_GetHue(string connectId, string puid, uint ivIndex)
    5. 设置视频色调NPPILY.IV\_SetHue(string connectId, string puid, uint ivIndex， uint value)
    6. 预览设置视频色调NPPILY.IV\_PreviewHue(string connectId, string puid, uint ivIndex， uint value)
    7. 获取视频饱和度NPPILY.IV\_GetSaturation(string connectId, string puid, uint ivIndex)
    8. 设置视频饱和度NPPILY.IV\_SetSaturation(string connectId, string puid, uint ivIndex， uint value)
    9. 预览设置视频饱和度NPPILY.IV\_PreviewSaturation(string connectId, string puid, uint ivIndex， uint value)
    10. 获取支持的流类型NPPILY.IV\_GetSupportedStreamTypeSets(string connectId, string puid, uint ivIndex)

返回值：NPPILY.Enum.NrcapStreamType中定义的值集合

* + 1. 获取支持的编码分辨率NPPILY.IV\_GetSupportedResolutionSets(string connectId, string puid, uint ivIndex)

返回值：以下值的集合：QCIF 176\*144(PAL) 176\*120(NTSC)

QVGA 320\*240

CIF 352\*288(PAL) 352\*240(NTSC)

VGA 640\*480

2CIF 704\*288(此分辨率时,帧信息头填704\*576,客户端拉伸显示)(PAL)

704\*240(此分辨率时,帧信息头填704\*480,客户端拉伸显示)(NTSC)

HD1 720\*288(此分辨率时,帧信息头填720\*576,客户端拉伸显示)(PAL)

720\*240(此分辨率时,帧信息头填720\*480,客户端拉伸显示)(NTSC)

4CIF 704\*576(PAL) 704\*480(NTSC)

D1 720\*576(PAL) 720\*480(NTSC)

960H 960\*576(PAL) 960\*480(NTSC)

720P 1280\*720

960P 1280\*960

1080P 1920\*1080

示例：operator.response可能取值为["CIF", "D1"]

* + 1. 获取视频编码分辨率NPPILY.IV\_GetResolution(string connectId, string puid, uint ivIndex, NPPILY.Enum.NrcapStreamType streamType)

返回值：参考NPPILY.IV\_GetSupportedStreamTypeSets中取值说明

* + 1. 设置视频编码分辨率NPPILY.IV\_SetResolution(string connectId, string puid, uint ivIndex, NPPILY.Enum.NrcapStreamType streamType，string resolution)

返回值：resolution取值参考NPPILY.IV\_GetSupportedStreamTypeSets中取值说明

示例：NPPILY.IV\_SetResolution(connectId, "151087211716864051", 0, NPPILY.Enum.NrcapStreamType.REALTIME, "CIF")

* + 1. 获取支持的工作模式NPPILY.IDL\_GetSupportedWorkModeSets(string connectId, string puid, unit idlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

调用：NPPILY.IDL\_GetSupportedWorkModeSets(connectId, puid, idlIndex);

说明：成功时返回以下值的组合：“AlertIn”,“Counter”,“StatusCap”；示例：operator.response可能取值为[“AlertIn”,“Counter”,“StatusCap”]

* + 1. 获取当前的工作模式NPPILY.IDL\_GetWorkMode(string connectId, string puid, unit idlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

调用：NPPILY.IDL\_GetWorkMode(connectId, puid, idlIndex);

说明：返回值成功时operator.response可能为“AlertIn”,“Counter”,“StatusCap”值之一

* + 1. 设置当前的工作模式NPPILY.IDL\_SetWorkMode(string connectId, string puid, unit idlIndex, string value)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

String value [NOT NULL] 工作模式值，“AlertIn”,“Counter”,“StatusCap”之一

调用：NPPILY.IDL\_SetWorkMode(connectId, puid, idlIndex);

* + 1. 支持的报警触发模式NPPILY.IDL\_GetSupportedAlertInModeSets(string connectId, string puid, unit idlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

调用：NPPILY.IDL\_GetSupportedAlertInModeSets(connectId, puid, idlIndex);

说明：以下四个值的组合，NC（闭合报警），NO（开路报警），OC（先开后闭报警），CO（先闭后开报警）；示例：返回值成功时operator.response可能为[“NC”,“NO”,“OC”,“CO”]

* + 1. 获取当前的报警触发模式NPPILY.IDL\_GetAlertInMode(string connectId, string puid, unit idlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

调用：NPPILY.IDL\_GetAlertInMode(connectId, puid, idlIndex);

说明：返回值成功时operator.response可能为“NC”,“NO”,“OC”,“CO”值之一

* + 1. 设置当前的报警触发模式NPPILY.IDL\_SetAlertInMode(string connectId, string puid, unit idlIndex, value)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

String value [NOT NULL] 取值范围“NC”,“NO”,“OC”,“CO”之一

调用：NPPILY.IDL\_SetAlertInMode(connectId, puid, idlIndex, value);

* + 1. 报警输入状态NPPILY.IDL\_GetAlertInStatus(string connectId, string puid, unit idlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

调用：NPPILY.IDL\_GetAlertInStatus(connectId, puid, idlIndex);

说明：返回值成功时operator.response可能为0（无报警）或1（有报警）

* + 1. 获取开始计数时间NPPILY.IDL\_GetCountBeginTime(string connectId, string puid, unit idlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

调用：NPPILY.IDL\_GetCountBeginTime(connectId, puid, idlIndex);

说明：返回值成功时operator.response可能为UTC时间秒数，如“1385129300”

* + 1. 计数总值NPPILY.IDL\_GetCount(string connectId, string puid, unit idlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

调用：NPPILY.IDL\_GetCount(connectId, puid, idlIndex);

* + 1. 状态是否有效NPPILY.IDL\_GetLevelValid(string connectId, string puid, unit idlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

调用：NPPILY.IDL\_GetLevelValid(connectId, puid, idlIndex);

* + 1. 读取报警布防时间表NPPILY.IDL\_GetGuardMap(string connectId, string puid, unit idlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

调用：NPPILY.IDL\_GetGuardMap(connectId, puid, idlIndex);

说明：返回值成功时operator.response可能为0x打头168个十六进制字符串，示例：“0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF”

* + 1. 设置报警布防时间表NPPILY.IDL\_SetGuardMap(string connectId, string puid, unit idlIndex, string mapValue)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint idlIndex [NOT NULL] 报警输入资源索引

String mapValue [NOT NULL] 布防时间表，168个十六进制字符组合（不须Ox打头）

调用：NPPILY.IDL\_SetGuardMap(connectId, puid, idlIndex, mapValue);

说明：特殊的，报警开（全布防）关（全撤销布防）：当mapValue为“FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF”表示全布防；为“0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000”表示撤销布防；

* + 1. 获取报警输出默认状态NPPILY.ODL\_GetDefaultConnectStatus(string connectId, string puid, unit odlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint odlIndex [NOT NULL] 报警输出资源索引

调用：NPPILY.ODL\_GetDefaultConnectStatus(connectId, puid, odlIndex);

返回值：operator.response -> 1表示接通，0表示断开

说明：表示设备加电之后，不对报警输出操作时候的默认状态

* + 1. 获取报警输出当前状态NPPILY.ODL\_GetConnectStatus(string connectId, string puid, unit odlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint odlIndex [NOT NULL] 报警输出资源索引

调用：NPPILY.ODL\_GetConnectStatus(connectId, puid, odlIndex);

返回值：operator.response -> 1表示接通，0表示断开，2表示闪烁

说明：闪烁即间隔通断

* + 1. 获取接通动作别名NPPILY.ODL\_GetAliasConnect(string connectId, string puid, unit odlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint odlIndex [NOT NULL] 报警输出资源索引

调用：NPPILY.ODL\_GetAliasConnect(connectId, puid, odlIndex);

* + 1. 设置接通动作别名NPPILY.ODL\_SetAliasConnect(string connectId, string puid, unit odlIndex, string aliasName)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint odlIndex [NOT NULL] 报警输出资源索引

String aliasName [NOT NULL] 别名字符串

调用：NPPILY.ODL\_SetAliasConnect(connectId, puid, odlIndex, aliasName);

* + 1. 获取断开动作别名NPPILY.ODL\_GetAliasBreak(string connectId, string puid, unit odlIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint odlIndex [NOT NULL] 报警输出资源索引

调用：NPPILY.ODL\_GetAliasBreak(connectId, puid, odlIndex);

* + 1. 设置断开动作别名NPPILY.ODL\_SetAliasBreak(string connectId, string puid, unit odlIndex, string aliasName)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint odlIndex [NOT NULL] 报警输出资源索引

String aliasName [NOT NULL] 别名字符串

调用：NPPILY.ODL\_SetAliasBreak(connectId, puid, odlIndex, aliasName);

* + 1. 控制报警输出状态NPPILY.ODL\_SetStatus(string connectId, string puid, unit odlIndex, uint value)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint odlIndex [NOT NULL] 报警输出资源索引

uint value [NOT NULL] 取值范围 1接通、0断开、2闪烁

调用：NPPILY.ODL\_SetStatus(connectId, puid, odlIndex，value);

* + 1. 获取前端录像持续时间NPPILY.SG\_GetRecordTimeSpan(string connectId, string puid, unit sgIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

调用：NPPILY.SG\_GetRecordTimeSpan(connectId, puid, sgIndex);

* + 1. 设置前端录像持续时间NPPILY.SG\_SetRecordTimeSpan(string connectId, string puid, unit sgIndex, uint value)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

uint value [NOT NULL] 录像文件的时间长度，单位为秒，满这个时间之后就换文件

调用：NPPILY.SG\_SetRecordTimeSpan(connectId, puid, sgIndex，value);

* + 1. 获取前端磁盘状态NPPILY.SG\_GetDiskStatus(string connectId, string puid, unit sgIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

调用：NPPILY.SG\_GetDiskStatus(connectId, puid, sgIndex);

返回值：operator.response ->

OK 正常

Space Lack 空间不足

Space Full 已满

Invalid Format 格式未识别

Not Mount 未加载

No Disk 无磁盘

RTC Error 时间错误

说明：如果有多块磁盘，那么只要有磁盘正常就返回OK，所有磁盘都没有接的时候，才返回无磁盘

* + 1. 获取前端磁盘空间不足时是否覆盖旧文件NPPILY.SG\_GetCoverOldRecordFile(string connectId, string puid, unit sgIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

调用：NPPILY.SG\_GetCoverOldRecordFile(connectId, puid, sgIndex);

* + 1. 设置前端磁盘空间不足时是否覆盖旧文件NPPILY.SG\_SetCoverOldRecordFile(string connectId, string puid, unit sgIndex, uint value)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

uint value [NOT NULL] 1覆盖，0不覆盖

调用：NPPILY.SG\_SetCoverOldRecordFile(connectId, puid, sgIndex，value);

* + 1. 获取前端磁盘空间不足时的剩余空间门限NPPILY.SG\_GetDiskInsufficientSpace(string connectId, string puid, unit sgIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

调用：NPPILY.SG\_GetDiskInsufficientSpace(connectId, puid, sgIndex);

* + 1. 设置前端磁盘空间不足时的剩余空间门限NPPILY.SG\_SetDiskInsufficientSpace(string connectId, string puid, unit sgIndex, uint value)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

uint value [NOT NULL] 单位MByte 大于0

调用：NPPILY.SG\_SetDiskInsufficientSpace(connectId, puid, sgIndex，value);

* + 1. 获取前端录像文件保留天数NPPILY.SG\_GetRecordFileReserveDays(string connectId, string puid, unit sgIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

调用：NPPILY.SG\_GetRecordFileReserveDays(connectId, puid, sgIndex);

* + 1. 设置前端录像文件保留天数NPPILY.SG\_SetRecordFileReserveDays(string connectId, string puid, unit sgIndex, uint value)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

uint value [NOT NULL] 单位天，必须大于0

调用：NPPILY.SG\_SetRecordFileReserveDays(connectId, puid, sgIndex，value);

* + 1. 获取前端存储文件系统类型NPPILY.SG\_GetFileSystemType(string connectId, string puid, unit sgIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

调用：NPPILY.SG\_GetFileSystemType(connectId, puid, sgIndex);

说明：返回值operator.response = CEFS、FAT32等，调用相应的接口进行查询、点播、下载存储文件。

* + 1. 获取前端磁盘信息NPPILY.SG\_GetDiskInfo(string connectId, string puid, unit sgIndex)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

调用：NPPILY.SG\_GetDiskInfo(connectId, puid, sgIndex);

返回值：operator.response 可能的响应如下（以下含两个盘符信息）

[{"status":"OK","letter":"A","volume\_label":"","type":"","file\_system":"CEFS","capacity":"3782","used\_space":"3713","usable\_space":"68"},{"status":"No Disk","letter":"B","volume\_label":"","type":"","file\_system":"Unknown","capacity":"0","used\_space":"0","usable\_space":"0"}]

含义如下：

status 磁盘状态，取值如下

OK 正常

Unknown Format 格式未识别

No Disk 无磁盘

Not Mount 未加载

RTC Error 时间错误

无此标签缺省为OK，只有OK的时候，下面的项才有意义

letter 盘符，字符串型，磁盘的标识，从A开始，比如"A","B",…

volume\_label 卷标,字符串型,标准文件系统的,表示磁盘挂载的路径,比如:/mnt/usb.CEFS的做标识用.

type 磁盘类型,字符串型,取值和含义如下:

TF TF卡/MicroSD卡

SD SD卡

USB U盘/USB硬盘

Hard 硬盘

file\_system 文件系统,字符串型

capacity 容量,32位整型字符串,单位MB,最大支持4KT

used\_space 已用空间,32位整型字符串,单位MB,最大支持4KT

usable\_space 可用空间,32位整型字符串,单位MB,最大支持4KT

* + 1. 开始前端初始化文件系统NPPILY.SG\_StartInitFileSystem(string connectId, string puid, unit sgIndex, string sgDiskLetter)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

String sgDiskLetter [NOT NULL] 磁盘盘符，如A，C

调用：NPPILY.SG\_StartInitFileSystem(connectId, puid, sgIndex， sgDiskLetter);

* + 1. 查询前端初始化文件系统进度NPPILY.SG\_QueryInitFileSystemProgress(string connectId, string puid, unit sgIndex, string sgDiskLetter)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

调用：NPPILY.SG\_QueryInitFileSystemProgress(connectId, puid, sgIndex);

返回值：operator.response -> 0~100,表示格式化磁盘的进度百分比,如果查到100表示格式化结束

* + 1. 手动启动前端存储NPPILY.SG\_ManualStart(string connectId, string puid, unit sgIndex, NPPILY.Enum.PuResourceType ivType, uint ivIndex, ivStreamType, uint duration)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

NPPILY.Enum.PuResourceType ivType [NOT NULL] 一般为IV

Uint ivIndex [NOT NULL] 视频资源索引

NPPILY.Enum.NrcapStreamType ivStreamType [NOT NULL] 流类型

Uint duration [NOT NULL] 前端手动存储持续时间，单位分钟.最小1分钟,最长1440分钟(24小时)

调用：NPPILY.SG\_ManualStart(connectId, puid, sgIndex, ivType, ivIndex, ivStreamType, duration);

* + 1. 手动停止前端存储NPPILY.SG\_ManualStop(string connectId, string puid, unit sgIndex, NPPILY.Enum.PuResourceType ivType, uint ivIndex, ivStreamType)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

NPPILY.Enum.PuResourceType ivType [NOT NULL] 一般为IV

Uint ivIndex [NOT NULL] 视频资源索引

NPPILY.Enum.NrcapStreamType ivStreamType [NOT NULL] 流类型

调用：NPPILY.SG\_ManualStop(connectId, puid, sgIndex, ivType, ivIndex, ivStreamType);

* + 1. 查询CEFS格式前端存储录像或图片NPPILY.SG\_CEFSQueryFiles(string connectId, string puid, unit sgIndex, unit ivIndex, UTC beginTime, UTC endTime, NPPILY.Enum.NrcapStreamType ivStreamType)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

Uint ivIndex[NOT NULL] 要查询的视频资源索引

UTC beginTime[NOT NULL] 开始时间，UTC时间秒，如1398758400

UTC endTime[NOT NULL] 结束时间，UTC时间秒，如1398844800

NPPILY.Enum.NrcapStreamType ivStreamType [NOT NULL] 流类型，传入“PICTURE”查前端图片，传入“STORAGE”查前端录像

调用：NPPILY.SG\_CEFSQueryFiles(connectId, puid, sgIndex, ivIndex, beginTime, endTime, streamType);下面为调用示例

查录像 -> var operator = NPPILY.SG\_CEFSQueryFiles(connectId, "151038403942712700", 0, 0, 1398744800, 1399934800, "STORAGE");

查图片 -> var operator = NPPILY.SG\_CEFSQueryFiles(connectId, "151038403942712700", 0, 0, 1398744800, 1399934800, "PICTURE");

其中connectId为实际建立的连接ID

返回值：成功时，如果operator.response为空说明查询结果为空，不为空时应该是数组，结构如下

查录像 -> [{"name": "?", "path": "?", "reasons": "?", "size": "?", "begin\_time": "?", "end\_time": "?"}, ...]；示例[{"name": "140429165510.avi", "path": "", "reasons": "Schedule", "size": "3200000", "begin\_time": "1398761710", "end\_time": "1398761735"}, {"name": "140429165535.avi", "path": "", "reasons": "Manual", "size": "4480000", "begin\_time": "1398761735", "end\_time": "1398761770"}]注意属性size单位字节，reasons值定义参考NPPILY.Enum.CEFSRecordReason

查图片 -> [{"name": "?", "path": "?", "reasons": "?", "size": "?", "time": "?", "no": "?"}, ...]； 示例[{"name": "140429170502.jpg", "path": "", "reasons": "Schedule", "size": "32", "time": "1398762302", "no": "0"}, {"name": "140429170502.jpg", "path": "", "reasons": "Schedule", "size": "32", "time": "1398762302", "no": "1"}, {"name": "140429170503.jpg", "path": "", "reasons": "Schedule", "size": "32", "time": "1398762303", "no": "0"}]注意属性time抓拍时间，no为同一秒内抓拍的图片编号，reasons值定义参考NPPILY.Enum.CEFSPictureReason

* + 1. 获取平台录像持续时间NPPILY.SC\_GetRecordTimeSpan(string connectId, string csuPuid, unit csuIndex)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

调用：NPPILY.SC\_GetRecordTimeSpan(connectId, csuPuid, csuIndex);

* + 1. 设置平台录像持续时间NPPILY.SC\_SetRecordTimeSpan(string connectId, string csuPuid, unit csuIndex, uint value)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

uint value [NOT NULL] 录像文件的时间长度，单位为秒，满这个时间之后就换文件 范围60-1800

调用：NPPILY.SC\_SetRecordTimeSpan(connectId, csuPuid, csuIndex，value);

* + 1. 获取平台磁盘状态NPPILY.SC\_GetDiskStatus(string connectId, string csuPuid, unit csuIndex)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

调用：NPPILY.SC\_GetDiskStatus(connectId, csuPuid, csuIndex);

返回值：operator.response ->

OK 磁盘正常

Invalid Format 磁盘未格式化

No Disk 无磁盘

* + 1. 获取平台存储磁盘满时是否覆盖旧文件NPPILY.SC\_GetCoverOldRecordFile(string connectId, string csuPuid, unit csuIndex)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

调用：NPPILY.SC\_GetCoverOldRecordFile(connectId, csuPuid, csuIndex);

* + 1. 设置平台存储磁盘满时是否覆盖旧文件NPPILY.SC\_SetCoverOldRecordFile(string connectId, string csuPuid, unit csuIndex， uint value)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

uint value [NOT NULL] 1覆盖，0不覆盖

调用：NPPILY.SC\_SetCoverOldRecordFile(connectId, csuPuid, csuIndex，value);

* + 1. 获取平台存储录像文件保存天数NPPILY.SC\_GetRecordFileReserveDays(string connectId, string csuPuid, unit csuIndex)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

调用：NPPILY.SC\_GetRecordFileReserveDays(connectId, csuPuid, csuIndex);

* + 1. 设置平台存储录像文件保存天数NPPILY.SC\_SetRecordFileReserveDays(string connectId, string csuPuid, unit csuIndex， uint value)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

uint value [NOT NULL] 取值范围1-365

调用：NPPILY.SC\_SetRecordFileReserveDays(connectId, csuPuid, csuIndex，value);

* + 1. 获取平台存储抓拍文件保存天数NPPILY.SC\_GetSnapshotReserveDays(string connectId, string csuPuid, unit csuIndex)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

调用：NPPILY.SC\_GetSnapshotReserveDays(connectId, csuPuid, csuIndex);

* + 1. 设置平台存储抓拍文件保存天数NPPILY.SC\_SetSnapshotReserveDays(string connectId, string csuPuid, unit csuIndex， uint value)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

uint value [NOT NULL] 取值范围1-365

调用：NPPILY.SC\_SetSnapshotReserveDays(connectId, csuPuid, csuIndex，value);

* + 1. 获取平台存储GPS数据保存天数NPPILY.SC\_GetGPSReserveDays(string connectId, string csuPuid, unit csuIndex)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

调用：NPPILY.SC\_GetGPSReserveDays(connectId, csuPuid, csuIndex);

* + 1. 设置平台存储GPS数据保存天数NPPILY.SC\_SetGPSReserveDays(string connectId, string csuPuid, unit csuIndex， uint value)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

uint value [NOT NULL] 取值范围1-365

调用：NPPILY.SC\_SetGPSReserveDays(connectId, csuPuid, csuIndex，value);

* + 1. 获取平台存储是否使能GPS存储NPPILY.SC\_GetEnableGPSDataStorage(string connectId, string csuPuid, unit csuIndex)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

调用：NPPILY.SC\_GetEnableGPSDataStorage(connectId, csuPuid, csuIndex);

* + 1. 设置平台存储是否使能GPS存储NPPILY.SC\_SetEnableGPSDataStorage(string connectId, string csuPuid, unit csuIndex， uint value)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

uint value [NOT NULL] 1使能，0非

调用：NPPILY.SC\_SetEnableGPSDataStorage(connectId, csuPuid, csuIndex，value);

* + 1. 获取平台存储磁盘信息NPPILY.SC\_GetDiskInfo(string connectId, string csuPuid, unit csuIndex)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

调用：NPPILY.SC\_GetDiskInfo(connectId, csuPuid, csuIndex);

返回值：operator.response ->

[{"status":"OK","letter":"C","volume\_label":"","type":"DRIVE\_FIXED","file\_system":"NTFS","capacity":"29996","used\_space":"21160","usable\_space":"8835"},{"status":"OK","letter":"D","volume\_label":"","type":"DRIVE\_FIXED","file\_system":"NTFS","capacity":"223003","used\_space":"62699","usable\_space":"160303"},{"status":"OK","letter":"E","volume\_label":"VedioRecord\_200G","type":"DRIVE\_FIXED","file\_system":"NTFS","capacity":"223929","used\_space":"162802","usable\_space":"61126"},{"status":"OK","letter":"F","volume\_label":"VedioRecord\_1800GB","type":"DRIVE\_FIXED","file\_system":"NTFS","capacity":"1907725","used\_space":"22231","usable\_space":"1885494"}]

含义如下：

status状态

letter:盘符

volume\_label:字符串型,卷标

type: 字符串型,磁盘类型

file\_system: 字符串型,文件系统

capacity:64bit整型,容量 单位MB

used\_space:64bit整型,已用空间

usable\_space:64bit整型,可用空间

* + 1. 手动启动平台存储NPPILY.SC\_ManualStart(string connectId, string csuPuid, unit csuIndex, string rec\_puid, uint ivIndex, ivStreamType, string rec\_reason, uint rec\_duration)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

String rec\_puid [NOT NULL] 视频设备PUID

Uint rec\_ivIndex [NOT NULL] 视频资源索引

NPPILY.Enum.NrcapStreamType ivStreamType [NOT NULL] 流类型

String rec\_reason [NOT NULL] 录像原因，可能的取值：Manual，各种事件等

Uint rec\_duration [NOT NULL] 平台手动存储持续时间，单位分钟.最小1分钟,最长1440分钟(24小时)

调用：NPPILY.SC\_ManualStart(connectId, puid, csuIndex, rec\_puid, rec\_ivIndex, rec\_streamType, rec\_reason, rec\_duration);

* + 1. 手动停止平台存储NPPILY.SC\_ManualStop(string connectId, string csuPuid, unit csuIndex, string rec\_puid, uint ivIndex, ivStreamType)

参数：string connectId [NOT NULL] 连接ID

String csuPuid [NOT NULL] 中心存储器设备PUID

Uint csuIndex [NOT NULL] 中心存储单元资源索引，一般为0

String rec\_puid [NOT NULL] 视频设备PUID

Uint rec\_ivIndex [NOT NULL] 视频资源索引

NPPILY.Enum.NrcapStreamType ivStreamType [NOT NULL] 流类型

调用：NPPILY.SC\_ManualStop(connectId, puid, csuIndex, rec\_puid, rec\_ivIndex, rec\_streamType);

* + 1. 外接磁盘加载模块NPPILY.RecordQR\_Startup()

说明：对外接磁盘进行操作时，无论有没有插入了外接磁盘，均可以调用此接口，必须保证首先加载（初始化）模块成功，否则对外接磁盘的其他操作均视为无效的。

* + 1. 外接磁盘清除模块NPPILY.RecordQR\_Cleanup()

说明：在完成或终止外接磁盘的相关操作时，可对其清除模块，在NPPILY.UnLoad()卸载插件时亦会自行清除的，故上层可不用必须调用。注意，清除模块与加载模块是互斥、相反的操作。

* + 1. 外接磁盘获取设备信息NPPILY.RecordQR\_GetDeviceInfo()

返回值：Object(NPPILY.Struct.ReturnValueStruct)

示例，如{"rv":0,"response":{"mac":"000F7C0B08A3","puid":"","tz":"GMT-08:00","name":"000F7C0B08A3","video\_name":{"0":"Camera0", "1":"Camera1" ... }}}，其中mac为设备的MAC，puid为设备的PUID（此时可以为空），tz为设备所属的时区，name为设备名称，video\_name为设备的视频资源，此节点下可以有多个子节点，以字符串型数字为属性名，如"0"表示第1路视频索引，如下面相关函数的videoIndex参数

* + 1. 外接磁盘查询录像日期NPPILY.RecordQR\_QueryRecordDate(uint videoIndex, uint offset, uint count)

参数：uint videoIndex [NOT NULL] 视频索引，从设备信息中获得

Uint offset [NULL] 分页查询偏移量，缺省为0

Uint count [NULL] 分页查询条数，缺省为1000

调用：NPPILY.RecordQR\_QueryRecordDate(videoIndex, offset, count);

返回值：Object(NPPILY.Struct.ReturnValueStruct)

示例，如{"rv":0,"response":["1397145600"]}，其中response属性可能为空，说明没有录像日期，也就没有录像文件了；如果有录像日期，那么response为天的零时日期（UTC秒表示）数组

* + 1. 外接磁盘查询录像文件NPPILY.RecordQR\_QueryRecordFiles(uint videoIndex, uint offset, uint count, utctime bUTCTime, utctime eUTCTime)

参数：uint videoIndex [NOT NULL] 视频索引，从设备信息中获得

Uint offset [NULL] 分页查询偏移量，缺省为0

Uint count [NULL] 分页查询条数，缺省为1000

utctime bUTCTime[ NULL] 查询开始时间，缺省为0

utctime eUTCTime[ NULL] 查询结束时间，缺省为0

调用：NPPILY.RecordQR\_QueryRecordFiles(videoIndex, offset, count, bUTCTime, eUTCTime);

返回值：Object(NPPILY.Struct.ReturnValueStruct)

示例，调用var operator = NPPILY.RecordQR\_QueryRecordFiles(0, 0, 2, 1397145600, 1397232000);operator可能为{"rv":0,"response":[{"name":"20140411145500\_0384.avi","path":"B:\\0000\\","size":"107343872","begin\_time":"1397199300","end\_time":"1397200200","reason":["Schedule"]},{"name":"20140411151000\_0384.avi","path":"B:\\0000\\","size":"105155584","begin\_time":"1397200200","end\_time":"1397201100","reason":["Schedule"]}]}，其中name为录像名称，path为外接磁盘中存储地址，size为大小（单位字节），begin\_time为录像文件开始时间，end\_time为录像文件结束时间,reason为录像原因（Schedule：前端计划录像，Manual：手动录像，AlertIn：发生报警，MotionDetected：侦测到移动，SignalLost：视频信号丢失）

说明：如果bUTCTime, eUTCTime之间跨度较大时，建议分页获取，把offset，count置为较小的一个范围，理论上返回条数不大于count条，如果等于count时，应offset=offset+count，count进行后续查询（这里的分页更应该视为分步获取）；下载录像的时候，必须是全地址，第一个参数为远程存储地址，即为上面示例中path+name组合（必须是原封不动的组合起来的），如上“B:\\0000\\20140411145500\_0384.avi”（那么如果组成了B:\0000\20140411145500\_0384.avi或B:/0000/20140411145500\_0384.avi会导致下载失败）

* + 1. 外接磁盘下载录像文件NPPILY.RecordQR\_CopyRecord(string remoteFilePath, string localFilePath)

参数：string remoteFilePath [NOT NULL] 录像在外接磁盘中存储地址

string localFilePath [NOT NULL] 下载录像本地的保存地址

调用：NPPILY.RecordQR\_CopyRecord(remoteFilePath, localFilePath);

示例var operator = NPPILY.RecordQR\_CopyRecord("B:\\0000\\20140411145500\_0384.avi", "C://video\_0\_20140411145500\_0384.avi");

说明：知晓下载状态信息，应“注册外接磁盘存储文件下载信息通知回调”，比如NPPILY.NCNotifyManager.Add(NPPILY.Enum.NCObjectNotify.copy\_file\_notify, function (notify) {

// notify的结构形如{"eventName":"copy\_file\_notify","\_HANDLE":"","errorCode":"0","keyData":{"localFilePath":"C://video\_0\_20140411145500\_0384.avi","currentBytes":102790832,"currentTime":17,"status":0,"statusDesc":"正在下载"}}，其中\_HANDLE、errorCode无意义，keyData节点为具体下载状态信息，localFilePath为本地保存地址（根据此属性区别不同的下载信息），currentBytes为已经下载的字节数，currentTime为下载已经进行的秒数，status为下载状态量（-1：下载异常出错，0：正在下载，1：下载完成），statusDesc为下载状态描述

});

* + 1. 外接磁盘查询抓图日期NPPILY.RecordQR\_QuerySnapshotDate(uint videoIndex, uint offset, uint count)

参数：uint videoIndex [NOT NULL] 视频索引，从设备信息中获得

Uint offset [NULL] 分页查询偏移量，缺省为0

Uint count [NULL] 分页查询条数，缺省为1000

调用：NPPILY.RecordQR\_QuerySnapshotDate(videoIndex, offset, count);

返回值：Object(NPPILY.Struct.ReturnValueStruct)

示例，如{"rv":0,"response":["1397145600"]}，其中response属性可能为空，说明没有抓图日期，也就没有录像文件了；如果有抓图日期，那么response为天的零时日期（UTC秒表示）数组

* + 1. 外接磁盘查询抓图文件NPPILY.RecordQR\_QuerySnapshotFiles(uint videoIndex, uint offset, uint count, utctime bUTCTime, utctime eUTCTime)

参数：uint videoIndex [NOT NULL] 视频索引，从设备信息中获得

Uint offset [NULL] 分页查询偏移量，缺省为0

Uint count [NULL] 分页查询条数，缺省为1000

utctime bUTCTime[ NULL] 查询开始时间，缺省为0

utctime eUTCTime[ NULL] 查询结束时间，缺省为0

调用：NPPILY.RecordQR\_QuerySnapshotFiles(videoIndex, offset, count, bUTCTime, eUTCTime);

返回值：Object(NPPILY.Struct.ReturnValueStruct)

示例，调用var operator = NPPILY.RecordQR\_QuerySnapshotFiles(0, 0, 2, 1399365600, 1399565606);operator可能为{"rv":0,"response":[{"name":"20140506164000\_0000.jpg","path":"B:\\0000\\","size":"56573","begin\_time":"1399365600","end\_time":"1399365600","reason":["Schedule"]},{"name":"20140506164006\_0000.jpg","path":"B:\\0000\\","size":"56573","begin\_time":"1399365606","end\_time":"1399365606","reason":["Schedule"]}]}，其中name为抓图名称，path为外接磁盘中存储地址，size为大小（单位字节），begin\_time为抓图开始时间，end\_time为抓图结束时间（抓图时begin\_time和end\_time应该是一样的），reason为抓图原因（Schedule：前端计划抓图，Manual：手动抓图，AlertIn：发生报警，MotionDetected：侦测到移动，SignalLost：视频信号丢失）

说明：如果bUTCTime, eUTCTime之间跨度较大时，建议分页获取，把offset，count置为较小的一个范围，理论上返回条数不大于count条，如果等于count时，应offset=offset+count，count进行后续查询（这里的分页更应该视为分步获取）；下载抓图的时候，必须是全地址，第一个参数为远程存储地址，即为上面示例中path+name组合（必须是原封不动的组合起来的），如上“B:\\0000\\20140506164000\_0000.jpg”（那么如果组成了B:\0000\20140506164000\_0000.jpg或B:/0000/20140506164000\_0000.jpg会导致下载失败）

* + 1. 外接磁盘下载抓图文件NPPILY.RecordQR\_CopySnapshot(string remoteFilePath, string localFilePath)

说明：参考“外接磁盘下载录像文件NPPILY.RecordQR\_CopyRecord”

* + 1. 外接磁盘停止下载文件NPPILY.RecordQR\_CancelCopy(string localFilePath)

参数：localFilePath [NOT NULL] 下载文件本地保存地址

说明：录像抓图文件停止下载共用此函数

* + 1. 创建会议室NPPILY. CreateMeetingRoom (string connectId, string szName, string szPassword)

参数：connectId[NOT NULL] 连接ID

szName[NOT NULL]要创建的会议室名称

szPassword[NULL]要创建的会议室密码，默认为空字符串

返回值：operator.response表示会议室流句柄（唯一标识），很重要，要记录下来，关闭会议室时传入此句柄

说明：创建、退出、加入、剔除等操作，均会通过事件NPPILY.Enum.NCObjectNotify.event\_notify通知，里面有指示用户的连接ID（CUID），可参考文件“”

* + 1. 申请加入会议室NPPILY. JoinMeetingRoom (string connectId, string szID, string szPassword)

参数：connectId[NOT NULL] 连接ID

szID[NOT NULL]要加入的会议室ID

szPassword[NULL]要加入的会议室密码，默认为空字符串

返回值：operator.response表示会议室流句柄（唯一标识），很重要，要记录下来，退出会议室时传入此句柄

说明：需要侦测NPPILY.Enum.NCObjectNotify.stream\_status\_notify通知，当状态status属性值为-1时，表示断开了流，那么就需要调用退出会议室接口

* + 1. 关闭或退出会议室NPPILY. CloseOrExitMeetingRoom (string connectId, string streamHandle)

参数：connectId[NOT NULL] 连接ID

streamHandle [NULL] 创建或加入会议室时返回的流句柄（唯一标识）

* + 1. 查询所有会议室信息NPPILY. QueryAllMeetingRoomInfo (string connectId, uint count)

参数：connectId[NOT NULL] 连接ID

count [NOT NULL] 查询的个数，默认为1024

返回值：operator结构形如{"rv":0,"response":[{"MeetingRoomName":"rm","MeetingRoomID":"9:009000100000000001:NTQ5YmY0YzcwMDAwMWE5YQ==","MeetingRoomCompere":"64A00308@005000100000000001"},{"MeetingRoomName":"together talking","MeetingRoomID":"9:009000100000000001:NTQ5ZDE0YTEwMDAwMWFkNw==","MeetingRoomCompere":"0DDD030C@005000100000000001"}]}

其中， MeetingRoomName为会议室名称，MeetingRoomID为会议室ID，MeetingRoomCompere为会议室所属的用户CUID

* + 1. 添加一个设备入会议室NPPILY. AddPUMeetingRoom (string connectId, string streamHandle, string puid)

参数：connectId[NOT NULL] 连接ID

streamHandle[NOT NULL] 会议室流句柄，同上定义

puid [NOT NULL] 设备PUID

* + 1. 从会议室中删除一个设备NPPILY. KickOutMemberPU (string connectId, string streamHandle, string puid)

参数：connectId[NOT NULL] 连接ID

streamHandle[NOT NULL] 会议室流句柄，同上定义

puid [NOT NULL] 设备PUID

* + 1. 从会议室中删除一个用户NPPILY. KickOutMemberCU (string connectId, string streamHandle, string szCUID)

参数：connectId[NOT NULL] 连接ID

streamHandle[NOT NULL] 会议室流句柄，同上定义

szCUID [NOT NULL] 用户连接CUID

* + 1. 修改会议室密码NPPILY. KickOutMemberPU (string connectId, string szID, string puid)

参数：connectId[NOT NULL] 连接ID

szID [NOT NULL] 会议室ID

new\_szPassword [NOT NULL] 会议室新的密码

* + 1. 获取平台用户连接ID（CUID）NPPILY. GetCUSessionID (string connectId)

参数：connectId[NOT NULL] 连接ID

返回值：成功时，operator.response结构形如0DDD030C@005000100000000001

* + 1. 获取接收发送总流量NPPILY. GetCurStreamsTotalBytes()

参数：connectId[NOT NULL] 连接ID

返回值：成功时，operator.response结构形如

{

receive: ?, // 已接收的总流量

send: ? // 已发送的总流量

}

* + 1. 实时报警事件处警NPPILY.EventLog.ProcessRuntimeEvent(string connectId, object options)

参数：connectId[NOT NULL] 连接ID

options[NOT NULL]

说明：options结构如下

{

nuType\_las(string) // Log网元类型，缺省13

nuid\_las(string) // Log网元ID，缺省013000100000000001

resourceSets(\*Object|Array) // 资源对象或数组，单个结构如下

{

objType(string) // 资源对象类型，（缺省）151表示设备，153表示用户

objID(\*string) // 资源对象取值，objType为151时，赋值设备PUID，objType为153时，赋值用户CUID

eventID(\*string) // 具体事件ID

time(\*string) // 系统发生的UTC时间

memo(string) // 处警备忘标记的信息，不要写太长，可省略

}

}

备注：系统产生了事件，WEB客户端接收到事件通知，然后在对具体的事件通知进行处警。接收事件示例参考demo/alertIn/index.html页面！

报警事件处警调用示例（实际用到的值请根据真实事件信息处理）：

NPPILY.EventLog.ProcessRuntimeEvent(connectId?, {

resourceSets: [{

objType: '153',

objID: '7C9904B4@005000100000000001',

eventID: 'EVT\_CU\_Online',

time: '1503971480'

},{

objType: '151',

objID: '151038401812031868',

eventID: 'EVT\_PU\_Online',

time: '1503971321',

memo: '处警设备上线测试'

},{

objType: '151',

objID: '151038400060819580',

eventID: 'EVT\_IDL\_AlertIn',

time: '1503970299',

memo: '处警设备发生报警测试'

}]

});

* + 1. 查询FAT32格式前端存储录像或图片NPPILY.SG\_FAT32QueryFiles(string connectId, string puid, unit sgIndex, unit ivIndex, UTC beginTime, UTC endTime, NPPILY.Enum.NrcapStreamType ivStreamType, unit offset, unit count)

参数：string connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器资源索引

Uint ivIndex[NOT NULL] 要查询的视频资源索引

UTC beginTime[NOT NULL] 开始时间，UTC时间秒，如1398758400

UTC endTime[NOT NULL] 结束时间，UTC时间秒，如1398844800

NPPILY.Enum.NrcapStreamType ivStreamType [NOT NULL] 流类型，传入“PICTURE”查前端图片，传入“STORAGE”查前端录像

Uint offset[NOT NULL] 分页查询偏移量

Uint count[NOT NULL] 分页查询条数

调用：NPPILY.SG\_FAT32QueryFiles(connectId, puid, sgIndex, ivIndex, beginTime, endTime, streamType, offset, count);下面为调用示例

查录像 -> var operator = NPPILY.SG\_FAT32QueryFiles (connectId, "152AED27B946CF2AA9", 0, 0, 1574222400, 1574222400, "STORAGE", 0, 10);

查图片 -> var operator = NPPILY.SG\_FAT32QueryFiles (connectId, "152AED27B946CF2AA9", 0, 0, 1574222400, 1574222400, "PICTURE", 0, 10);

其中connectId为实际建立的连接ID

返回值：成功时，如果operator.response为空说明查询结果为空，不为空时应该是数组，结构如下

查录像 ->[{"name":"20191121\_092554.mp4","path":"/storage/1F57-15E5/TerminalHelmet/Storage/Record/","reasons":"本地录像","size":78238790,"begin\_time":1574299554,"end\_time":1574300148},{"name":"20191121\_093548.mp4","path":"/storage/1F57-15E5/TerminalHelmet/Storage/Record/","reasons":"本地录像","size":78232732,"begin\_time":1574300148,"end\_time":1574300743},{"name":"20191121\_094543.mp4","path":"/storage/1F57-15E5/TerminalHelmet/Storage/Record/","reasons":"本地录像","size":78250729,"begin\_time":1574300743,"end\_time":1574301339}] 注意属性size单位字节

查图片 ->[{"name":"20191121\_104338.jpg","path":"/storage/emulated/0/TerminalHelmet//Storage/Snapshot/","reasons":"本地录像","size":455982,"time":1574390618,"no":0}] 注意属性time抓拍时间，no为同一秒内抓拍的图片编号

备注：分页查询时，某次查询返回条数operator.response.length >= count，则说明仍需下次查询，否则符合条件的文件已查询完。

* + 1. 下载FAT32格式前端存储文件NPPILY.Download.StartFAT32FileDownload(string connectId, string puid, uint sgIndex, uint ivIndex, string remoteFilePath, string localSaveAsFile)

参数：String connectId [NOT NULL] 连接ID

String puid [NOT NULL] 设备PUID

Uint sgIndex [NOT NULL] 前端存储器（SG）资源索引

Uint ivIndex [NOT NULL] 视频（IV）资源索引

String remoteFilePath[NOT NULL] FAT32前端存储文件全路径，由查询结果中path+name属性合成，比如/storage/1F57-15E5/TerminalHelmet/Storage/Record/20191121\_092554.mp4、/storage/emulated/0/TerminalHelmet//Storage/Snapshot/20191121\_104338.jpg等

String localSaveAsFile [NOT NULL] 下载到本地保存的全路径名，如C:/vod.mp4等

调用：NPPILY.Download. StartFAT32FileDownload (connectId, puid, sgIndex, ivIndex, remoteFilePath,localSaveAsFile)

返回值：NPPILY.Struct.ReturnValueStruct

说明：返回值rv节点为常量NrcapError.NRCAP\_SUCCESS时，response节点代表下载通道（流）句柄，这个标识很重要，要注意保存下来，在NPPILY.Enum.NCObjectNotify.stream\_status\_notify的回调函数中，匹配此句柄可以得到对应的下载状态信息，keyData结构为

{

download\_length: ?, // - 已经下载的字节数

download\_time: ?, // - 已经下载的累计时间

bit\_rate: ? // - 下载速率，单位KBps

}

* + 1. 停止下载FAT32格式前端存储文件NPPILY.Download.StopFAT32FileDownload(string connectId, String dlStreamHandle)

参数：String connectId [NOT NULL] 连接ID

String dlStreamHandle [NOT NULL] 下载通道（流）句柄

调用：NPPILY.Download.StopFAT32FileDownload(connectId, dlStreamHandle)

* 1. 调用说明

NPPInterlayer.js需要引用到调用的页面内，同时NPPInterlayer.js依赖于底层脚本库NPPInterface.js文件。NPPInterlayer.js有一个NPPILY全局对象，这个是完成所有NPPILY功能接口的全局对象，不可有其它对象与其重名。

举个例子：如果要想播放视频最简单的几个步骤需要如何做呢？

1. 您需要在页面里引入NPPInterface.js，NPPInterlayer.js
2. 调用初始化函数NPPILY.Init，接口见3.7.1。
3. 初始化成功后，需要连接服务器并获取资源。NPPILY.Connect，接口见3.7.3，调用NPPILY. Connect后，其连接服务器成功后，会返回connectId，这个connectId在获取资源，播放视频的时候都是必须的。
4. 成功连接服务器后，需要获取资源。NPPILY.ForkResource，注意在ForkResource的时候一定要传connectId，ForkResource会返回Array结构的资源列表，结构说明见3.3.8,3.3.9。每次Fork的资源只是当前节点的子资源或当前节点的描述，这个要根据传入的forkLevel决定。
5. 获取完资源后找到资源类型为”IV”资源后再调用播放视点的接口就可以了。
6. 资源的问题解决了，就是窗口了，需要创建一个窗口容器对象WindowContainer，见3.3.6说明，对象放入NPPILY.WindowContainers全局对象里。WindowContainer.window需要通过NPPILY.CreateWindow函数创建，见3.7.6说明。另外，视频大小根据创建的窗口容器的大小决定。
7. 这时候资源和窗口都已经准备完成，调用NPPILY.PlayVideo，见3.7.7接口说明，开始监看视频。
8. 在关闭网页前必须调用NPPILY.Unload方法，清除NPPILY对象，断开与服务器的连接。
9. 附录
   1. 错误码

|  |  |
| --- | --- |
| 错误码(十六进制) | 描述 |
| 0 | 操作成功 |
| -1 | 操作失败 |
| 0x0201 | 发生错误 |
| 0x0203 | 抛出异常错误 |
| 0x0300 | NC插件不是当前匹配版本 |
| 0x0301 | NC插件已初始化 |
| 0x0302 | NC插件未初始化 |
| 0x0304 | Nrcap插件初始化失败 |
| 0x0305 | Nrcap插件初始化抛出异常 |
| 0x0307 | Nrcap插件卸载失败 |
| 0x0309 | NC未加载 |
| 0x030B | WND未加载 |
| 0x030C | WA未加载 |
| 0x030D | 建立连接失败 |
| 0x030F | 连接已经存在 |
| 0x0310 | 正在建立连接中 |
| 0x0311 | 连接已经建立 |
| 0x0315 | 接口未定义错误 |
| 0x0320 | connectId错误 |
| 0x0321 | puid错误 |
| 0x0322 | 中心存储器PUID错误 |
| 0x0323 | 索引错误 |
| 0x0325 | 初始化窗口失败 |
| 0x0327 | 视频窗口不存在 |
| 0x0329 | 播放失败 |
| 0x032A | 窗口没有播放 |
| 0x032C | 设备不在线 |
| 0x032D | 资源句柄不存在或为空 |
| 0x032F | 资源句柄错误 |
| 0x0330 | 喊话或对讲已经存在 |
| 0x0331 | 喊话或对讲互斥存在 |
| 0x0333 | 音频输出资源被占用 |
| 0x0335 | 下载已经存在 |
| 0x0337 | 启动设备搜索失败 |
| 0x0339 | 搜索设备失败 |
| 0x033B | 关闭设备搜索失败 |
| 0x033D | 外接磁盘模块未加载 |
| 0x5FFE | 地址不可达 |
| 0x5ED3 | 用户名或EPID不存在 |
| 0x5ECE | 密码错误 |
| 0x5ECC | 路由失败 |
| 0x5E08 | 没有支持重定向的用户接入服务 |
| 0x6FFB | 连接超时 |
|  | （优先看以上错误码定义，其他的再看后面的） |
|  |  |
| Nrcap7错误定义 | |
| 0xFFFF | 未定义错误 |
| 0x0001 | 已初始化 |
| 0x0002 | 未初始化 |
| 0x0003 | 初始化网络协议栈失败 |
| 0x0004 | 分配内存出错 |
| 0x0005 | 非法的会话句柄 |
| 0x0006 | 非法的设备句柄 |
| 0x0007 | 非法的资源句柄 |
| 0x0008 | 非法的流句柄 |
| 0x0009 | 地址格式错误 |
| 0x000A | 解析响应报文错误 |
| 0x000B | 响应报文格式错误 |
| 0x000C | 错误的资源类型 |
| 0x000D | 缓冲区不足 |
| 0x000E | 错误的输入参数 |
| 0x000F | 创建数据通道失败 |
| 0x0010 | 创建NCS模块失败 |
| 0x0011 | 创建DCS模块失败 |
| 0x0012 | 服务器状态错误 |
| 0x0012 | 上报报文格式错误 |
| 0x0013 | 不支持的请求 |
| 0x0014 | 域名解析失败 |
| 前端板卡返回的错误码 | |
| 0x1801 | 设置值与实际值不匹配 |
| 0x1802 | 属性不支持 |
| 0x1803 | 没有读权限或者不支持读操作 |
| 0x1804 | 没有控制权限 |
| 0x1805 | 不支持交叠,覆盖前面操作 |
| 0x1806 | 资源耗尽,操作失败 |
| 0x1807 | 非法的资源 |
| 0x1808 | 非法的参数（属性值） |
| 0x1809 | 非法的消息体格式 |
| 网元返回的错误码 | |
| 0x2201 | // 错误的消息体格式 |
| 0x2202 | // 错误的参数（属性值） |
| 0x2203 | // 不支持的操作 |
| 0x2204 | // 目标鉴权失败 |
| 0x2205 | // 优先级鉴权失败 |
| 0x2206 | // EPID鉴权失败 |
| 0x2207 | // 操作鉴权失败 |
| 0x2208 | // 资源鉴权失败 |
| 0x2209 | // 命令超时 |
| 0x220A | // 路由失败 |
| 0x2210 | // 没有对象操作权限 |
| 0x2211 | // 对象不存在 |
| 0x2212 | // 对象已存在 |
| 0x2213 | // 超过支持的最大用户数 |
| 0x2214 | // 目标用户支持的操作集过大 |
| 0x2220 | // 请求的令牌不存在 |
| 0x2221 | // 没有可用的分发单元 |
| 0x2222 | // 音频输出通道已被占用 |
| 0x2230 | // 非法的资源 |
| 0x2231 | // 超过最大的流转发负荷 |
| 命令通道返回错误码 | |
| 0x6001 | // 操作正在进行中 |
| 0x6002 | // 函数参数错误 |
| 0x6003 | // 请求超时 |
| 0x6004 | // 状态错误 |
| 0x6005 | // 无扩展参数 |
| 0x5FFF | // 未定义错误,一般不会返回这个 |
| 0x5FFE | // SOCKET连接错误 |
| 0x5FFD | // 连接超时 |
| 0x5FFC | // 内存分配出错 |
| 0x5FFB | // 发送数据时出错 |
| 0x5FFA | // 接收数据时出错 |
| 0x5FF9 | // 操作被中止 |
| 0x5F98 | // 注册时返回的扩展参数长度错误 |
| 0x5F94 | // 注册时返回HTTP头中无Content-Length |
| 0x5F93 | // 注册时返回HTTP头中的Content-Length溢出 |
| 0x5F92 | // 注册时返回HTTP内容中没有XML头 |
| 0x5F91 | // 注册时返回HTTP内容中XML格式错误 |
| 0x5F90 | // 注册时返回的重定向报文中的结果错误 |
| 0x5F37 | // 接收命令时起始字符 |
| 0x5F36 | // 接收命令时版本错误 |
| 0x5F35 | // 接收命令时控制字段错误 |
| 0x5F34 | // 接收命令时消息类型错误 |
| 0x5F33 | // 接收命令时包长度错误 |
| 0x5F32 | // 接收命令时包类型错误 |
| 0x5F31 | // 接收命令时包体长度非8字节整数倍 |
| 0x5ED5 | // 注册时返回未知错误,-1和其他不在范围内的值 |
| 0x5ED3 | // 用户不存在 |
| 0x5ED2 | // 用户被禁用 |
| 0x5ED1 | // EPID被禁用 |
| 0x5ED0 | // 密码错误 |
| 0x5ECF | // 无效的USBKey |
| 0x5ECE | // 用户已满 |
| 0x5ECD | // 认证超时 |
| 0x5ECC | // 路由失败 |
| 0x5ECB | // 无效的AuthKey |
| 0x5B67 | // 重定向时返回未知错误,-1和其他不在范围内的值 |
| 0x5B65 | // 报文格式出错 |
| 0x5B64 | // 不支持的终端类型 |
| 0x5B63 | // 没有在线的PUI或CUI |
| 0x5B62 | // 没有支持重定向的CUI |
| 0x5B61 | // 不支持该厂商ID |
| 数据通道返回错误码 | |
| 0x7001 | // 操作暂时无法完成 |
| 0x7002 | // 帧长度错误 |
| 0x7003 | // 接收到结束包,比如下载完成等 |
| 0x6FFF | // 未定义错误,一般不会返回这个 |
| 0x6FFE | // 参数错误 |
| 0x6FFD | // 无效DC7句柄 |
| 0x6FFC | // 内存错误 |
| 0x6FFB | // 超时 |
| 0x6FFA | // 连接错误 |
| 0x6FF9 | // TCP发送出错 |
| 0x6FF8 | // TCP接收出错 |
| 0x6F9B | // 无效令牌 |
| 0x6F9A | // 通道类型错误 |
| 0x6F99 | // 通道已经存在 |
| 0x6F37 | // 包版本错误 |
| 0x6F36 | // 包类型错误 |
| 0x6F35 | // 包长度错误 |
| 0x6F34 | // 帧开始标志错误 |
| 0x6F33 | // 帧结束标志错误 |
| 0x6F2D | // 帧类型错误 |
| 0x6F2C | // 数据帧长度错误 |
| 0x8001 | // 操作正在进行中 |
| 0x7FFF | // 传入的参数有错 |
| Ox7FFE | // 超时 |
| 0x7FFD | // TCP接收出错 |
| 0x7FFC | // Post报文太长 |
| 0x7FFB | // Post报文格式不对 |
| 0x7FFA | // TCP发送出错 |
| 0x7FF9 | // 挑战失败 |
| 0x7FF8 | // 认证失败 |
| 0x7FF7 | // 不接受数据连接 |
| 0x7FF6 | // 未知错误 |
| 0x9003 | // 接收到空包 |
| 0x9002 | // 帧长度错误,  // 这个表示接收到的帧长度大于DSS7\_RecvRawFrame传入的最大长度  // 上层应该扩大缓冲，或进行相应的处理 |
| 0x9001 | // 操作还在进行中 |
| 0x8FFF | // 分配内存出错 |
| 0x8FFE | // NC7通道出错 |
| 0x8FFD | // 命令响应出错,包括响应不符合协,或其他服务器返回的错误 |
| 0x8FFC | // 参数错误 |
| 0x8FFB | // DC7通道出错 |
| 0x8FFA | // 超时 |
| 0x8FF9 | // 事物号非法 |
| 0x8FF8 | // 未知错误 |
|  |  |

* 1. 辅助函数对象NPPUtils
     1. NPPUtils.Hash哈希方法体结构

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| set | Function(key, value) | 设置节点  (key，value)代表键值对 |
| get | Function(key) | 返回NPPUtils.Hash节点值 |
| unset | Function(key) | 移除某个节点 |
| remove | Function(key) | 移除某个节点 |
| containsKey | Function(key) | 检测某个节点键是否存在 |
| containsValue | Function(value) | 检测某个节点值是否存在 |
| keys | Function() | 返回所有节点键数组 |
| values | Function() | 返回所有节点值数组 |
| size | Function() | 返回节点键值对个数 |
| clear | Function() | 清空所有节点键值对 |
| \_self | Function() | 返回所有节点信息，Object对象 |
| each | Function(iterator, content) | 遍历哈希键值对  iterator遍历方法体，回调时iterator(item, index)，其中item结构为{key:?,value:?}  index为哈希节点自然索引；  content为this的作用域，可省略  注意：iterator返回false或空继续下一循环，返回true跳出循环 |
| any | Function(iterator, content) | 任一项符合条件即返回true，iterator函数中应自行设置条件，如function(item, index) {  return (item.key.length >= 10);  }含义表示是否具有至少一个key的长度不小于10 |
| all | Function(iterator, content) | 所有项均符合条件才返回true，iterator函数中应自行设置条件，如function(item, index) {  return (item.key.length >= 10);  }含义表示是否所有key的长度不小于10 |

* + 1. NPPUtils.Array数组辅助对象

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| indexOf | | function(array,value,from) | 查找首次匹配索引  array为某个数组对象  value为需要匹配查找的值  from为开始查找的索引，可省略，从0查起 |
| lastIndexOf | | function(array,value,from) | 查找最后匹配索引  array为某个数组对象  value为需要匹配查找的值  from为开始查找的索引，可省略，从0查起 |

* + 1. NPPUtils.Timer定时器对象

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| interval | Number | 轮询周期频率（勿动）  暂定100ms每次 |
| count | Number | 轮询计数（勿动） |
| timer | Object | 总定时器对象（勿动）  以interval为周期轮询一次 |
| events | NPPUtils.Hash | 存放轮询侦测任务（勿动） |
| Start | Function() | 开启定时器对象  全局可以只调用一次，或者任意调用 |
| Stop | Function() | 停止定时器对象  NPPILY.UnLoad中会自动调用 |
| Set | Function(ev, cb) | 注册轮询任务  ev(string) 某类轮询任务的标识  cb(object)  {  name(string) 子任务名称  fu(function) 子任务回调函数  interval(number) 子任务执行间隔，ms单位  }  如：Set(  “download”, {  name: “getStatus”  fu: ...  interval: 1000  }  ) |
| ContainsKey | Function(ev, cbName) | 检测某个轮询子任务是否存在  如ContainsKey(“download”, “getStatus”) |
| UnSet | Function(ev, cbName) | 移除某个轮询子任务 |
| Call | Function() | 总定时器轮询函数（勿动） |

* + 1. NPPUtils.Log对象

如果NPPILY注册开启了调试，那么可以使用NPPUtils.Log(fnString, logString)函数，如NPPUtils.Log(“NPPILY.Init”, “NPPILY has inited~”)

* + 1. NPPUtils.DateFormat(mask,date)格式化返回当前客户端系统时间
    2. NPPUtils.GetDateTimeUTCSeconds(date)返回UTC时间戳
    3. NPPUtils.DTStrToTimestamp(dateStr)标准的时间字符串转为时间戳
    4. NPPUtils.Zeroize(value,length)根据长度左补零
    5. NPPUtils.CheckByteLength(value, minlen, maxlen)验证字串长度
    6. NPPUtils.GetStringRealLength(source) 中英文混合字符串真实长度
    7. NPPUtils.Regexs

|  |  |  |
| --- | --- | --- |
| 变量 | 类型 | 描述 |
| uint | RegExp | 验证数值 |
| domain | String | 验证ip或域名，需要new RegExp(domain) |
| puid | RegExp | 验证PUID |
| strip | RegExp | 去除字符串两端空格 |

* + 1. NPPUtils.UTF8toUnicode utf8编码字符转为unicode编码
    2. NPPUtils.UnicodetoUTF8 unicode编码字符转为utf8编码
    3. NPPUtils.UnicodetoGB2312 unicode编码字符转为GB2312编码
    4. NPPUtils.GB2312toUnciode GB2312编码字符转为unicode编码
  1. 文件操作对象NPPILY.Folder

注意：Folder对象是封装在NPPILY下面的对象，放在附录里为便于查找

* + 1. NPPILY.Folder.GetSystemRoot获取操作系统根目录

参数：空

调用：NPPILY.Folder.GetSystemRoot();

* + 1. NPPILY.Folder.DebugSwitch(boolean bValue)切换插件调试

参数：boolean bValue 1/true开启，0/false关闭（缺省）

调用：NPPILY.Folder.DebugSwitch(bValue);

* + 1. NPPILY.Folder.GetFileFolder()打开目录选择对话框
    2. NPPILY.Folder.OpenFolder(string folderPath)打开文件夹

参数：string folderPath 文件夹全路径，如C:/SaveFolder

调用：NPPILY.Folder.OpenFolder(folderPath);

* + 1. NPPILY.Folder.CreateDirectory(string folderPath)创建文件夹

参数：string folderPath 文件夹全路径，如C:/SaveFolder

调用：NPPILY.Folder.CreateDirectory(folderPath);

说明：如果目录没有存在将会创建一个，存在的不会再覆盖创建

* + 1. NPPILY.Folder.DeleteDirectory(string folderPath)删除文件夹

参数：string folderPath 文件夹全路径，如C:/SaveFolder

调用：NPPILY.Folder.DeleteDirectory(folderPath);

* + 1. NPPILY.Folder.DeleteFile(string fileName)删除文件

参数：string fileName需要删除的文件全路径名称，如C:/SaveFolder/Snapshot.bmp

调用：NPPILY.Folder.DeleteFile(fileName);

* + 1. NPPILY.Folder.FileExist(string fileName)检测文件是否存在

参数：string fileName文件全路径名称，如C:/SaveFolder/Snapshot.bmp

调用：NPPILY.Folder.FileExist(fileName);

返回值：rv为0时，response才有意义，response=1存在、=0不存在；rv为其他值时，表示调用不成功。

* + 1. NPPILY.Folder.ReadFile(string fileName)读文件内容

参数：string fileName文件全路径名称，如C:/SaveFolder/123.txt

调用：NPPILY.Folder.ReadFile(fileName);

* + 1. NPPILY.Folder.WriteFile(string fileName, string content, uint contentLength, boolean bValue)写文件内容

参数：String fileName文件全路径名称，如C:/SaveFolder/123.txt

String content 写的具体内容

Uint contentLength 写的内容真实长度

Boolean bValue 1/true写追加，0/false写覆盖（缺省）

调用：NPPILY.Folder.WriteFile(fileName, content, contentLength, bValue);

说明：文件不存在可创建，前提是上级目录首先存在

* + 1. NPPILY.Folder.GetFolderFiles(string folderPath, fileType, bValue, bReturnFullPath)按类型获取目录下的文件列表

参数：String folderPath文件全路径名称，如C:/SaveFolder

String fileType为文件的类型，如"avi/AVI", "jpg", "doc"等等（不可省略）

Boolean bValue 是否深度递归获取，缺省为0否

bReturnFullPath(uint) 是否返回全路径，缺省为0否（只有文件名，不含上级目录名）

调用：NPPILY.Folder.GetFolderFiles(folderPath, fileType, bValue, bReturnFullPath);

说明：如果fileType为某个具体的类型时，理论上将返回此类型的全部文件

* + 1. NPPILY.Folder.GetFolders(string folderPath, bValue, bReturnFullPath)获取目录下的文件夹列表

参数：String folderPath文件全路径名称，如C:/SaveFolder

Boolean bValue 是否深度递归获取，缺省为0否

bReturnFullPath(uint) 是否返回全路径，缺省为0否（只有文件名，不含上级目录名）

调用：NPPILY.Folder.GetFolders(folderPath, bValue, bReturnFullPath);

* + 1. NPPILY.Folder.Base64Enc(string segment)进行Base64编码

参数：String segment要Base64编码的字符串内容

调用：NPPILY.Folder.Base64Enc(segment);

说明：成功时返回值response应为UTF8 Base64编码后的字符串

* + 1. NPPILY.Folder.Base64Dec(string base64EnStr, boolean bUsingUTF8)进行Base64解码

参数：String base64EnStr要Base64解码的字符串

bUsingUTF8 是否使用UTF8编码的Base64解码，缺省为1是

调用：NPPILY.Folder.Base64Dec(base64EnStr, bUsingUTF8);

说明：也可以对gb2312 Base64编码字符串的解码，那么bUsingUTF8置为0

* + 1. NPPILY.Folder.DesEnc(string base64Raw, string base64Key)Des编码
    2. NPPILY.Folder.DesDec(string base64EnStr, string base64KeyStr)Des解码
    3. NPPILY.Folder.GetRandTokenForUser(string base64EnStr, string keyStr)
    4. NPPILY.Folder.SetRandTokenForUser(string rawData, string keyStr)
    5. NPPILY.Folder.ReadFileEx(string fileName)读取指定文件内容，经Base64编码后返回

参数：string fileName文件全路径名称，如C:/SaveFolder/Snapshot.bmp

调用：NPPILY.Folder.ReadFileEx(fileName);

说明：可以读一些较小文件，如对图片读出Base64数据可以用于前端页面上传后台数据处理

* + 1. NPPILY.Folder.CopyFile(string srcFileName, string dstFileName)将文件复制到指定目录下

参数：string srcFileName需要复制的文件全路径，如C:/SaveFolder/Snapshot.bmp

String dstFileName 保存到目标目录下文件名，可以只是文件名，将放在srcFileName同一目录下

调用：NPPILY.Folder.CopyFile(srcFileName, dstFileName);

返回值：rv=0，response才有意义，response=0表复制成功（对于某些不合法的目标文件名，实际复制是不成功的），=-1源文件应传入绝对路径，=-2源文件不存在；rv为其他值时，表示调用不成功

* + 1. NPPILY.Folder.StartHttpDownload(string url, string saveDir, boolean autoRun)开始下载解码库插件

参数：string url网络地址，如http://127.0.0.1/npsdk/CreMedia7.0\_Plugin\_Setup.exe

String saveDir本地下载保存目录，如C:/

Boolean autoRun 下载完成后是否自动运行，1是0否，默认为1

调用：NPPILY.Folder.StartHttpDownload(url, saveDir, autoRun);

* + 1. NPPILY.Folder.CloseHttpDownload()停止下载解码库插件
    2. NPPILY.Folder.GetStatus()获取解码库插件下载状态

调用：NPPILY.Folder.GetStatus();

说明：开始下载后，上层需要定时侦测下载的状态，当检测到下载完成时应停止下载；侦测到下载完毕后，可以停止下载。

返回值：response结构为NPPILY.Struct.HttpDownloadStatusStruct

* + 1. NPPILY.Folder.GetPluginVersion()获取解码库插件版本号

调用：NPPILY.Folder.GetPluginVersion();

说明：获取解码库插件的版本号，如果响应为空或者版本号不是最新的（上层应规定一个版本，每次可以验证一下），那么版本太低需执行下载更新

* + 1. NPPILY.Folder.PUDetectStartup()开启设备搜索功能
    2. NPPILY.Folder.PUDetectGetPUList()搜索设备列表

调用：NPPILY.Folder.PUDetectGetPUList();

说明：搜索设备时上层调用此接口即可，内部会检测并开启设备搜索功能；可能需要周期刷新调用获取设备列表，第一次成功调用时很可能会返回空的数组（response节点值），故最好紧接着再次调用，以后刷新一次调用一次即可！

返回值：response可能结构为[Object, ...]，Object结构如下

{

"ip"(string) IP地址

"mac"(string) 设备MAC

"name"(string) 设备名称

"model"(string) 设备类型

"subnet\_mask(string) 子网掩码

"enable\_dhcp"(string) 是否使能DHCP

"port"(string) 端口

"hardware\_ver"(string) 硬件版本

"hardware\_model"(string) 硬件型号

"software\_ver"(string) 软件版本

"puid"(string) 设备PUID

}

* + 1. NPPILY.Folder.PUDetectCleanup()关闭设备搜索功能

说明：上层可以不必调用此接口来关闭设备搜索，因为在卸载NPPILY.UnLoad()时会自行关闭。