

# 设备(智能交通终端)

网络 SDK 编程指南

V5.2

# 声 明

非常感谢您购买我公司的产品，如果您有什么疑问或需要请随时联系我们。

- 我们已尽量保证手册内容的完整性与准确性，但也不免出现技术上不准确、与产品功能及操作不相符或印刷错误等情况，如有任何疑问或争议，请以我司最终解释为准。
- 产品和手册将实时进行更新，恕不另行通知。
- 本手册中内容仅为用户提供参考指导作用，请以 SDK 实际内容为准。

# 目 录

声 明 .....	I
目 录 .....	II
1 SDK 简介 .....	1
2 SDK 版本更新 .....	4
3 函数调用顺序 .....	8
3.1 SDK 接口调用主要流程 .....	8
3.2 报警模块流程 .....	9
3.2.1 报警（布防）流程 .....	9
3.2.2 报警（监听）流程 .....	10
3.3 IP 通道配置 .....	11
4 函数调用实例 .....	12
4.1 交通数据接收示例代码 .....	12
5 函数说明 .....	17
5.1 SDK 初始化 .....	17
5.1.1 初始化 SDK NET_DVR_Init .....	17
5.1.2 释放 SDK 资源 NET_DVR_Cleanup .....	17
5.2 SDK 本地功能 .....	17
SDK 本地参数配置 .....	17
5.2.1 获取 SDK 本地参数 NET_DVR_GetSDKLocalCfg .....	17
5.2.2 设置 SDK 本地参数 NET_DVR_SetSDKLocalCfg .....	18
连接和接收超时时间及重连设置 .....	19
5.2.3 设置网络连接超时时间和连接尝试次数 NET_DVR_SetConnectTime .....	19
5.2.4 设置重连功能 NET_DVR_SetReconnect .....	19
5.2.5 设置接收超时时间 NET_DVR_SetRecvTimeOut .....	19
多网卡绑定 .....	19
5.2.6 获取所有 IP，用于支持多网卡接口 NET_DVR_GetLocalIP .....	19
5.2.7 设置 IP 绑定 NET_DVR_SetValidIP .....	20
SDK 版本、状态和能力 .....	20
5.2.8 获取 SDK 的版本号和 build 信息 NET_DVR_GetSDKBuildVersion .....	20
5.2.9 获取当前 SDK 的状态信息 NET_DVR_GetSDKState .....	20
5.2.10 获取当前 SDK 的功能信息 NET_DVR_GetSDKAbility .....	20
SDK 启用写日志 .....	21
5.2.11 启用写日志文件 NET_DVR_SetLogToFile .....	21
异常消息回调 .....	21
5.2.12 注册接收异常、重连等消息的窗口句柄或回调函数 NET_DVR_SetExceptionCallBack_V30 .....	21
获取错误信息 .....	23
5.2.13 返回最后操作的错误码 NET_DVR_GetLastError .....	23
5.2.14 返回最后操作的错误码信息 NET_DVR_GetErrorMsg .....	24
5.3 用户注册 .....	24
5.3.1 激活设备 NET_DVR_ActivateDevice .....	24

5.3.2	通过解析服务器，获取设备的动态 IP 地址和端口号 NET_DVR_GetDVRIPByResolveSvr_EX.....	24
5.3.3	用户注册设备 NET_DVR_Login_V40 .....	25
5.3.4	用户注销 NET_DVR_Logout.....	25
5.4	获取设备能力集.....	25
5.4.1	获取设备能力集 NET_DVR_GetDeviceAbility .....	25
5.5	实时预览.....	26
5.5.1	设置播放显示模式 NET_DVR_SetShowMode.....	26
5.5.2	主码流动态产生一个关键帧 NET_DVR_MakeKeyFrame .....	27
5.5.3	子码流动态产生一个关键帧 NET_DVR_MakeKeyFrameSub .....	27
5.5.4	实时预览 NET_DVR_RealPlay_V40 .....	27
5.5.5	停止预览 NET_DVR_StopRealPlay .....	28
5.5.6	获取预览时用来解码和显示的播放库句柄 NET_DVR_GetRealPlayerIndex .....	28
5.6	预览时解码效果的参数控制.....	29
5.6.1	设置播放库的帧缓冲区个数 NET_DVR_SetPlayerBufNumber.....	29
5.7	预览时播放声音控制.....	29
5.7.1	设置声音播放模式 NET_DVR_SetAudioMode .....	29
5.7.2	独占声卡模式下开启声音 NET_DVR_OpenSound .....	29
5.7.3	独占声卡模式下开启声音 NET_DVR_CloseSound .....	30
5.7.4	共享声卡模式下开启声音 NET_DVR_OpenSoundShare .....	30
5.7.5	共享声卡模式下关闭声音 NET_DVR_CloseSoundShare .....	30
5.7.6	调节播放音量 NET_DVR_Volume.....	30
5.8	实时预览数据捕获.....	31
5.8.1	注册回调函数，捕获实时码流数据 NET_DVR_SetRealDataCallBack.....	31
5.8.2	捕获数据并保存到指定的文件中 NET_DVR_SaveRealData.....	31
5.8.3	停止数据捕获 NET_DVR_StopSaveRealData .....	32
5.9	布防、撤防.....	32
	设置报警等信息上传的回调函数 .....	32
5.9.1	注册回调函数，接收设备报警消息 NET_DVR_SetDVRMessageCallBack_V31.....	32
	布防撤防 .....	33
5.9.2	建立报警上传通道，获取报警等信息 NET_DVR_SetupAlarmChan_V41 .....	33
5.9.3	撤销报警上传通道 NET_DVR_CloseAlarmChan_V30 .....	33
5.10	监听报警.....	33
5.10.1	启动监听，接收设备主动上传的报警等信息 NET_DVR_StartListen_V30 .....	33
5.10.2	停止监听（支持多线程）NET_DVR_StopListen_V30 .....	35
5.11	远程参数配置.....	35
5.11.1	获取设备的配置信息 NET_DVR_GetDVRConfig .....	35
5.11.2	设置设备的配置信息 NET_DVR_SetDVRConfig.....	36
5.12	批量配置参数.....	37
5.12.1	批量获取配置信息 NET_DVR_GetDeviceConfig .....	37
5.12.2	批量设置配置信息 NET_DVR_SetDeviceConfig.....	38
5.13	长连接参数配置.....	39
5.13.1	启动长连接远程配置 NET_DVR_StartRemoteConfig.....	39
5.13.2	逐个获取查找到的结果信息 NET_DVR_GetNextRemoteConfig .....	41
5.13.3	关闭长连接配置接口所创建的句柄，释放资源 NET_DVR_StopRemoteConfig .....	41

5.14	远程控制.....	42
5.14.1	远程控制 NET_DVR_RemoteControl.....	42
5.15	录像回放和下载.....	42
5.15.1	按时间回放录像文件 NET_DVR_PlayBackByTime_V40.....	42
5.15.2	按时间下载录像文件 NET_DVR_GetFileByTime_V40.....	42
5.15.3	控制录像回放的状态 NET_DVR_PlayBackControl_V40.....	43
	回放录像文件时的数据捕获.....	44
5.15.4	捕获回放的录像数据, 并保存成文件 NET_DVR_PlayBackSaveData.....	44
5.15.5	停止保存录像数据 NET_DVR_StopPlayBackSave.....	44
5.15.6	注册回调函数, 捕获录像数据 NET_DVR_SetPlayDataCallBack_V40.....	44
5.16	图片下载.....	45
5.16.1	获取图片数据并存放在指定的内存空间中 NET_DVR_GetPicture_V30.....	45
5.17	设备维护管理.....	45
	远程升级.....	45
5.17.1	设置远程升级时网络环境 NET_DVR_SetNetworkEnvironment.....	45
5.17.2	远程升级 NET_DVR_Upgrade.....	46
5.17.3	获取远程升级的进度 NET_DVR_GetUpgradeProgress.....	46
5.17.4	获取远程升级的状态 NET_DVR_GetUpgradeState.....	46
5.17.5	关闭远程升级句柄, 释放资源 NET_DVR_CloseUpgradeHandle.....	46
	日志查找.....	47
5.17.6	查找设备的日志信息 NET_DVR_FindDVRLog_V30.....	47
5.17.7	逐条获取查找到的日志信息 NET_DVR_FindNextLog_V30.....	47
5.17.8	释放查找日志的资源 NET_DVR_FindLogClose_V30.....	47
	恢复设备默认参数.....	48
5.17.9	恢复设备默认参数 NET_DVR_RestoreConfig.....	48
	导入/导出配置文件.....	48
5.17.10	导出配置文件 NET_DVR_GetConfigFile_V30.....	48
5.17.11	导出配置文件 NET_DVR_GetConfigFile.....	48
5.17.12	导入配置文件 NET_DVR_SetConfigFile_EX.....	48
5.17.13	导入配置文件 NET_DVR_SetConfigFile.....	49
	关机和重启.....	49
5.17.14	重启设备 NET_DVR_RebootDVR.....	49
5.17.15	关闭设备 NET_DVR_ShutDownDVR.....	49
5.18	获取设备支持的云台协议.....	50
5.18.1	获取设备支持的云台协议 NET_DVR_GetPTZProtocol.....	50
5.19	获取 IPC 协议列表.....	50
5.19.1	获取设备支持的 IPC 协议表 NET_DVR_GetIPCProtoList.....	50
5.20	获取设备状态.....	50
5.20.1	获取设备状态信息 NET_DVR_GetDeviceStatus.....	50
6	错误代码及说明.....	52
6.1	网络通讯库错误码.....	52
6.2	RTSP 通讯库错误码.....	55
6.3	软解码库错误码.....	55
7	附录:结构体.....	57

# 1 SDK 简介

设备网络 SDK 是基于设备私有网络通信协议开发的，为嵌入式网络硬盘录像机、NVR、视频服务器、网络摄像机、网络球机、智能交通等网络产品服务的配套模块，用于远程访问和控制设备软件的二次开发。本文档主要介绍智能交通相关的功能，适用于但不仅限于以下产品型号：

终端服务器：TS-50XX 等

视频分析仪：VAR-50XX、DS-TP5004-D、DS-TP6200-HD、DS-TP6200-HV 等

停车场管理系统：DS-TP5016-P 等

出入口终端服务器：DS-TP5008-E 等

该文档仅介绍设备网络 SDK 主要功能：实时码流预览、参数配置以及获取上传的抓拍结果信息以及录像回放等功能。更多功能说明请参见《设备网络 SDK 使用手册.chm》。

设备网络 SDK 包含网络通讯库、播放库等功能组件，我们提供 Windows 和 Linux 两个版本的 SDK，各自所包含的组件如下：

表 1.1 Windows SDK 组件

网络通讯库	外部接口	HCNetSDK.h	头文件	
		HCNetSDK.lib	LIB 库文件	
		HCNetSDK.dll	DLL 库文件	
	核心组件	HCCore.lib	LIB 库文件	
		HCCore.dll	DLL 库文件	
组件库	设备配置核心组件	HCCoreDevCfg.dll	DLL 库文件	HCNetSDKCom 文件夹
	预览组件	HCPreview.lib	LIB 库文件	HCNetSDKCom 文件夹
		HCPreview.dll	DLL 库文件	HCNetSDKCom 文件夹
	回放组件	HCPlayBack.dll	DLL 库文件	HCNetSDKCom 文件夹
	语音组件	HCVoiceTalk.dll	DLL 库文件	HCNetSDKCom 文件夹
	报警组件	HCAalarm.lib	LIB 库文件	HCNetSDKCom 文件夹
		HCAalarm.dll	DLL 库文件	HCNetSDKCom 文件夹
	显示组件	HCDisplay.dll	DLL 库文件	HCNetSDKCom 文件夹
	行业应用管理配置组件	HCIIndustry.dll	DLL 库文件	HCNetSDKCom 文件夹
	维护管理配置组件	HCGeneralCfgMgr.lib	LIB 库文件	HCNetSDKCom 文件夹
		HCGeneralCfgMgr.dll	DLL 库文件	HCNetSDKCom 文件夹
RTSP 通讯库		StreamTransClient.dll	DLL 库文件	HCNetSDKCom 文件夹
转封装库		SystemTransform.dll	DLL 库文件	HCNetSDKCom 文件夹
字符转码库		libiconv2.dll	DLL 库文件	HCNetSDKCom 文件夹
模拟能力集		LocalXml.zip	XML 文件包	
帧分析库		AnalyzeData.dll	DLL 库文件	HCNetSDKCom 文件夹

语音对讲库		AudioIntercom.dll	DLL 库文件	HCNetSDKCom 文件夹
		OpenAL32.dll	DLL 库文件	HCNetSDKCom 文件夹
播放库	核心库文件	PlayM4.h、WindowsPlayM4.h	头文件	
		PlayCtrl.lib	LIB 库文件	
		PlayCtrl.dll	DLL 库文件	
	视频渲染库	SuperRender.dll	DLL 库文件	
	音频渲染库	AudioRender.dll	DLL 库文件	
	小鹰眼库	EagleEyeRender.dll	DLL 库文件	
	GPU 硬解码库	HWDecode.dll	DLL 库文件	
	鱼眼库	MP_Render.dll	DLL 库文件	
	视频后处理库	MP_VIE.dll	DLL 库文件	
	测温信息抓图库	YUVProcess.dll	DLL 库文件	
	DirectX 组件库	D3DCompiler_43.dll	DLL 库文件	

表 1.2 Linux SDK 组件

网络通讯库	外部接口	HCNetSDK.h	头文件	
		libhcnetSDK.so	SO 库文件	
	核心组件	libHCCore.so	SO 库文件	
组件库	设备配置核心组件	libHCCoreDevCfg.so	SO 库文件	HCNetSDKCom 文件夹
	预览组件	libHCPreview.so	SO 库文件	HCNetSDKCom 文件夹
	回放组件	libHCPlayBack.so	SO 库文件	HCNetSDKCom 文件夹
	语音组件	libHCVoiceTalk.so	SO 库文件	HCNetSDKCom 文件夹
	报警组件	libHCAAlarm.so	SO 库文件	HCNetSDKCom 文件夹
	显示组件	libHCDisplay.so	SO 库文件	HCNetSDKCom 文件夹
	行业应用管理配置组件	libHCIndustry.so	SO 库文件	HCNetSDKCom 文件夹
	维护管理配置组件	libHCGeneralCfgMgr.so	SO 库文件	HCNetSDKCom 文件夹
hpr 库		libhpr.so	SO 库文件	
RTSP 通讯库		libStreamTransClient.so	SO 库文件	HCNetSDKCom 文件夹
转封装库		libSystemTransform.so	SO 库文件	HCNetSDKCom 文件夹
字符转码库		libiconv2.so	SO 库文件	HCNetSDKCom 文件夹
帧分析库		libanalyzedata.so	SO 库文件	HCNetSDKCom 文件夹
播放库	核心库文件	PlayM4.h、LinuxPlayM4.h	头文件	
		libPlayCtrl.so	SO 库文件	
	视频渲染库	libSuperRender.so	SO 库文件	

	音频渲染库	libAudioRender.so	SO 库文件	
--	-------	-------------------	--------	--

本版本的设备网络 SDK 开发包中包含以上各个组件, **HCNetSDK.dll、HCCore.dll 必须加载**(对于 Linux SDK, 即 libhcnetsdk.so、libHCCore.so), 其他组件, 用户可以根据需要选择其中的一部分或者全部, 以下将对各个组件在 SDK 中的作用和使用条件分别说明。

- **网络通讯库:** 设备网络 SDK 的主体, 主要用于网络客户端与各类产品之间的通讯交互, 负责远程功能调控, 远程参数配置及码流数据的获取和处理等。设备网络 SDK V5.0 针对产品应用业务进行细化, 对之前版本的 SDK 的功能模块进行组件化, 其中外部接口 (HCNetSDK.dll) 仍然保持和设备网络 SDK V4.x 版本保存一致(向下兼容), 其他单独的业务功能 (预览、回放等) 可以加载单独的模块组件, 多个业务功能也可以组合使用。**更新 SDK 时, HCNetSDK.dll、HCCore.dll 以及 HCNetSDKCom 文件夹下的功能组件库文件都需要更新加载, 且 HCNetSDKCom 文件夹名不能修改。**
- **hpr 库:** 网络通讯库的依赖库, Linux SDK 使用时和网络通讯库同时加载。
- **RTSP 通讯库:** 支持 RTSP 传输协议的网络库。当需要对支持 RTSP 协议的产品进行取流等操作时必须加载该项组件。
- **转封装库:** 库的功能可以分为两种: 一种是将标准码流转换成采用我们公司封装格式的码流。当用户需要对支持 RTSP 协议的产品捕获采用本公司封装格式的码流数据时 (即当设置 NET\_DVR\_RealPlay\_V40 接口中的回调函数捕获数据或者调用 NET\_DVR\_SetRealDataCallBack 接口捕获数据时) 必须加载该组件。另一种功能是将标准码流转换成其他格式的封装, 如 3GPP、PS 等。例如, 当用户需要对支持 RTSP 协议的产品实时捕获指定封装格式的码流数据 (对应的 SDK 接口为 NET\_DVR\_SaveRealData) 时必须加载该项组件。
- **语音对讲库:** 用于语音对讲时通过声卡采集数据并按照指定的编码格式编码码流或者解码播放音频码流数据 (不带封装格式的码流数据)。V4.2.2.5 及以前版本 SDK 均采用 windows API 实现相关功能。之后版本默认使用语音对讲库的方式, 通过接口 NET\_DVR\_SetSDKLocalCfg 可以选择之前的 windows API 模式。OpenAL32.dll 为依赖库, 语音对讲库模式下必须加载。**Windows64 位或者 Linux 系统下无语音对讲功能。**
- **字符转换库:** 电脑字符集和设备字符集不一致时, SDK 内部需要进行字符编码转换, SDK 默认使用 libiconv 库进行类型转换。如果用户不想使用 libiconv 编码库, 可以调用 NET\_DVR\_SetSDKLocalCfg (类型: NET\_SDK\_LOCAL\_CFG\_TYPE\_BYTE\_ENCODE) 设置字符转码回调函数, 将用户自己的字符编码接口告知 SDK, 然后 SDK 将使用用户提供的字符编码接口进行字符串处理。
- **模拟能力集:** 如果需要获取设备能力集 (NET\_DVR\_GetDeviceAbility), 建议调用 NET\_DVR\_SetSDKLocalCfg 启用模拟能力集, 此时需要加载 LocalXml.zip (要求和网络通讯库放在同一个目录下)。
- **帧分析库:** 用于分析视音频帧数据, 调用 NET\_DVR\_SetESRealPlayCallBack、NET\_DVR\_SetPlayBackESCallBack 设置裸码流回调函数等接口时, 必须加载该库文件。
- **播放库:** 主要用于对实时码流数据进行解码显示 (实现预览功能) 和对录像文件进行回放解码等。用户如果需要在 SDK 内部进行对实时流和录像码流播放显示时 (即 NET\_DVR\_RealPlay\_V40 接口的第二个结构体参数的播放句柄设置成有效句柄时) 必须加载该组件, 而如果用户仅需要用网络通讯库捕获到数据后再外部自行处理就不需要加载该组件, 这种情况下用户在外部分自行解码将更灵活, 可参见软解码库函数说明《播放器 SDK 编程指南》。

## 2 SDK 版本更新

### Version 5.1.1.10 (build20150429)

- TS-5004-A V4.0.4
- 新增交通数据查询功能（对应接口：[NET\\_DVR\\_StartRemoteConfig](#)）：  
命令：NET\_DVR\_GET\_TRAFFIC\_DATA。
- 新增交通流量查询功能（对应接口：[NET\\_DVR\\_StartRemoteConfig](#)）：  
命令：NET\_DVR\_GET\_TRAFFIC\_FLOW。
- 新增支持图片下载接口：  
[NET\\_DVR\\_GetPicture\\_V30](#)。

### Version 5.0.3.15 (build20150129)

- 事件取证分析仪 V4.1.0
- [NET\\_DVR\\_TFS\\_ALARM](#)(违章取证报警信息)使用 22 个保留字节新增参数：byVehicleAttribute(车辆属性)、byPilotSafebelt(主驾驶员是否系安全带)、byCopilotSafebelt(副驾驶员是否系安全带)、byPilotSunVisor(主驾驶是否打开遮阳板)、byCopilotSunVisor(副驾驶是否打开遮阳板)、byPilotCall(主驾驶员是否在打电话)。
- [NET\\_DVR\\_VEHICLE\\_INFO](#)(车辆信息参数)使用 3 个保留字节新增参数：byVehicleModel(车辆子品牌年款)、wVehicleLogoRecog(车辆主品牌)。
- [TRAFFIC\\_AID\\_TYPE](#)(交通事件类型)新增枚举类型：VEHICLE\_OCCUPANCY\_NONVEHICLE(机动车占用非机动车位)、GASSER(加塞)。

### Version 5.0.3.10 (build20150112)

- iDS-TP6200-HP V4.0.2
- 新增车辆二次识别任务提交功能（对应接口：[NET\\_DVR\\_SetDeviceConfig](#)）：  
命令：NET\_DVR\_SET\_VEHICLE\_RECOG\_TASK。
- 新增车辆二次识别任务获取功能（对应接口：[NET\\_DVR\\_StartRemoteConfig](#) 和 [NET\\_DVR\\_GetNextRemoteConfig](#)）：命令：NET\_DVR\_GET\_VEHICLE\_RECOG\_TASK。
- 新增车辆二次识别数据上传信息类型（对应接口：[NET\\_DVR\\_SetDVRMessageCallBack\\_V31](#) 和 [NET\\_DVR\\_StartListen\\_V30](#)）：COMM\_VEHICLE\_RECOG\_RESULT。
- 新增车辆二次识别检测能力集（对应接口：[NET\\_DVR\\_GetDeviceAbility](#)，对应能力集类型：DEVICE\_ABILITY\_INFO）：VehicleRecogAbility。
- [NET\\_DVR\\_SETUPALARM\\_PARAM](#)(报警布防参数)使用 2 个保留字节新增参数：wTaskNo(任务处理号)。
- 设备软硬件能力集([DEVICE\\_SOFTHARDWARE\\_ABILITY](#))新增节点：<VehicleRecogAbility>(支持车辆二次检测能力)。

### Version 4.3.0.4 (build20140610)

- 更新原因：VAR-500X-X(D) V4.0.2
- [TRAFFIC\\_AID\\_TYPE](#)(交通事件类型)新增枚举类型：OVERLINE(压线)、LANECHANGE(变道)、TURNAROUND(掉头)。
- [NET\\_DVR\\_AID\\_ALARM\\_V41](#)(交通事件报警信息)使用 97 个保留字节新增参数：byLaneNo(关联车道号)、byMonitoringSiteID[MONITORSITE\_ID\_LEN](监测点编号)、byDeviceID[DEVICE\_ID\_LEN](设备编号)。
- [NET\\_DVR\\_TFS\\_ALARM](#)(违章取证报警信息)使用 2 个保留字节新增参数：bySpecificVehicleType(车辆类型)、

byLaneNo(关联车道号)。

- [NET\\_DVR\\_ONE\\_AID\\_RULE\\_V41](#)(单条交通事件规则)使用 1 个保留字节新增参数: byLaneNo(关联车道号)。
- [NET\\_DVR\\_LANE\\_PARAM\\_V41](#)(车道参数)使用 5 个保留字节新增参数: byLaneNo(关联车道号)、dwOversizeVehicle(大型车数量)。
- [NET\\_DVR\\_TPS\\_ALARM\\_V41](#)(交通统计上传报警信息)使用 104 个保留字节新增参数: byMonitoringSiteID[MONITORSITE\_ID\_LEN](监测点编号)、byDeviceID[DEVICE\_ID\_LEN](设备编号)、dwStartTime(开始统计时间)、dwStopTime(结束统计时间)。
- 设备软硬件能力集(DEVICE\_SOFTHARDWARE\_ABILITY)新增节点: <VcaChanAbility>。
- 设备通用能力集(DEVICE\_ABILITY\_INFO), 其中智能通道分析能力(VcaChanAbility)扩展, 节点<AidRule>: 新增子节点<laneNo>; <eventType>新增取值: overLine、illegalLaneChange、turnAround。

### Version 4.2.8.5 (build20140317)

- 更新原因: TS-5012-X V4.0.5
- 新增云存储配置功能 (对应接口: [NET\\_DVR\\_GetDeviceConfig](#) 和 [NET\\_DVR\\_SetDeviceConfig](#)): 命令: NET\_DVR\_GET\_CLOUDSTORAGE\_CFG、NET\_DVR\_SET\_CLOUDSTORAGE\_CFG。
- [NET\\_ITS\\_PICTURE\\_INFO](#)(抓拍图片信息)使用 1 个保留字节新增参数: byDataType(数据上传方式, 支持上传云存储 URL), 参数 byType 新增取值: 4-二值图、5-码流。
- [NET\\_DVR\\_AID\\_ALARM\\_V41](#)(交通事件报警信息)使用 3 个保留字节新增参数: byDataType(数据上传方式, 支持上传云存储 URL)、wMilliSecond(时标毫秒)。
- 设备通用能力集(DEVICE\_ABILITY\_INFO), 其中智能交通设备能力(ITDeviceAbility)扩展, <ITSAbility>新增子节点<supportMilliCheckTime>、<CloudStorage>。

### Version 4.2.7.6(build20131231)

- DS-TP5008-E(出入口终端服务器) V1.1.0。
- 新增批量配置功能 (对应接口 [NET\\_DVR\\_GetDeviceConfig](#) 和 [NET\\_DVR\\_SetDeviceConfig](#)): 命令: NET\_DVR\_GET\_ITS\_EXDEVCFG(获取 ITS 外接设备信息)、NET\_DVR\_SET\_ITS\_EXDEVCFG(设置 ITS 外接设备信息)。
- 新增连接配置功能 (对应接口 [NET\\_DVR\\_StartRemoteConfig](#) 和 [NET\\_DVR\\_GetNextRemoteConfig](#)): 命令: NET\_DVR\_GET\_ITS\_EXDEVSTATUS(获取 ITS 所有外接设备信息)。
- 新增远程控制功能 (对应接口 [NET\\_DVR\\_RemoteControl](#)): 命令: NET\_DVR\_SET\_ITS\_ENDEVCMDC(控制下发卡片信息)、NET\_DVR\_SET\_ENISSUED\_DATADEL(清除下发的卡片数据)。
- 新增报警信息上传 (对应接口 [NET\\_DVR\\_StartListen\\_V30](#)): [NET\\_ITS\\_PASSVEHICLE\\_COST\\_ITEM](#)(出入口过车收费明细信息)、[NET\\_ITS\\_HANDOVER\\_INFO](#)(出入口交接班数据信息)。
- [NET\\_DVR\\_COMPRESSION\\_INFO\\_V30](#) 参数 byResolution 新增取值: 59-1600\*900。
- [NET\\_ITS\\_GATE\\_LANE\\_CFG](#) 使用 12 个保留字节新增参数: byCharge(是否收费)、byChargeMode(收费类型)、byLedRelativeIndex[8](LED 索引)、byGateRelativeIndex(出入口控制机相对索引)、byFarRrRelativeIndex(远距离读卡器相对索引)。
- [NET\\_ITS\\_GATE\\_VEHICLE](#) 使用 32 个保留字节新增参数: bySwipeTime[32](增加刷卡时间)。
- 新增操作日志次类型: 0x2011~0x2038, 0x210d~0x2110。
- 新增设备类型: TRAFFIC\_ECT、TRAFFIC\_PARKING\_SERVER。

### Version 4.2.5.6(build20130808)

- VAR-500X-X(P)未礼让行人。
- [NET\\_DVR\\_VEHICLE\\_INFO](#) 的参数 byIllegalType 新增取值：14-未礼让行人。
- [NET\\_ITS\\_PLATE\\_RESULT](#) 的参数 willegalType 违章类型增加国标定义说明，新增 1357（未礼让行人）。

### Version 4.2.5.6(build20130808)

- DS-TP5016-P(停车场管理系统)、DS-TP5008-E(出入口终端服务器)。
- 新增远程配置功能(对应接口 [NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)):  
内外置灯参数配置([NET\\_DVR\\_LAMP\\_CTRL\\_INFO](#))、特殊车位参数配置([NET\\_DVR\\_PARKSPACE\\_ATTRIBUTE](#))、出入口参数配置([NET\\_ITS\\_IPC\\_CHAN\\_LANE\\_CFG](#))。
- 新增远程设置功能(对应接口 [NET\\_DVR\\_SetDVRConfig](#)):  
设置指示灯外控参数([NET\\_DVR\\_LAMP\\_EXTERNAL\\_CFG](#))、设置车位强制抓图([NET\\_DVR\\_COMPEL\\_CAPTURE](#))、设置远程设备控制参数([NET\\_ITS\\_REMOTE\\_CONTROL](#))
- 新增报警信息上传(对应接口 [NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#) 和 [NET\\_DVR\\_StartListen\\_V30](#)): 指示灯外控报警信息([NET\\_DVR\\_EXTERNAL\\_CONTROL\\_ALARM](#))、出入口车辆抓拍数据([NET\\_ITS\\_GATE\\_VEHICLE](#))、出入口人脸抓拍数据([NET\\_ITS\\_GATE\\_FACE](#))、停车场数据上传([NET\\_ITS\\_PARK\\_VEHICLE](#))、车辆名单报警信息([NET\\_ITS\\_ECT\\_BLOCKLIST](#))。
- 新增出入口终端工作状态的功能(对应接口 [NET\\_DVR\\_GetDeviceStatus](#)): [NET\\_ITS\\_ECTWORKSTATE](#)
- 新增出入口终端通道状态的功能(对应接口 [NET\\_DVR\\_StartRemoteConfig](#) 和 [NET\\_DVR\\_GetNextRemoteConfig](#)): [NET\\_ITS\\_ECT\\_CHANNELSTATE](#)
- 新增操作日志次类型：0x2001~0x210c
- 前端参数配置（[NET\\_DVR\\_GET\\_CCDPARAMCFG](#)、[NET\\_DVR\\_SET\\_CCDPARAMCFG](#)、[NET\\_DVR\\_GET\\_CCDPARAMCFG\\_EX](#)、[NET\\_DVR\\_SET\\_CCDPARAMCFG\\_EX](#)）时，接口 [NET\\_DVR\\_GetDVRConfig](#)、[NET\\_DVR\\_SetDVRConfig](#) 中参数 IChannel 扩展为有效，表示通道号，IP 通道起始通道号为 33。
- 智能终端能力集 [DEVICE\\_ABILITY\\_INFO](#) 新增能力节点：<LampCtrlInfo>、<parkSpaceAttributeParam>、<lampExternalCfg>、<compelCaptureCfg>、<externalControlAlarm>。
- [NET\\_ITS\\_SINGLE\\_DEVICE\\_INFO](#) 的参数 dwDeviceType 新增取值：10- 道闸，11- 出入口控制机(票箱)，12- LED 显示屏，13- 远距离读卡器，14- 近距离读卡器，15- 红外扫描枪，16- 票据打印机，17- 节点管理器(诱导服务器相关)。

### Version 4.2.2.1 (build20130716)

- VAR-500X\_事件检测取证系统 v4.0.0。
- 新增监听报警上传类型（对应接口：[NET\\_DVR\\_StartListen\\_V30](#)）：  
交通取证报警信息([NET\\_DVR\\_TFS\\_ALARM](#))、(交通事件报警信息扩展)[NET\\_DVR\\_AID\\_ALARM\\_V41](#)。

### Version 4.2.1.3 (2012-12-10)

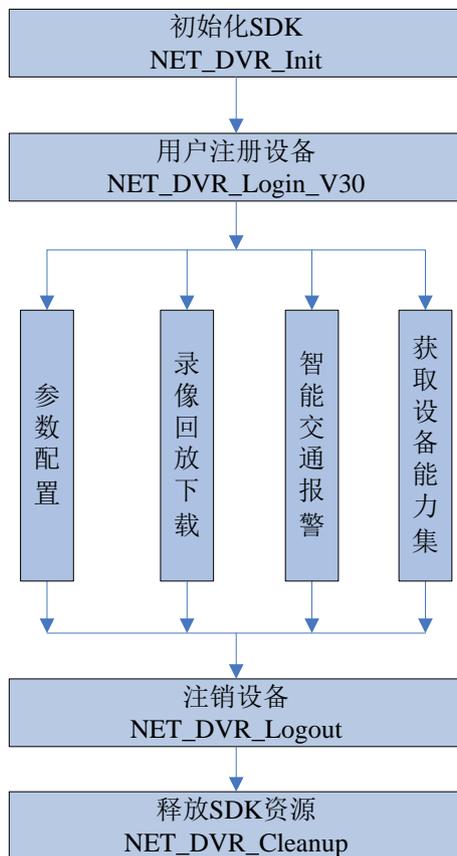
- SDK 新增支持 ITS 智能终端设备：TRAFFIC\_TS\_SERVER、TRAFFIC\_VAR
- 新增配置功能（对应接口：[NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)）：数据上传配置（[NET\\_ITS\\_UPLOAD\\_CFG](#)）、获取路口信息（[NET\\_ITS\\_ROADINFO](#)）
- 新增获取终端状态功能（对应接口：[NET\\_DVR\\_GetDeviceStatus](#)）：[NET\\_ITS\\_WORKSTATE](#)
- 新增监听报警上传类型（对应接口：[NET\\_DVR\\_StartListen\\_V30](#)）：  
[NET\\_ITS\\_PLATE\\_RESULT](#)、[NET\\_ITS\\_ROADINFO](#)
- 新增能力集（对应接口：[NET\\_DVR\\_GetDeviceAbility](#)）：  
[DEVICE\\_ABILITY\\_INFO](#)
- [NET\\_DVR\\_RECORDSCHED](#) 和 [NET\\_DVR\\_RECORDDAY](#) 参数 byRecordType 增加取值类型：14-智能交通事件，

仅 VAR 支持

## 3 函数调用顺序

### 3.1 SDK 接口调用主要流程

表 3.1 SDK 调用主要流程



智能交通功能模块包括参数配置、智能交通报警和获取设备能力集等功能。

- 智能交通报警，包括布防和监听两种方式。
  - 抓拍结果数据([NET ITS PLATE RESULT](#))、交通违章取证信息([NET DVR TFS ALARM](#))、交通事件报警信息([NET DVR AID ALARM V41](#))、交通统计数据([NET ITS TRAFFIC COLLECT](#))、出入口车辆抓拍数据([NET ITS GATE VEHICLE](#))、出入口人脸抓拍数据([NET ITS GATE FACE](#))、停车场数据([NET ITS PARK VEHICLE](#))、交通取证报警信息([NET DVR TFS ALARM](#))、交通参数统计报警信息([NET DVR TPS ALARM V41](#))、交通事件报警信息([NET DVR AID ALARM V41](#))等信息，只能通过报警“监听”的方式获取，智能交通设备上需要配置上传中心的 IP 地址和端口。接口调用详见【[报警监听模块流程](#)】。
  - 车辆名单报警([NET ITS ECT BLOCKLIST](#))、路口设备异常报警([NET ITS ROADINFO](#))、指示灯外控报警信息([NET DVR EXTERNAL CONTROL ALARM](#))，除了可以使用监听方式，也可以使用布防模式获取，布防具体接口调用详见【[报警布防模块流程](#)】。
- 参数配置：主要完成对智能交通设备基本参数的配置功能，包括终端基本信息获取、数据上传配置、出入口参数配置等，调用的接口为 [NET DVR GetDVRConfig](#) 和 [NET DVR SetDVRConfig](#)。终端服务器和视频监控记录分析仪前端接网络摄像机，其信息获取、IP 通道添加删除、修改等，也通过这两个接口实现，

其相关配置参数为 [NET\\_DVR\\_IPPARACFG\\_V40](#)。智能交通设备的参数配置一般建议通过 IE 登录设备直接通过控件界面进行相关配置。

- 录像回放下载：智能交通事件联动录像文件查找、回放以及下载，具体请参考回放下载模块流程。
- 获取设备能力集：通过 [NET\\_DVR\\_GetDeviceAbility](#) 接口获取智能交通设备能力信息。

## 3.2 报警模块流程

### 3.2.1 报警（布防）流程

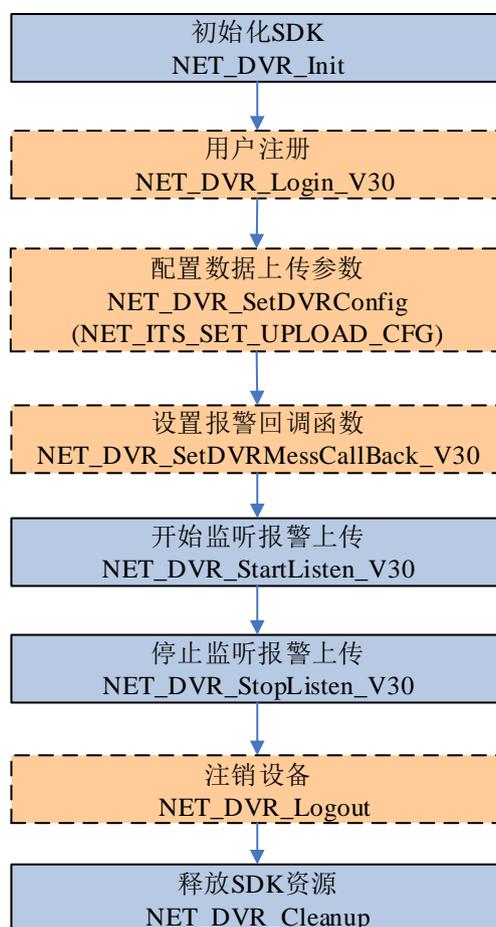
表 3.2 报警布防流程



- “布防”报警方式：是指 SDK 主动连接设备，并发起报警上传命令，设备发生报警立即发送给 SDK。
- 先登录设备（[NET\\_DVR\\_Login\\_V30](#)），设置报警回调函数（[NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#)），然后调用 [NET\\_DVR\\_SetupAlarmChan\\_V41](#) 进行布防，设备上传的相关信息即在设置的回调函数中获取。整个报警上传过程结束后还需要调用撤防接口等操作。

### 3.2.2 报警（监听）流程

表 3.3 报警监听流程



- “监听”报警方式：是指 SDK 不主动发起连接设备，只是在设定的端口上监听接收设备主动上传的报警信息。
- 客户端调用 [NET\\_DVR\\_StartListen\\_V30](#) 开启 SDK 的监听端口并且设置报警回调函数，准备接收设备上传的报警信息。设备上传相关信息即在设置的回调函数中获取。
  - 这个过程设备上需要先配置报警主机地址（即 PC 机地址）和报警主机端口（即 PC 的监听端口），报警主机就在该端口上监听接收设备主动上传的报警信息。如果报警主机地址和报警主机端口已配置完成，那么“报警（监听）的流程图”中虚线框“用户注册”和“配置报警主机地址和端口”部分就可以省略。如果事先没有配置，可以调用参数配置接口 [NET\\_DVR\\_GetDVRConfig](#)（命令：[NET\\_ITS\\_GET\\_UPLOAD\\_CFG](#)）和 [NET\\_DVR\\_SetDVRConfig](#)（命令：[NET\\_ITS\\_SET\\_UPLOAD\\_CFG](#)）获取和设置设备的数据上传参数（[NET\\_ITS\\_UPLOAD\\_CFG](#)），包括 IP 地址和端口、上传的数据类型、是否上传等。
- 该方式适用于多个设备主动向客户端上传报警，而且不需要设备登录即可完成，设备重启后不影响报警上传。智能终端服务器支持两个报警主机地址和端口号的配置。

[调用实例代码](#)

### 3.3 IP 通道配置

智能交通服务器实现 IPC 接入功能，需要调用 IP 接入配置参数来进行资源的获取和重新分配。

客户端通过注册设备 ([NET\\_DVR\\_Login\\_V30](#)) 返回的设备信息 [NET\\_DVR\\_DEVICEINFO\\_V30](#) 可以获取到模拟通道数 (byChanNum)、模拟通道起始通道号 (byStartChan) 和设备支持的最大 IP 通道数 (byIPChanNum + byHighDChanNum\*256)、数字通道起始通道号 (byStartDChan)。一般情况下，智能交通设备的模拟通道从 1 开始，IP 通道号从 33 开始。

从 byStartChan 到 byStartChan+byChanNum-1 对应为模拟通道的通道号。IP 通道号为 byStartDChan 到 byStartDChan+ (byIPChanNum + byHighDChanNum\*256) -1。

如果设备支持 IP 通道 (byIPChanNum>0)，则通过远程参数配置接口 [NET\\_DVR\\_GetDVRConfig](#) (配置命令 [NET\\_DVR\\_GET\\_IPPARACFG\\_V40](#)) 可以获取得到设备详细的 IP 通道信息 ([NET\\_DVR\\_IPPARACFG\\_V40](#))，包括模拟通道是否禁用、IP 通道个数 (dwDChanNum)、IP 通道起始通道号 (dwStartDChan)、IP 通道取流模式、IP 通道有效状态和在线状态等。通过远程参数配置接口 [NET\\_DVR\\_SetDVRConfig](#) (配置命令 [NET\\_DVR\\_SET\\_IPPARACFG\\_V40](#)) 可对设备进行 IP 资源配置，包括添加、修改、删除 IP 通道等。

## 4 函数调用实例

### 4.1 交通数据接收示例代码

[相关模块流程图](#)

```

#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
using namespace std;

typedef HWND (WINAPI *PROCGETCONSOLEWINDOW)();
PROCGETCONSOLEWINDOW GetConsoleWindow;

int iNum=0;
void CALLBACK MSesGCallback(LONG ICommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD dwBufLen, void*
pUser)
{
    int i=0;
    char filename[100];
    FILE *fSnapPic=NULL;
    FILE *fSnapPicPlate=NULL;

    //以下代码仅供参考，实际应用中不建议在该回调函数中直接处理数据保存文件，
    //例如可以使用消息的方式(PostMessage)在消息响应函数里进行处理。
    //仅处理交通数据上传

    switch (ICommand)
    {
        case COMM_ITS_PLATE_RESULT: //交通抓拍的终端图片
        {
            NET_ITS_PLATE_RESULT struITSPlateResult={0};
            memcpy(&struITSPlateResult, pAlarmInfo, sizeof(struITSPlateResult));

            for (i=0;i<struITSPlateResult.dwPicNum;i++)
            {
                printf("车牌号: %s\n", struITSPlateResult.struPlateInfo.sLicense);//车牌号
                switch(struITSPlateResult.struPlateInfo.byColor)//车牌颜色
                {
                    case VCA_BLUE_PLATE:
                        printf("车辆颜色: 蓝色\n");
                        break;
                    case VCA_YELLOW_PLATE:

```

```

        printf("车辆颜色: 黄色\n");
        break;
    case VCA_WHITE_PLATE:
        printf("车辆颜色: 白色\n");
        break;
    case VCA_BLACK_PLATE:
        printf("车辆颜色: 黑色\n");
        break;
    default:
        break;
}
//保存场景图
//struTSPlateResult.struPicInfo[i].byTyp: 数据类型, 0-车牌图、1- 场景图、2- 合成图、3- 特写图
if ((struTSPlateResult.struPicInfo[i].dwDataLen != 0)&&(struTSPlateResult.struPicInfo[i].byType==
1) || (struTSPlateResult.struPicInfo[i].byType == 2))
{
    sprintf(filename,"testITSpic%d_%d.jpg",iNum,i);
    fSnapPic=fopen(filename,"wb");
    fwrite(struTSPlateResult.struPicInfo[i].pBuffer, struTSPlateResult.struPicInfo[i].dwDataLen,1,fSnapPic);
    iNum++;
    fclose(fSnapPic);
}
//车牌小图片
if ((struTSPlateResult.struPicInfo[i].dwDataLen != 0)&&(struTSPlateResult.struPicInfo[i].byType == 0))
{
    sprintf(filename,"testPicPlate%d_%d.jpg",iNum,i);
    fSnapPicPlate=fopen(filename,"wb");
    fwrite(struTSPlateResult.struPicInfo[i].pBuffer, struTSPlateResult.struPicInfo[i].dwDataLen, 1, \
fSnapPicPlate);

    iNum++;
    fclose(fSnapPicPlate);
}
//其他信息处理.....
}
break;
}
case COMM_ITS_GATE_VEHICLE: //出入口车辆抓拍数据
{
    NET_ITS_GATE_VEHICLE struITSGateVehicle={0};
    memcpy(&struITSGateVehicle, pAlarmInfo, sizeof(struITSGateVehicle));
    //信息处理.....
    break;
}
case COMM_ITS_PARK_VEHICLE: //停车场数据

```

```

    {
        NET_ITS_PARK_VEHICLE struITSParkVehicle={0};
        memcpy(&struITSParkVehicle, pAlarmInfo, sizeof(struITSParkVehicle));
        //信息处理.....
        break;
    }
case COMM_ALARM_TFS: //交通取证报警信息
    {
        NET_DVR_TFS_ALARM struTFSAlarm={0};
        memcpy(&struTFSAlarm, pAlarmInfo, sizeof(struTFSAlarm));
        //信息处理.....
        break;
    }
case COMM_ALARM_TPS_V41: //交通参数统计报警信息
    {
        NET_DVR_TPS_ALARM_V41 struTPSAlarmV41={0};
        memcpy(&struTPSAlarmV41, pAlarmInfo, sizeof(struTPSAlarmV41));
        //信息处理.....
        break;
    }
    //.....其他类型报警信息处理
default:
    break;
}
return;
}
void main()
{
    //-----
    //初始化
    NET_DVR_Init();

    //获取控制台窗口句柄
    HMODULE hKernel32 = GetModuleHandle("kernel32");
    GetConsoleWindow = (PROCGETCONSOLEWINDOW)GetProcAddress(hKernel32,"GetConsoleWindow");

    //设置连接时间与重连时间
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    //注册设备(监听报警可以不注册)
    LONG IUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;

```

```

IUserID = NET_DVR_Login_V30("172.6.24.119", 8000, "admin", "12345", &struDeviceInfo);
if (IUserID < 0)
{
    printf("Login error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

WORD wLocalPort=5650; //监听端口为 5650
LONG IHandle = NET_DVR_StartListen_V30(NULL, wLocalPort, MSesGCallback, NULL);
//监听 IP 为客户端本机 IP, 并设置报警回调函数

if (IHandle < 0)
{
    printf("NET_DVR_StartListen_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}
//-----
//启动预览(如果只是监听报警, 预览不需要调用)
HWND hWnd = GetConsoleWindow(); //获取窗口句柄
NET_DVR_PREVIEWINFO struPlayInfo = {0};
struPlayInfo.hPlayWnd = hWnd; //需要 SDK 解码时句柄设为有效值, 仅取流不解码时可设为空
struPlayInfo.IChannel = 1; //预览通道号
struPlayInfo.dwStreamType = 0; //0-主码流, 1-子码流, 2-码流 3, 3-码流 4, 以此类推
struPlayInfo.dwLinkMode = 0; //0- TCP 方式, 1- UDP 方式, 2- 多播方式, 3- RTP 方式, 4-RTP/RTSP, 5-RSTP/HTTP

LONG IRealPlayHandle = NET_DVR_RealPlay_V40(IUserID, &struPlayInfo, NULL, NULL); //启动预览
if (IRealPlayHandle < 0)
{
    printf("NET_DVR_RealPlay_V40 error\n");
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}
Sleep(20000); //测试, 等待 20 秒

//关闭预览
NET_DVR_StopRealPlay(IRealPlayHandle);
printf("\n");
printf("\n");
printf("\n");

```

```
//停止监听
if (!NET_DVR_StopListen_V30(IHandle))
{
    printf("NET_DVR_StopListen_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}
//注销用户
NET_DVR_Logout(IUserID);
//释放 SDK 资源
NET_DVR_Cleanup();
return;
}
```

## 5 函数说明

### 5.1 SDK 初始化

#### 5.1.1 初始化 SDK `NET_DVR_Init`

函数: `BOOL NET_DVR_Init()`  
参数: 无  
返回值: `TRUE` 表示成功, `FALSE` 表示失败。  
说明: 调用设备网络 SDK 其他函数的前提。

[返回目录](#)

#### 5.1.2 释放 SDK 资源 `NET_DVR_Cleanup`

函数: `BOOL NET_DVR_Cleanup()`  
参数: 无  
返回值: `TRUE` 表示成功, `FALSE` 表示失败。  
说明: 在结束之前最后调用。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

[返回目录](#)

### 5.2 SDK 本地功能

#### SDK 本地参数配置

#### 5.2.1 获取 SDK 本地参数 `NET_DVR_GetSDKLocalCfg`

函数: `BOOL NET_DVR_GetSDKLocalCfg(NET_SDK_LOCAL_CFG_TYPE enumType, void *lpOutBuff)`  
参数: `[in] enumType` 配置类型, 不同的取值对应不同的 SDK 参数, 详见表 5.1  
`[out] lpOutBuff` 输出参数, 不同的配置类型, 输出参数对应不同的结构, 详见表 5.1  
返回值: `TRUE` 表示成功, `FALSE` 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。  
说明:

表 5.1 本地参数类型

enumType 宏定义	类型值	含义	IpOutBuff 对应结构体
NET_SDK_LOCAL_CFG_TYPE_TCP_PORT_BIND	0	本地 TCP 端口绑定配置	<a href="#">NET_DVR_LOCAL_TCP_PORT_BIND_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_UDP_PORT_BIND	1	本地 UDP 端口绑定配置	<a href="#">NET_DVR_LOCAL_UDP_PORT_BIND_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_MEM_POOL	2	内存池本地配置	<a href="#">NET_DVR_LOCAL_MEM_POOL_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_MODULE_RECV_TIMEOUT	3	按模块配置超时时间	<a href="#">NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_ABILITY_PARSE	4	是否使用能力集解析库	<a href="#">NET_DVR_LOCAL_ABILITY_PARSE_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_TALK_MODE	5	对讲模式配置	<a href="#">NET_DVR_LOCAL_TALK_MODE_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_CHECK_DEV	10	心跳交互间隔时间配置	<a href="#">NET_DVR_LOCAL_CHECK_DEV</a>

[返回目录](#)

## 5.2.2 设置 SDK 本地参数 **NET\_DVR\_SetSDKLocalCfg**

函数: `BOOL NET_DVR_SetSDKLocalCfg(NET_SDK_LOCAL_CFG_TYPE enumType, void* const lpInBuff)`

参数: [in] enumType 配置类型, 不同的取值对应不同的 SDK 参数, 详见表 5.2  
[in] lpInBuff 输入参数, 不同的配置类型, 输出参数对应不同的结构, 详见表 5.2

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

表 5.2 本地参数类型

enumType 宏定义	类型值	含义	lpInBuff 对应结构体
NET_SDK_LOCAL_CFG_TYPE_TCP_PORT_BIND	0	本地 TCP 端口绑定配置	<a href="#">NET_DVR_LOCAL_TCP_PORT_BIND_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_UDP_PORT_BIND	1	本地 UDP 端口绑定配置	<a href="#">NET_DVR_LOCAL_UDP_PORT_BIND_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_MEM_POOL	2	内存池本地配置	<a href="#">NET_DVR_LOCAL_MEM_POOL_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_MODULE_RECV_TIMEOUT	3	按模块配置超时时间	<a href="#">NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_ABILITY_PARSE	4	是否使用能力集解析库	<a href="#">NET_DVR_LOCAL_ABILITY_PARSE_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_TALK_MODE	5	对讲模式配置	<a href="#">NET_DVR_LOCAL_TALK_MODE_CFG</a>
NET_SDK_LOCAL_CFG_TYPE_CHECK_DEV	10	心跳交互间隔时间配置	<a href="#">NET_DVR_LOCAL_CHECK_DEV</a>
NET_SDK_LOCAL_CFG_TYPE_CHAR_ENCODE	13	配置字符编码相关处理回调	<a href="#">NET_DVR_LOCAL_BYTE_ENCODE_CONVERT</a>
NET_DVR_LOCAL_CFG_TYPE_LOG	15	日志参数配置	<a href="#">NET_DVR_LOCAL_LOG_CFG</a>

[返回目录](#)

## 连接和接收超时时间及重连设置

### 5.2.3 设置网络连接超时时间和连接尝试次数 **NET\_DVR\_SetConnectTime**

函数: BOOL NET\_DVR\_SetConnectTime(DWORD dwWaitTime,DWORD dwTryTime)

参数: [in]dwWaitTime 超时时间, 单位毫秒, 取值范围[300,75000], 实际最大超时时间因系统的 connect 超时时间而不同。  
[in]dwTryTimes 连接尝试次数 (保留)

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: **SDK 默认建立连接的超时时间为 3 秒。** SDK4.0 及以后版本中当设置的超时时间超过或低于限制的值时接口不返回失败, 将取最接近的上下限限制值作为实际的超时时间。

[返回目录](#)

### 5.2.4 设置重连功能 **NET\_DVR\_SetReconnect**

函数: BOOL NET\_DVR\_SetReconnect (DWORD dwInterval,BOOL bEnableRecon)

参数: [in]dwInterval 重连间隔, 单位:毫秒  
[in]bEnableRecon 是否重连, 0-不重连, 1-重连, 参数默认为 1

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 该接口可以同时控制预览、透明通道和布防的重连功能。不调用该接口时, SDK 默认启动预览、透明通道和布防的重连功能, 重连时间间隔为 5 秒。

[返回目录](#)

### 5.2.5 设置接收超时时间 **NET\_DVR\_SetRecvTimeOut**

函数: BOOL NET\_DVR\_SetRecvTimeOut(DWORD nRecvTimeOut)

参数: [in] nRecvTimeOut 接收超时时间, 单位毫秒, 默认为 5000, 最小为 3000 毫秒

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 该接口用于设置接收超时时间, 例如预览接收实时流数据、回放下载接收录像数据、报警接收报警信息等接收超时时间。

[返回目录](#)

## 多网卡绑定

### 5.2.6 获取所有 IP, 用于支持多网卡接口 **NET\_DVR\_GetLocalIP**

函数: BOOL NET\_DVR\_GetLocalIP(char strIP[16][16], DWORD \*pValidNum, BOOL \*pEnableBind)

参数: [out] strIP 存放 IP 的缓冲区, 不能为空

[out] pValidNum 所有有效 IP 的数量

[out] pEnableBind 是否绑定

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

**说明:** 该接口获取客户端本地多网卡的所有 IP 地址, 可以通过接口 NET\_DVR\_SetValidIP 选择要使用的 IP 地址。

[返回目录](#)

## 5.2.7 设置 IP 绑定 [NET\\_DVR\\_SetValidIP](#)

函数: BOOL NET\_DVR\_SetValidIP(DWORD dwIPIndex, BOOL bEnableBind)

参数: [in] dwIPIndex 选择使用的 IP 下标, 由 NET\_DVR\_GetLocalIP 获取

[in] bEnableBind 是否绑定

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

**说明:**

[返回目录](#)

## SDK 版本、状态和能力

### 5.2.8 获取 SDK 的版本号和 build 信息 [NET\\_DVR\\_GetSDKBuildVersion](#)

函数: DWORD NET\_DVR\_GetSDKBuildVersion()

参数:

返回值: 获取 SDK 的版本号和 build 信息。

**说明:** SDK 的版本号和 build 信息。2 个高字节表示版本号: 25~32 位表示主版本号, 17~24 位表示次版本号; 2 个低字节表示 build 信息。如 0x03000101: 表示版本号为 3.0, build 号是 0101。

[返回目录](#)

### 5.2.9 获取当前 SDK 的状态信息 [NET\\_DVR\\_GetSDKState](#)

函数: BOOL NET\_DVR\_GetSDKState(LPNET\_DVR\_SDKSTATE pSDKState);

参数: [out] pSDKState SDK 状态信息, 详见结构体: [NET\\_DVR\\_SDKSTATE](#)

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

**说明:**

[返回目录](#)

### 5.2.10 获取当前 SDK 的功能信息 [NET\\_DVR\\_GetSDKAbility](#)

函数: BOOL NET\_DVR\_GetSDKAbility(LPNET\_DVR\_SDKABL pSDKAbl)

参数: [out] pSDKAbl SDK 功能信息, 详见结构体: [NET\\_DVR\\_SDKABL](#)

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

## SDK 启用写日志

### 5.2.11 启用写日志文件 **NET\_DVR\_SetLogToFile**

函数: BOOL NET\_DVR\_SetLogToFile(DWORD bLogEnable, char\* strLogDir, BOOL bAutoDel)

参数: [in]bLogEnable 日志的等级 (默认为 0):  
 0-表示关闭日志  
 1-表示只输出 ERROR 错误日志  
 2-输出 ERROR 错误信息和 DEBUG 调试信息  
 3-输出 ERROR 错误信息、DEBUG 调试信息和 INFO 普通信息等所有信息

[in]strLogDir 日志文件的路径, windows 默认值为"C:\\SdkLog\\"; linux 默认值"/home/sdklog/"

[in]bAutoDel 是否删除超出的文件数, 默认值为 TRUE

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

- 日志文件路径必须是绝对路径, 且以"\"结尾, 例如"C:\\SdkLog\\", 建议用户先手动创建文件。若未指定文件路径, 则采用默认路径"C:\\SdkLog\\"。
- 可多次调用该接口创建新的日志文件, 更改目录时到下一次写文件时才会使用新的目录写文件。
- bAutoDel 为 TRUE 时表示覆盖模式, 日志文件个数超过 SDK 限制个数时将会自动删除超出的文件。SDK 限制个数默认为 10 个, 可以调用接口 [NET\\_DVR\\_SetSDKLocalCfg](#)(配置类型: NET\_DVR\_LOCAL\_CFG\_TYPE\_LOG)进行修改配置。

[返回目录](#)

## 异常消息回调

### 5.2.12 注册接收异常、重连等消息的窗口句柄或回调函数

#### **NET\_DVR\_SetExceptionCallBack\_V30**

函数: [Windows 系统下:](#)

BOOL NET\_DVR\_SetExceptionCallBack\_V30 (UINT nMessage, HWND hWnd, fExceptionCallBack cbExceptionCallBack, void\* pUser)

[Linux 系统下:](#)

BOOL NET\_DVR\_SetExceptionCallBack\_V30(UINT nMessage, void\* hWnd, fExceptionCallBack

cbExceptionCallBack,void\* pUser)

参 数: [in]nMessage 消息,Linux 下该参数保留  
[in]hWnd 接收异常消息的窗口句柄, Linux 下该参数保留  
[in]cbExceptionCallBack 接收异常消息的回调函数, 回调当前异常的相关信息  
[in]pUser 用户数据

```
typedef void(CALLBACK* fExceptionCallBack)(DWORD dwType, LONG IUserID, LONG IHandle, void *pUser)
```

[out]dwType 异常或重连等消息的类型, 详见表 5.3  
[out]IUserID 登录 ID  
[out]IHandle 出现异常的相应类型的句柄  
[out]pUser 用户数据

表 5.3 异常消息类型

dwType 宏定义	宏定义值	含义
EXCEPTION_EXCHANGE	0x8000	用户交互时异常 (注册心跳超时, 心跳间隔为 2 分钟)
EXCEPTION_AUDIOEXCHANGE	0x8001	语音对讲异常
EXCEPTION_ALARM	0x8002	报警异常
EXCEPTION_PREVIEW	0x8003	网络预览异常
EXCEPTION_SERIAL	0x8004	透明通道异常
EXCEPTION_RECONNECT	0x8005	预览时重连
EXCEPTION_ALARMRECONNECT	0x8006	报警时重连
EXCEPTION_SERIALRECONNECT	0x8007	透明通道重连
SERIAL_RECONNECTSUCCESS	0x8008	透明通道重连成功
EXCEPTION_PLAYBACK	0x8010	回放异常
EXCEPTION_DISKFMT	0x8011	硬盘格式化
EXCEPTION_EMAILTEST	0x8013	邮件测试异常
EXCEPTION_BACKUP	0x8014	备份异常
PREVIEW_RECONNECTSUCCESS	0x8015	预览时重连成功
ALARM_RECONNECTSUCCESS	0x8016	报警时重连成功
RESUME_EXCHANGE	0x8017	用户交互恢复

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明: Windows 下该函数的 hWnd 和 cbExceptionCallBack 不能同时为 NULL, Linux 下 cbExceptionCallBack 不能设置为 NULL, 否则将接收不到异常消息。

如果此结构是以回调方式反馈异常消息, 那么应用程序中的异常回调函数实现如下, 该函数中的参数 dwType 表示异常消息类型 (见上表); IHandle 表示发生异常的相应类型的句柄。

#### 示例代码:

```
//注册接收异常消息的回调函数
NET_DVR_SetExceptionCallBack_V30(WM_NULL, NULL, g_ExceptionCallBack, NULL);
//接收异常消息的回调函数的外部实现
void CALLBACK g_ExceptionCallBack(DWORD dwType, LONG IUserID, LONG IHandle, void *pUser)
```

```

{
    char tempbuf[256];
    ZeroMemory(tempbuf,256);
    switch(dwType)
    {
    case EXCEPTION_AUDIOEXCHANGE:           //语音对讲时网络异常
        sprintf(tempbuf,"语音对讲时网络异常!!!");
        TRACE("%s",tempbuf);
        //TODO: 关闭语音对讲
        break;
    case EXCEPTION_ALARM:                   //报警上传时网络异常
        sprintf(tempbuf,"报警上传时网络异常!!!");
        TRACE("%s",tempbuf);
        //TODO: 关闭报警上传
        break;
    case EXCEPTION_PREVIEW:                 //网络预览时异常
        sprintf(tempbuf,"网络预览时网络异常!!!");
        TRACE("%s",tempbuf);
        //TODO: 关闭网络预览
        break;
    case EXCEPTION_SERIAL:                   //透明通道传输时异常
        sprintf(tempbuf,"透明通道传输时网络异常!!!");
        TRACE("%s",tempbuf);
        //TODO: 关闭透明通道
        break;
    case EXCEPTION_RECONNECT:               //预览时重连
        break;
    default:
        break;
    }
};

```

[返回目录](#)

## 获取错误信息

### 5.2.13 返回最后操作的错误码 **NET\_DVR\_GetLastError**

函数: `DWORD NET_DVR_GetLastError()`

参数:

返回值: 返回最后操作的错误码。详见[错误码宏定义](#)

说明: 返回值为错误码。错误码主要分为网络通讯库错误码、RTSP 通讯库错误码和软硬解库错误码。

[返回目录](#)

### 5.2.14 返回最后操作的错误码信息 **NET\_DVR\_GetErrorMsg**

函数: char\* NET\_DVR\_GetErrorMsg(LONG \*pErrorNo)  
 参数: [out]pErrorNo 错误码数值的指针  
 返回值: 返回值为错误码信息的指针。错误码主要分为网络通讯库错误码、RTSP 通讯库错误码和软硬解库错误码。详见[错误码宏定义](#)  
 说明:

[返回目录](#)

## 5.3 用户注册

### 5.3.1 激活设备 **NET\_DVR\_ActivateDevice**

函数: BOOL NET\_DVR\_ActivateDevice(char\* sDVRIP, WORD wDVRPort, LPNET\_DVR\_ACTIVATECFG lpActivateCfg)  
 参数: [in]sDVRIP 设备 IP 地址  
 [in]wDVRPort 设备端口  
 [in]lpActivateCfg 激活参数, 包括激活使用的初始密码, 详见结构体:  
[NET\\_DVR\\_ACTIVATECFG](#)  
 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。  
 说明: 出厂设备需要先激活, 然后再使用激活使用的初始密码登录设备。

[返回目录](#)

### 5.3.2 通过解析服务器, 获取设备的动态 IP 地址和端口号

#### **NET\_DVR\_GetDVRIPByResolveSvr\_EX**

函数: BOOL NET\_DVR\_GetDVRIPByResolveSvr\_EX (char\* sServerIP, WORD wServerPort, BYTE\* sDVRName, WORD wDVRNameLen, BYTE\* sDVRSerialNumber, WORD wDVRSerialLen, char\* sGetIP, DWORD\* dwPort)  
 参数: [in]sServerIP 解析服务器的 IP 地址  
 [in]wServerPort 解析服务器的端口号, IP Server 解析服务器端口号为 7071, HiDDNS 服务器的端口号为 80  
 [in]sDVRName 设备名称  
 [in]wDVRNameLen 设备名称的长度  
 [in]sDVRSerialNumber 设备的序列号  
 [in]wDVRSerialLen 设备序列号的长度  
 [out]sGetIP 获取到的设备 IP 地址指针  
 [out]dwPort 获取到的设备端口号指针  
 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过

错误码判断出错原因。

**说明：** 该接口中的设备名称和设备序列号不能同时为空。通过设备域名或者序列号解析出设备当前 IP 地址和端口，然后调用 [NET\\_DVR\\_Login\\_V40](#) 登录设备。  
支持的解析服务器有 IPServer 和 hiDDNS。

[返回目录](#)

### 5.3.3 用户注册设备 **NET\_DVR\_Login\_V40**

**函数：** LONG NET\_DVR\_Login\_V40(LPNET\_DVR\_USER\_LOGIN\_INFO pLoginInfo, LPNET\_DVR\_DEVICEINFO\_V40 lpDeviceInfo)

**参数：** [in]pLoginInfo 登录参数，包括设备地址、登录用户、密码等，详见结构体：  
[NET\\_DVR\\_USER\\_LOGIN\\_INFO](#)  
[out]lpDeviceInfo 设备信息(同步登录即 pLoginInfo 中 bUseAsynLogin 为 0 时有效)，  
详见结构体：[NET\\_DVR\\_DEVICEINFO\\_V40](#)

**返回值：** 异步登录的状态、用户 ID 和设备信息通过 NET\_DVR\_USER\_LOGIN\_INFO 结构体中设置的回调函数(fLoginResultCallBack)返回。对于同步登录，接口返回-1 表示登录失败，其他值表示返回的用户 ID 值。用户 ID 具有唯一性，后续对设备的操作都需要通过此 ID 实现。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

**说明：**

- pLoginInfo 中 bUseAsynLogin 为 0 时登录为同步模式，接口返回成功即表示登录成功；pLoginInfo 中 bUseAsynLogin 为 1 时登录为异步模式，登录是否成功在输入参数设置的回调函数中返回。
- 设备同时最多允许 128 个用户注册。
- SDK 支持 2048 个注册，返回 UserID 的取值范围为 0~2047。

[返回目录](#)

### 5.3.4 用户注销 **NET\_DVR\_Logout**

**函数：** BOOL NET\_DVR\_Logout(LONG IUserID)

**参数：** [in]IUserID 用户 ID 号，NET\_DVR\_Login\_V40 的返回值

**返回值：** TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

**说明：**

[返回目录](#)

## 5.4 获取设备能力集

### 5.4.1 获取设备能力集 **NET\_DVR\_GetDeviceAbility**

**函数：** BOOL NET\_DVR\_GetDeviceAbility(LONG IUserID, DWORD dwAbilityType, char\* pInBuf, DWORD dwInLength, char\* pOutBuf, DWORD dwOutLength)

**参数：** [in] IUserID NET\_DVR\_Login\_V30 的返回值  
[in] dwAbilityType 能力类型，具体定义见表 5.4

[in] pInBuf	输入缓冲区指针
[in] dwInLength	输入缓冲区的长度
[out] pOutBuf	输出缓冲区指针
[in] dwOutLength	接收数据的缓冲区的长度

表 5.4 设备能力集类型

宏定义	宏定义值	含义
DEVICE_SOFTWARE_ABILITY	0x001	设备软硬件能力
DEVICE_ABILITY_INFO	0x011	设备通用能力类型，具体能力根据发送的能力节点来区分

返回值： TRUE 表示成功， FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说明： 接口中 pInBuf 参数的具体定义格式按照不同的设备规定有所不同，需要输入参数和输出参数的格式定义如表 5.5 所示。

表 5.5 设备能力集描述

能力类型宏定义	能力类型说明	pInBuf	pOutBuf
DEVICE_SOFTWARE_ABILITY	获取设备软硬件能力	无	<a href="#">设备软硬件能力 XML 描述</a>
DEVICE_ABILITY_INFO	设备通用能力类型，具体能力根据发送的能力节点来区分	<a href="#">获取设备能力的输入描述</a>	<a href="#">智能终端能力集描述(ITDeviceAbility)</a>
		<a href="#">获取智能通道分析能力集</a>	<a href="#">智能通道分析能力 XML 描述 (VcaChanAbility)</a>
		<a href="#">车辆二次识别检测能力输入描述</a>	<a href="#">车辆二次识别检测能力 (VehicleRecogAbility)</a>

[返回目录](#)

## 5.5 实时预览

### 5.5.1 设置播放显示模式 NET\_DVR\_SetShowMode

函数： BOOL NET\_DVR\_SetShowMode (DWORD dwShowType, COLORREF colorKey)

参数： [in] dwShowType 显示模式

```
enum{
    NORMALMODE = 0,
    OVERLAYMODE
}
```

[in] colorKey 用户设置的透明色，在 OVERLAY 模式下需要设置，colorKey 是一个 32 位的值 0x00bbgrr，最高字节为 0，后三个字节分别表示 b、g、r 的值

返回值： TRUE 表示成功， FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说明：
 

- 设置播放器显示模式，需在预览之前设置。
- NORMALMODE 模式可以同时显示多窗口，但是对显卡有一定的要求；OVERLAYMODE 模式只能同时显示一个窗口，但是对显卡基本没有要求。透明色相当于一层透视膜，显示的画面只

能穿过这种颜色，而其他的颜色将挡住显示的画面，用户应该在显示窗口中涂上这种颜色才能看到显示画面，一般应该使用一种不常用的颜色作为透明色，colorKey 是一个 32 位的值 0x00bbgrr，最高字节为 0，后三个字节分别表示 b、g、r 的值。

- ▶ 播放器有两种显示模式：普通模式和 OVERLAY 方式，使用 OVERLAY 模式的优点是：大部分显卡都支持 OVERLAY，在一些不支持 BLT 硬件缩放和颜色转换的显卡上(如 SIS 系列显卡)使用 OVERLAY 模式，可以大大降低 CPU 利用率并提高画面质量(相对于软件实现缩放、颜色转换)。缺点是：同时只能播放一路图象，不能实现大规模集中监控。在一块显卡中同一时刻只能有一个 OVERLAY 表面处于活动状态，如果此时系统中已经有程序使用了 OVERLAY，那么播放器就不能再创建 OVERLAY 表面，它将自动改成普通的模式，并不返回 FALSE，一些常用的播放器，例如我们卡的预览都可能使用了 OVERLAY 表面，同样，如果我们的 SDK 中使用了 OVERLAY 表面，那么其他的程序将不能再使用 OVERLAY 表面。

[返回目录](#)

### 5.5.2 主码流动态产生一个关键帧 NET\_DVR\_MakeKeyFrame

函数： BOOL NET\_DVR\_MakeKeyFrame( LONG IUserID, DWORD IChannel)

参数： [in] IUserID                      NET\_DVR\_Login\_V30 的返回值  
[in] IChannel                      通道号

返回值： TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说明： 此接口用于重置 I 帧，根据设置的预览参数([NET\\_DVR\\_PREVIEWINFO](#))为主码流或者子码流分别调用 NET\_DVR\_MakeKeyFrame 或者 [NET\\_DVR\\_MakeKeyFrameSub](#) 实现重置 I 帧。

[返回目录](#)

### 5.5.3 子码流动态产生一个关键帧 NET\_DVR\_MakeKeyFrameSub

函数： BOOL NET\_DVR\_MakeKeyFrameSub( LONG IUserID, DWORD IChannel)

参数： [in] IUserID                      NET\_DVR\_Login\_V30 的返回值  
[in] IChannel                      通道号

返回值： TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说明： 此接口用于重置 I 帧，根据设置的预览参数([NET\\_DVR\\_PREVIEWINFO](#))为主码流或者子码流分别调用 [NET\\_DVR\\_MakeKeyFrame](#) 或者该接口实现重置 I 帧。

[返回目录](#)

### 5.5.4 实时预览 NET\_DVR\_RealPlay\_V40

函数： LONG NET\_DVR\_RealPlay\_V40(LONG IUserID, LPNET\_DVR\_PREVIEWINFO lpPreviewInfo, REALDATACALLBACK fRealDataCallBack\_V30, void \*pUser)

参数： [in] IUserID                      NET\_DVR\_Login\_V30 的返回值  
[in] lpPreviewInfo              预览参数，详见结构体：[NET\\_DVR\\_PREVIEWINFO](#)  
[in] fRealDataCallBack\_V30      码流数据回调函数

[in] pUser	用户数据
<code>typedef void(CALLBACK *REALDATACALLBACK)(LONG IRealHandle, DWORD dwDataType, BYTE *pBuffer, DWORD dwBufSize, void *pUser)</code>	
[out] IRealHandle	当前的预览句柄
[out] dwDataType	数据类型, 详见表 5.6
[out] pBuffer	存放数据的缓冲区指针
[out] dwBufSize	缓冲区大小
[out] pUser	用户数据

表 5.6 码流数据类型

dwDataType 宏定义	宏定义值	含义
NET_DVR_SYSHEAD	1	系统头数据
NET_DVR_STREAMDATA	2	流数据 (包括复合流或音视频分开的视频流数据)

返回值: -1 表示失败, 其他值作为 NET\_DVR\_StopRealPlay 等函数的句柄参数。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

**说明:** 该接口中的回调函数可以置为空, 这样该函数将不回调码流数据给用户, 不过用户仍可以通过接口 [NET\\_DVR\\_SetRealDataCallBack](#) 或 [NET\\_DVR\\_SetStandardDataCallBack](#) 注册捕获码流数据的回调函数以捕获码流数据。

客户端异常离线时, 设备端对取流连接的保持时间为 10 秒。

[返回目录](#)

### 5.5.5 停止预览 **NET\_DVR\_StopRealPlay**

函数: BOOL NET\_DVR\_StopRealPlay (LONG IRealHandle)

参数: [in] IRealHandle 预览句柄, NET\_DVR\_RealPlay\_V40 的返回值

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

**说明:**

[返回目录](#)

### 5.5.6 获取预览时用来解码和显示的播放库句柄 **NET\_DVR\_GetRealPlayerIndex**

函数: int NET\_DVR\_GetRealPlayerIndex(LONG IRealHandle)

参数: [in] IRealHandle 预览句柄, NET\_DVR\_RealPlay\_V40 的返回值

返回值: -1 表示失败, 其他值表示播放句柄。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

**说明:** 用户可以通过返回的句柄自行实现播放库 SDK 提供的其他功能, 详见本公司提供的软解码库函数说明《播放器 SDK 编程指南》。例如使用 `PlayM4_GetBMP(LONG nPort,.....)`、`PlayM4_GetJPEG(LONG nPort,.....)` 这两个接口时, 即可实现将当前预览图像以 BMP 或 JPEG 格式抓图保存到内存中: `PlayM4_GetBMP(NET_DVR_GetRealPlayerIndex(),.....)`  
`PlayM4_GetJPEG(NET_DVR_GetRealPlayerIndex(),.....)`

[返回目录](#)

## 5.6 预览时解码效果的参数控制

### 5.6.1 设置播放库的帧缓冲区个数 **NET\_DVR\_SetPlayerBufNumber**

函数: BOOL NET\_DVR\_SetPlayerBufNumber(LONG IRealHandle,DWORD dwBufNum)

参数: [in] IRealHandle NET\_DVR\_RealPlay\_V40 的返回值  
[in] dwBufNum 所要设置的单视频播放时缓冲区最大的帧数, 取值范围[1,50], SDK 默认的帧缓冲区大小为 15

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 设置网络延时和播放流畅度可以通过此接口来进行调节。

**dwBufNum** 值越大, 播放的流畅性越好, 相对延时就大;

**dwBufNum** 值越小, 播放的延时就小, 但是当网络不太顺畅的时候, 会有丢帧现象, 影响播放的流畅性。

若当前为混合流时, 为保证音视频同步效果建议设置缓冲帧数大于等于 6 帧。此函数必须紧跟在 [NET\\_DVR\\_RealPlay\\_V40](#) 后使用, 在图像播放之后设置则不起作用。

[返回目录](#)

## 5.7 预览时播放声音控制

### 5.7.1 设置声音播放模式 **NET\_DVR\_SetAudioMode**

函数: BOOL NET\_DVR\_SetAudioMode(DWORD dwMode)

参数: [in] dwMode 声音播放模式: 1—独占声卡, 单路音频模式;  
2—共享声卡, 多路音频模式

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 不调用该接口设置声音播放模式, 默认为独占播放。

[返回目录](#)

### 5.7.2 独占声卡模式下开启声音 **NET\_DVR\_OpenSound**

函数: BOOL NET\_DVR\_OpenSound(LONG IRealHandle)

参数: [in] IRealHandle NET\_DVR\_RealPlay\_V40 的返回值

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 如果当前是共享模式播放, 调用该接口将返回失败。以独占方式只能打开一路通道播放, 即依次打开多个通道时仅打开最后一路。

[返回目录](#)

### 5.7.3 独占声卡模式下开启声音 **NET\_DVR\_CloseSound**

函 数: BOOL NET\_DVR\_CloseSound()

参 数: 无

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

### 5.7.4 共享声卡模式下开启声音 **NET\_DVR\_OpenSoundShare**

函 数: BOOL NET\_DVR\_OpenSoundShare(LONG IRealHandle)

参 数: [in] IRealHandle NET\_DVR\_RealPlay\_V40 的返回值

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

### 5.7.5 共享声卡模式下关闭声音 **NET\_DVR\_CloseSoundShare**

函 数: BOOL NET\_DVR\_CloseSoundShare (LONG IRealHandle)

参 数: [in] IRealHandle NET\_DVR\_RealPlay\_V40 的返回值

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

### 5.7.6 调节播放音量 **NET\_DVR\_Volume**

函 数: BOOL NET\_DVR\_Volume(LONG IRealHandle,WORD wVolume)

参 数: [in] IRealHandle NET\_DVR\_RealPlay\_V40 的返回值

[in] wVolume 音量, 取值范围[0,0xffff]

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明: 该接口调节的是 PC 的音量。

[返回目录](#)

## 5.8 实时预览数据捕获

### 5.8.1 注册回调函数，捕获实时码流数据 **NET\_DVR\_SetRealDataCallBack**

**函数：** BOOL NET\_DVR\_SetRealDataCallBack(LONG IRealHandle, fRealDataCallBack cbRealDataCallBack, DWORD dwUser)

**参数：** [in] IRealHandle                    预览句柄，NET\_DVR\_RealPlay\_V40 的返回值  
[in] cbRealDataCallBack            码流数据回调函数  
[in] dwUser                            用户数据

```
typedef void(CALLBACK *fRealDataCallBack)(LONG IRealHandle, DWORD dwDataType, BYTE *pBuffer,
DWORD dwBufSize, DWORD dwUser)
```

[out] IRealHandle                    当前的预览句柄  
[out] dwDataType                    数据类型，详见表 5.7  
[out] pBuffer                        存放数据的缓冲区指针  
[out] dwBufSize                    缓冲区大小  
[out] dwUser                        用户数据

表 5.7 码流数据类型

dwDataType 宏定义	宏定义值	含义
NET_DVR_SYSHEAD	1	系统头数据
NET_DVR_STREAMDATA	2	流数据（包括复合流或音视频分开的视频流数据）

**返回值：** TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

**说明：** 此函数包括开始和停止用户处理 SDK 捕获的数据，当回调函数 cbRealDataCallBack 设为非 NULL 值时，表示回调和处理数据；当设为 NULL 时表示停止回调和处理数据。回调的第一个包是 40 个字节的文件头，供后续解码使用，之后回调的是压缩的码流。回调数据最大为 256K 字节。

[返回目录](#)

### 5.8.2 捕获数据并保存到指定的文件中 **NET\_DVR\_SaveRealData**

**函数：** BOOL NET\_DVR\_SaveRealData(LONG IRealHandle, char \*sFileName)

**参数：** [in] IRealHandle                    NET\_DVR\_RealPlay\_V40 的返回值  
[in] sFileName                        文件路径指针，绝对路径，包括文件名

**返回值：** TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

**说明：** V5.0.3.2 或以后版本，通过该接口保存录像，文件最大限制为 1024MB，大于 1024M 时，SDK 自动新建文件进行保存，文件开始将 40 字节头自动写入，文件名命名规则为“在接口传入的文件名基础上增加数字标识(例如：\*\_1.mp4、\*\_2.mp4)”。

[返回目录](#)

### 5.8.3 停止数据捕获 **NET\_DVR\_StopSaveRealData**

函数: BOOL NET\_DVR\_StopSaveRealData(LONG IRealHandle )  
 参数: [in] IRealHandle NET\_DVR\_RealPlay\_V40 的返回值  
 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

## 5.9 布防、撤防

### 设置报警等信息上传的回调函数

#### 5.9.1 注册回调函数, 接收设备报警消息

#### **NET\_DVR\_SetDVRMessageCallBack\_V31**

函数: BOOL NET\_DVR\_SetDVRMessageCallBack\_V31(MSGCallBack fMessageCallBack, void\* pUser)  
 参数: [in]fMessageCallBack 报警信息回调函数  
 [in]pUser 用户数据

```
typedef BOOL (CALLBACK *MSGCallBack)(LONG ICommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD dwBufLen, void *pUser)
```

[out]ICommand 上传的消息类型, 详见表 5.8  
 [out]pAlarmer 报警设备信息, 详见 [NET\\_DVR\\_ALARMER](#)  
 [out]pAlarmInfo 报警信息, 详见表 5.9  
 [out]dwBufLen 报警信息缓存大小  
 [out]pUser 用户数据

表 5.8 布防报警信息类型

ICommand 宏定义	宏定义值	含义
COMM_ITS_BLOCKLIST_ALARM	0x3057	车辆名单报警上传
COMM_VEHICLE_RECOG_RESULT	0x3062	车辆二次识别结果上传
COMM_ITS_ROAD_EXCEPTION	0x4500	路口设备异常报警上传
COMM_ITS_EXTERNAL_CONTROL_ALARM	0x4520	指示灯外控报警

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 该接口中回调函数的第一个参数 (ICommand) 和第三个参数 (pAlarmInfo) 是密切关联的, 其关系如表 5.9 所示。

表 5.9 报警信息结构

消息类型 (ICommand)	上传内容	pAlarmInfo 对应的结构体
COMM_ITS_BLOCKLIST_ALARM	车辆名单报警上传	<a href="#">NET_ITS_ECT_BLOCKLIST</a>
COMM_VEHICLE_RECOG_RESULT	车辆二次识别结果上传	<a href="#">NET_DVR_VEHICLE_RECOG_RESULT</a>
COMM_ITS_ROAD_EXCEPTION	路口设备异常报警信息	<a href="#">NET_ITS_ROADINFO</a>
COMM_ITS_EXTERNAL_CONTROL_ALARM	指示灯外控报警信息	<a href="#">NET_DVR_EXTERNAL_CONTROL_ALARM</a>

[返回目录](#)

## 布防撤防

### 5.9.2 建立报警上传通道，获取报警等信息 [NET\\_DVR\\_SetupAlarmChan\\_V41](#)

函数: LONG NET\_DVR\_SetupAlarmChan\_V41(LONG IUserID, LPNET\_DVR\_SETUPALARM\_PARAM lpSetupParam)

参数: [in] IUserID 用户 ID, NET\_DVR\_Login\_V30 的返回值  
[in] lpSetupParam 报警布防参数, 详见结构体: [NET\\_DVR\\_SETUPALARM\\_PARAM](#)

返回值: -1 表示失败, 其他值作为 NET\_DVR\_CloseAlarmChan\_V30 函数的句柄参数。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 启动布防前, 需要调用注册回调函数的接口 (如 [NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#)) 才能获取到上传的报警等信息。

[返回目录](#)

### 5.9.3 撤销报警上传通道 [NET\\_DVR\\_CloseAlarmChan\\_V30](#)

函数: BOOL NET\_DVR\_CloseAlarmChan\_V30(LONG IAlarmHandle)

参数: [in] IAlarmHandle NET\_DVR\_SetupAlarmChan\_V41 的返回值

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

## 5.10 监听报警

### 5.10.1 启动监听，接收设备主动上传的报警等信息 [NET\\_DVR\\_StartListen\\_V30](#)

函数: LONG NET\_DVR\_StartListen\_V30(char \*sLocalIP, WORD wLocalPort, MSGCallBack DataCallback, void\* pUserData)

参数: [in] sLocalIP PC 机本地 IP 地址, 可以置为 NULL  
[in] wLocalPort PC 本地监听端口号。由用户设置, 必须和设备端设置的一致  
[in] DataCallback 回调函数, 不能为 NULL  
[in] pUserData 用户数据

```
typedef void(CALLBACK *MSGCallBack)(LONG ICommand,NET_DVR_ALARMER *pAlarmer,char
*pAlarmInfo,DWORD dwBufLen,void *pUser)
```

[out] ICommand	上传的消息类型，详见表 5.10
[out] pAlarmer	报警设备信息，详见结构体： <a href="#">NET_DVR_ALARMER</a>
[out] pAlarmInfo	报警信息，详见表 5.11
[out] dwBufLen	报警信息缓存大小
[out] pUser	用户数据

表 5.10 监听报警信息类型

宏定义	宏定义值	含义
COMM_ALARM_TFS	0x1113	交通取证报警信息上传
COMM_ALARM_TPS_V41	0x1114	交通参数统计报警信息上传
COMM_ALARM_AID_V41	0x1115	交通事件报警信息上传(扩展)
COMM_ITS_PLATE_RESULT	0x3050	智能交通终端图片上传
COMM_ITS_TRAFFIC_COLLECT	0x3051	ITS 智能终端统计数据上传
COMM_ITS_GATE_VEHICLE	0x3052	出入口车辆抓拍数据上传
COMM_ITS_GATE_FACE	0x3053	出入口人脸抓拍数据上传
COMM_ITS_GATE_COSTITEM	0x3054	出入口过车收费明细上传
COMM_ITS_GATE_HANDOVER	0x3055	出入口交接班数据上传
COMM_ITS_PARK_VEHICLE	0x3056	停车场数据上传
COMM_ITS_BLOCKLIST_ALARM	0x3057	车辆名单报警上传
COMM_VEHICLE_RECOG_RESULT	0x3062	车辆二次识别结果上传
COMM_ITS_ROAD_EXCEPTION	0x4500	路口设备异常报警上传
COMM_ITS_EXTERNAL_CONTROL_ALARM	0x4520	指示灯外控报警

返回值：-1 表示失败，其他值作为 NET\_DVR\_CloseAlarmChan\_V30 函数的句柄参数。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

**说明：**该接口中回调函数的第一个参数（ICommand）和第三个参数（pAlarmInfo）是密切关联的，其关系如表 5.11 所示。

表 5.11 报警信息结构

消息类型（ICommand）	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM_TFS	交通取证报警信息	<a href="#">NET_DVR_TFS_ALARM</a>
COMM_ALARM_TPS_V41	交通参数统计报警信息	<a href="#">NET_DVR_TPS_ALARM_V41</a>
COMM_ALARM_AID_V41	交通事件报警信息扩展	<a href="#">NET_DVR_AID_ALARM_V41</a>
COMM_ITS_PLATE_RESULT	智能交通终端图片上传	<a href="#">NET_ITS_PLATE_RESULT</a>
COMM_ITS_TRAFFIC_COLLECT	终端统计数据上传	<a href="#">NET_ITS_TRAFFIC_COLLECT</a>
COMM_ITS_GATE_VEHICLE	出入口车辆抓拍数据上传	<a href="#">NET_ITS_GATE_VEHICLE</a>
COMM_ITS_GATE_FACE	出入口人脸抓拍数据上传	<a href="#">NET_ITS_GATE_FACE</a>
COMM_ITS_GATE_COSTITEM	出入口过车收费明细	<a href="#">NET_ITS_PASSVEHICLE_COST_ITEM</a>
COMM_ITS_GATE_HANDOVER	出入口交接班数据	<a href="#">NET_ITS_HANDOVER_INFO</a>

COMM_ITS_PARK_VEHICLE	停车场数据上传	<a href="#">NET_ITS_PARK_VEHICLE</a>
COMM_ITS_BLOCKLIST_ALARM	车辆名单报警上传	<a href="#">NET_ITS_BLOCKLIST_ALARM</a>
COMM_VEHICLE_RECOG_RESULT	车辆二次识别结果上传	<a href="#">NET_DVR_VEHICLE_RECOG_RESULT</a>
COMM_ITS_ROAD_EXCEPTION	路口设备异常报警信息	<a href="#">NET_ITS_ROADINFO</a>
COMM_ITS_EXTERNAL_CONTROL_ALARM	指示灯外控报警信息	<a href="#">NET_DVR_EXTERNAL_CONTROL_ALARM</a>

- 要使 PC 能够收到设备主动发过来的报警等信息，必须将设备的网络配置中的“远程主机地址”设置成 PC 机的 IP 地址（与接口中的 sLocalIP 参数一致），“远程主机端口号”设置成 PC 机的监听端口号（与接口中的 wLocalPort 参数一致）。
- **该接口中的回调函数优先级高于其他回调函数，即设置了该接口中的回调函数，其他报警回调函数将接收不到报警信息。**

[返回目录](#)

## 5.10.2 停止监听（支持多线程） **NET\_DVR\_StopListen\_V30**

函数： BOOL NET\_DVR\_StopListen\_V30(LONG IListenHandle)

参数： [in] IListenHandle 监听句柄，NET\_DVR\_StartListen\_V30 的返回值

返回值： TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说明：

[返回目录](#)

## 5.11 远程参数配置

### 5.11.1 获取设备的配置信息 **NET\_DVR\_GetDVRConfig**

函数： BOOL NET\_DVR\_GetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpOutBuffer, DWORD dwOutBufferSize, LPDWORD lpBytesReturned)

参数： [in] IUserID 用户 ID 号，NET\_DVR\_Login\_V30 的返回值

[in] dwCommand 设备配置命令，详见表 5.12

[in] IChannel 通道号，如果命令不需要通道号，该参数无效，置为 0xFFFFFFFF 即可

[out] lpOutBuffer 接收数据的缓冲指针，详见表 5.12

[in] dwOutBufferSize 接收数据的缓冲长度(以字节为单位)，不能为 0

[out] lpBytesReturned 实际收到的数据长度指针，不能为 NULL

返回值： TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说明：不同的获取功能对应不同的结构体和命令号，如表 5.12 所示。

- 当连接 ITS 终端进行配置，命令通道如果是模拟通道则到终端，如果是数字通道则到相机。
- 通道号是指设备视频通道号，通过注册设备 ([NET\\_DVR\\_Login\\_V30](#)) 返回的设备信息 (NET\_DVR\_DEVICEINFO\_V30) 获取模拟通道个数 (byChanNum)、模拟通道起始通道号 (byStartChan) 和设备支持的最大 IP 通道数 (byIPChanNum+ byHighDChanNum\*256)、数字

通道起始通道号 (byStartDChan)。

表 5.12 设备参数获取

dwCommand 宏定义	dwCommand 含义	通道号	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_DEVICECFG_V40	获取设备参数(扩展)	无效	<a href="#">NET_DVR_DEVICECFG_V40</a>	1100
NET_DVR_GET_IPPARACFG_V40	获取 IP 接入配置参数	通道号	<a href="#">NET_DVR_IPPARACFG_V40</a>	1062
NET_DVR_GET_DECODERCFG_V30	获取 (RS485)云台解码器参数	通道号	<a href="#">NET_DVR_DECODERCFG_V30</a>	1042
NET_DVR_GET_TIMECFG	获取时间参数	无效	<a href="#">NET_DVR_TIME</a>	118
NET_DVR_GET_EXCEPTIONCFG_V30	获取异常参数	无效	<a href="#">NET_DVR_EXCEPTION_V30</a>	1034
NET_DVR_GET_NETCFG_V30	获取网络参数	无效	<a href="#">NET_DVR_NETCFG_V30</a>	1000
NET_DVR_GET_NTPCFG	获取网络应用参数(NTP)	无效	<a href="#">NET_DVR_NTPPARA</a>	224
NET_DVR_GET_PICCFG_V30	获取图象参数	通道号	<a href="#">NET_DVR_PICCFG_V30</a>	1002
NET_DVR_GET_COMPRESSCFG_V30	获取压缩参数	通道号	<a href="#">NET_DVR_COMPRESSIONCFG_V30</a>	1040
NET_DVR_GET_RECORDCFG_V30	获取录像计划参数	通道号	<a href="#">NET_DVR_RECORD_V30</a>	1004
NET_DVR_GET_SHOWSTRING_V30	获取叠加字符参数	通道号	<a href="#">NET_DVR_SHOWSTRING_V30</a>	1030
NET_DVR_GET_HDCFG	获取硬盘管理参数	无效	<a href="#">NET_DVR_HDCFG</a>	1054
NET_ITS_GET_UPLOAD_CFG	获取数据上传配置	通道号	<a href="#">NET_ITS_UPLOAD_CFG</a>	5065
NET_ITS_GET_ROAD_INFO	获取路口信息	通道号	<a href="#">NET_ITS_ROADINFO</a>	5076
NET_ITS_GET_GATEIPC_CHAN_CFG	获取出入口参数	通道号	<a href="#">NET_ITS_IPC_CHAN_LANE_CFG</a>	5078
NET_ITS_GET_LAMP_CTRLCFG	获取内外置灯参数(ITS 停车场)	通道号	<a href="#">NET_DVR_LAMP_CTRL_INFO</a>	5090
NET_ITS_GET_PARKSPACE_ATTRIBUTE_CFG	获取特殊车位参数(ITS 停车场)	通道号	<a href="#">NET_DVR_PARKSPACE_ATTRIBUTE</a>	5092
NET_DVR_GET_CCDPARAMCFG_EX	获取相机前端参数(扩展)	通道号	<a href="#">NET_DVR_CAMERAPARAMCFG_EX</a>	3368
NET_DVR_GET_AID_RULECFG_V41	获取交通事件规则参数	通道号	<a href="#">NET_DVR_AID_RULECFG_V41</a>	5013
NET_DVR_GET_TPS_RULECFG_V41	获取交通统计规则参数	通道号	<a href="#">NET_DVR_TPS_RULECFG_V41</a>	5015

[返回目录](#)

### 5.11.2 设置设备的配置信息 **NET\_DVR\_SetDVRConfig**

函数: BOOL NET\_DVR\_SetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpInBuffer, DWORD dwInBufferSize)

参数:

- [in] IUserID 用户 ID 号, NET\_DVR\_Login\_V30 的返回值
- [in] dwCommand 设备配置命令, 详见表 5.13
- [in] IChannel 通道号。如果命令不需要通道号, 该参数无效, 置为 0xFFFFFFFF 即可
- [in] lpInBuffer 输入数据的缓冲指针, 详见表 5.13
- [in] dwInBufferSize 输入数据的缓冲长度(以字节为单位)

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 不同的获取功能对应不同的结构体和命令号, 如表 5.13 所示。

- 当连接 ITS 终端进行配置，命令通道如果是模拟通道则到终端，如果是数字通道则到相机。
- 通道号是指设备视频通道号，通过注册设备（[NET\\_DVR\\_Login\\_V30](#)）返回的设备信息（[NET\\_DVR\\_DEVICEINFO\\_V30](#)）获取模拟通道个数（byChanNum）、模拟通道起始通道号（byStartChan）和设备支持的最大 IP 通道数（byIPChanNum+ byHighDChanNum\*256）、数字通道起始通道号（byStartDChan）。

表 5.13 设备参数设置

dwCommand 宏定义	dwCommand 含义	通道号	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_DEVICECFG_V40	设置设备参数(扩展)	无效	<a href="#">NET_DVR_DEVICECFG_V40</a>	1101
NET_DVR_SET_IPPARACFG_V40	设置 IP 接入配置参数	通道号	<a href="#">NET_DVR_IPPARACFG_V40</a>	1063
NET_DVR_SET_DECODERCFG_V30	设置（RS485）云台解码器参数	通道号	<a href="#">NET_DVR_DECODERCFG_V30</a>	1043
NET_DVR_SET_TIMECFG	设置时间参数	无效	<a href="#">NET_DVR_TIME</a>	119
NET_DVR_SET_EXCEPTIONCFG_V30	设置异常参数	无效	<a href="#">NET_DVR_EXCEPTION_V30</a>	1035
NET_DVR_SET_NETCFG_V30	设置网络参数	无效	<a href="#">NET_DVR_NETCFG_V30</a>	1001
NET_DVR_SET_NTPCFG	设置网络应用参数(NTP)	无效	<a href="#">NET_DVR_NTPPARA</a>	225
NET_DVR_SET_PICCFG_V30	设置图像参数	通道号	<a href="#">NET_DVR_PICCFG_V30</a>	1003
NET_DVR_SET_COMPRESSCFG_V30	设置压缩参数	通道号	<a href="#">NET_DVR_COMPRESSIONCFG_V30</a>	1041
NET_DVR_SET_RECORDCFG_V30	设置录像参数	通道号	<a href="#">NET_DVR_RECORD_V30</a>	1005
NET_DVR_SET_SHOWSTRING_V30	设置叠加字符参数	通道号	<a href="#">NET_DVR_SHOWSTRING_V30</a>	1031
NET_DVR_SET_HDCFG	设置硬盘管理参数	无效	<a href="#">NET_DVR_HDCFG</a>	1055
NET_ITS_SET_UPLOAD_CFG	设置数据上传配置	通道号	<a href="#">NET_ITS_UPLOAD_CFG</a>	5066
NET_ITS_REMOTE_DEVICE_CONTROL	设置远程设备控制(ITS 出入口)	无效	<a href="#">NET_ITS_REMOTE_CONTROL</a>	5077
NET_ITS_SET_GATEIPC_CHAN_CFG	设置出入口参数	通道号	<a href="#">NET_ITS_IPC_CHAN_LANE_CFG</a>	5079
NET_ITS_SET_LAMP_CTRLCFG	设置内外置灯参数(ITS 停车场)	通道号	<a href="#">NET_DVR_LAMP_CTRL_INFO</a>	5091
NET_ITS_SET_PARKSPACE_ATTRIBUTE_CFG	设置特殊车位参数(ITS 停车场)	通道号	<a href="#">NET_DVR_PARKSPACE_ATTRIBUTE</a>	5093
NET_ITS_SET_LAMP_EXTERNAL_CFG	设置外控配置参数(ITS 停车场)	通道号	<a href="#">NET_DVR_LAMP_EXTERNAL_CFG</a>	5095
NET_ITS_SET_COMPEL_CAPTURE	设置车位强制抓图(ITS 停车场)	通道号	<a href="#">NET_DVR_COMPEL_CAPTURE</a>	5096
NET_DVR_SET_CCDPARAMCFG_EX	设置相机前端参数(扩展)	通道号	<a href="#">NET_DVR_CAMERAPARAMCFG_EX</a>	3369
NET_DVR_SET_AID_RULECFG_V41	设置交通事件规则参数	通道号	<a href="#">NET_DVR_AID_RULECFG_V41</a>	5014
NET_DVR_SET_TPS_RULECFG_V41	设置交通统计规则参数	通道号	<a href="#">NET_DVR_TPS_RULECFG_V41</a>	5016

[返回目录](#)

## 5.12 批量配置参数

### 5.12.1 批量获取配置信息 [NET\\_DVR\\_GetDeviceConfig](#)

函 数： BOOL NET\_DVR\_GetDeviceConfig(LONG IUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpOutBuffer, DWORD dwOutBufferSize)

参 数:	[in] IUserID	用户 ID 号, NET_DVR_Login_V30 的返回值
	[in] dwCommand	设备配置命令, 详见表 5.14
	[in] dwCount	一次要获取的配置个数, 0 和 1 都表示 1 个监控点信息, 2 表示 2 个监控点信息, 以此递增, 最大 64 个
	[in] lpInBuffer	配置条件缓冲区, 详见表 5.15
	[in] dwInBufferSize	缓冲区长度
	[out] lpStatusList	错误信息列表, 和要查询的监控点一一对应, 例如 lpStatusList[2] 就对应 lpInBuffer[2], 由用户分配内存, 每个错误信息为 4 个字节, 参数值: 0 或者 1 表示成功, 其他值为失败对应的错误号
	[out] lpOutBuffer	设备返回的参数内容 (详见表 5.15), 和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 0 值, 对应 lpOutBuffer 的内容就是无效的
	[in] dwOutBufferSize	输出缓冲区大小

表 5.14 参数批量获取命令

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_GET_CLOUDSTORAGE_CFG	获取云存储配置参数	5058
NET_DVR_GET_ITS_EXDEVCFG	获取 ITS 外接设备信息 (ITS 终端)	5085
NET_DVR_GET_FORENSICS_MODE	获取违章取证方式	5217

返回值: TRUE 表示成功但不代表每一个配置都成功, 哪一个成功, 对应查看 lpStatusList[n] 值; FALSE 表示全部失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明: 该接口是带有发送数据的批量获取配置信息的通用接口, lpInBuffer 指定需要获取的 dwCount 个配置信息, lpOutBuffer 保存获取到的 dwCount 个配置信息。不同的获取功能对应不同的结构体, 如表 5.15 所示。

表 5.15 批量获取设备参数

dwCommand	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_GET_CLOUDSTORAGE_CFG	dwCount 个 <a href="#">NET_DVR_CLOUDSTORAGE_COND</a>	dwCount 个 <a href="#">NET_DVR_CLOUDSTORAGE_CFG</a>
NET_DVR_GET_ITS_EXDEVCFG	dwCount 个 <a href="#">NET_DVR_EXTERNAL_DEVCOND</a>	dwCount 个 <a href="#">NET_DVR_EXTERNAL_DEVCFG</a>
NET_DVR_GET_FORENSICS_MODE	dwCount 个 4 字节通道号	dwCount 个 <a href="#">NET_DVR_FORENSICS_MODE</a>

[返回目录](#)

## 5.12.2 批量设置配置信息 [NET\\_DVR\\_SetDeviceConfig](#)

函 数: BOOL NET\_DVR\_SetDeviceConfig(LONG IUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpInParameterBuffer, DWORD dwInParameterBufferSize)

参 数:	[in] IUserID	用户 ID 号, NET_DVR_Login_V30 的返回值
	[in] dwCommand	设备配置命令, 详见表 5.16
	[in] dwCount	一次要设置的监控点个数, 0 和 1 都表示 1 个监控点信息, 2 表示 2 个监控点信息, 以此递增, 最大 64 个
	[in] lpInBuffer	配置条件缓冲区, 详见表 5.17
	[in] dwInBufferSize	缓冲区长度
	[out] lpStatusList	错误信息列表, 和要查询的监控点一一对应, 例如 lpStatusList[2]

就对应 `lpInBuffer[2]`，由用户分配内存，每个错误信息为 4 个字节，参数值：0 或者 1 表示成功，其他值为失败对应的错误号

[in] `lpInParamBuffer` 需要设置给设备的参数内容（详见表 5.17），和要查询的监控点一一对应。如果某个监控点对应的 `IpStatusList` 信息为大于 0 值，表示对应的 `lpInBuffer` 设置失败，为 0 则设置成功

[in] `dwInParamBufferSize` 设置内容缓冲区大小

表 5.16 参数批量设置命令

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_SET_VEHICLE_RECOG_TASK	车辆二次识别任务提交（ITS 终端）	422
NET_DVR_SET_CLOUDSTORAGE_CFG	设置云存储配置参数	5059
NET_DVR_SET_ITS_EXDEVCFG	设置 ITS 外接设备信息（ITS 终端）	5084
NET_DVR_SET_FORENSICS_MODE	设置违章取证方式	5218

返回值：TRUE 表示成功，但不代表每一个配置都成功，哪一个成功，对应查看 `IpStatusList[n]` 值；FALSE 表示全部失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

**说明：**该接口是带有发送数据的批量设置监控点配置信息的通用接口。`lpInBuffer` 指定需要设置的 `dwCount` 个监控点信息，`lpOutBuffer` 保存将要设置的 `dwCount` 个监控点的配置信息。不同的获取功能对应不同的结构体和命令号，如表 5.17 所示。

表 5.17 批量设置设备参数

dwCommand	lpInBuffer 对应结构体	lpInParamBuffer 对应结构体
NET_DVR_SET_VEHICLE_RECOG_TASK	dwCount 个 <a href="#">NET_DVR_VEHICLE_RECOG_COND</a>	dwCount 个 <a href="#">NET_DVR_VEHICLE_RECOG_CFG</a>
NET_DVR_SET_CLOUDSTORAGE_CFG	dwCount 个 <a href="#">NET_DVR_CLOUDSTORAGE_COND</a>	dwCount 个 <a href="#">NET_DVR_CLOUDSTORAGE_CFG</a>
NET_DVR_SET_ITS_EXDEVCFG	dwCount 个 <a href="#">NET_DVR_EXTERNAL_DEVCOND</a>	dwCount 个 <a href="#">NET_DVR_EXTERNAL_DEVCFG</a>
NET_DVR_SET_FORENSICS_MODE	dwCount 个 4 字节通道号	dwCount 个 <a href="#">NET_DVR_FORENSICS_MODE</a>

[返回目录](#)

## 5.13 长连接参数配置

### 5.13.1 启动长连接远程配置 `NET_DVR_StartRemoteConfig`

函数：LONG NET\_DVR\_StartRemoteConfig(LONG IUserID, DWORD dwCommand, LPVOID lpInBuffer, DWORD dwInBufferLen, fRemoteConfigCallback cbStateCallback, LPVOID pUserData)

参数：

- [in] IUserID 用户 ID 号，NET\_DVR\_Login\_V30 返回值
- [in] dwCommand 配置命令，详见表 5.18
- [in] lpInBuffer 输入参数，具体内容跟配置命令相关，详见表 5.18
- [in] dwInBufferLen 输入缓冲的大小
- [in] cbStateCallback 状态回调函数
- [in] pUserData 用户数据

```
typedef void(CALLBACK *fRemoteConfigCallback)(DWORD dwType, void *lpBuffer, DWORD dwBufLen, void *pUserData)
```

[out] dwType 配置状态

[out] lpBuffer 存放数据的缓冲区指针  
 [out] dwBufLen 缓冲区大小  
 [out] pUserData 用户数据

返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET\\_DVR\\_GetLastError](#)。

说明: 不同的配置命令 (dwCommand) 对应不同的输入参数 (lpInBuffer), 回调获取信息也不同, 详见表 5.18。调用该接口启动长连接远程配置后, 对于有些命令, 还需要调用其他接口获取或者设置相关参数, 如表 5.19 所示。

表 5.18 长连接配置

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer	回调函数
NET_DVR_GET_VEHICLE_RECOG_TASK	443	车辆二次识别任务获取	<a href="#">NET_DVR_VEHICLE_RECOG_TASK_COND</a>	NULL
NET_ITS_GET_ECT_CHAN_INFO	5082	获取出入口终端通道状态	NULL	NULL
NET_DVR_GET_ITS_EXDEVSTATUS	5086	获取 ITS 所有外接设备信息	<a href="#">NET_DVR_EXTERNAL_DEVCOND</a>	NULL
NET_DVR_GET_TRAFFIC_DATA	3141	获取交通数据	<a href="#">NET_DVR_TRAFFIC_DATA_QUERY_COND</a>	返回状态、 信息数据
NET_DVR_GET_TRAFFIC_FLOW	3142	获取交通流量	<a href="#">NET_DVR_TRAFFIC_FLOW_QUERY_COND</a>	返回状态、 信息数据

表 5.19 后续接口调用

dwCommand 宏定义	含义	后续接口调用
NET_DVR_GET_VEHICLE_RECOG_TASK	车辆二次识别任务获取	<a href="#">NET_DVR_GetNextRemoteConfig</a>
NET_ITS_GET_ECT_CHAN_INFO	获取出入口终端通道状态	<a href="#">NET_DVR_GetNextRemoteConfig</a>
NET_DVR_GET_ITS_EXDEVSTATUS	获取 ITS 所有外接设备信息	<a href="#">NET_DVR_GetNextRemoteConfig</a>
NET_DVR_GET_TRAFFIC_DATA	获取交通数据	NULL
NET_DVR_GET_TRAFFIC_FLOW	获取交通流量	NULL

表 5.20 长连接回调参数

dwCommand 宏定义	dwType	lpInBuffer
NET_DVR_GET_TRAFFIC_DATA	typedef enum{ NET_SDK_CALLBACK_TYPE_STATUS = 0, //回调状态值 NET_SDK_CALLBACK_TYPE_PROGRESS, //回调进度值	dwType 为 0 时, lpInBuffer 对应状态, 详见: NET_SDK_CALLBACK_STATUS_NORMAL dwType 为 2 时, lpInBuffer 为查找到的信息数据, 对应结构体 <a href="#">NET_DVR_TRAFFIC_DATA_QUERY_RESULT</a>
NET_DVR_GET_TRAFFIC_FLOW	NET_SDK_CALLBACK_TYPE_DATA //回调数据内容 }NET_SDK_CALLBACK_TYPE;	dwType 为 0 时, lpInBuffer 对应状态, 详见: NET_SDK_CALLBACK_STATUS_NORMAL dwType 为 2 时, lpInBuffer 为查找到的信息数据, 对应结构体 <a href="#">NET_DVR_TRAFFIC_FLOW_QUERY_RESULT</a>

[返回目录](#)

### 5.13.2 逐个获取查找到的结果信息 **NET\_DVR\_GetNextRemoteConfig**

函数: LONG NET\_DVR\_GetNextRemoteConfig(LONG IHandle, void \*IpOutBuff, DWORD dwOutBuffSize)

参数: [in] IHandle 查找句柄, NET\_DVR\_StartRemoteConfig 的返回值  
[in] IpOutBuff 输出数据缓冲区, 与 NET\_DVR\_StartRemoteConfig 的命令 (dwCommand) 有关, 详见表 5.22  
[in] dwOutBuffSize 缓冲区长度

返回值: -1 表示失败, 其他值表示当前的获取状态等信息, 如表 5.21 所示。获取错误码调用 [NET\\_DVR\\_GetLastError](#)。

表 5.21 长连接状态

宏定义	宏定义值	含义
NET_SDK_GET_NEXT_STATUS_SUCCESS	1000	成功读取到数据, 处理完本次数据后需要再次调用 NET_DVR_GetNextRemoteConfig 获取下一条数据
NET_SDK_GET_NETX_STATUS_NEED_WAIT	1001	需等待设备发送数据, 继续调用 NET_DVR_GetNextRemoteConfig
NET_SDK_GET_NEXT_STATUS_FINISH	1002	数据全部取完, 可调用 NET_DVR_StopRemoteConfig 结束长连接
NET_SDK_GET_NEXT_STATUS_FAILED	1003	出现异常, 可调用 NET_DVR_StopRemoteConfig 结束长连接

**说明:** 调用该接口获取查找结果之前, 必须先调用 [NET\\_DVR\\_StartRemoteConfig](#) 得到当前的查找句柄。此接口用于获取一条已查找到的信息, 若要获取全部已查找到的信息, 需要循环调用此接口。

表 5.22 长连接获取参数

dwCommand 宏定义	宏定义值	控制功能	IpOutBuff 对应结构体
NET_DVR_GET_VEHICLE_RECOG_TASK	443	车辆二次识别任务获取	<a href="#">NET_DVR_VEHICLE_RECOG_TASK_INFO</a>
NET_ITS_GET_ECT_CHAN_INFO	5082	获取出入口终端通道状态	<a href="#">NET_ITS_ECT_CHANNELSTATE</a>
NET_DVR_GET_ITS_EXDEVSTATUS	5086	获取 ITS 所有外接设备信息	<a href="#">NET_DVR_EXTERNAL_DEVSTATUS</a>

[返回目录](#)

### 5.13.3 关闭长连接配置接口所创建的句柄, 释放资源

#### **NET\_DVR\_StopRemoteConfig**

函数: BOOL NET\_DVR\_StopRemoteConfig(LONG IHandle)

参数: [in] IHandle 句柄, NET\_DVR\_StartRemoteConfig 的返回值

返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET\\_DVR\\_GetLastError](#)。

**说明:**

[返回目录](#)

## 5.14 远程控制

### 5.14.1 远程控制 **NET\_DVR\_RemoteControl**

函数: BOOL NET\_DVR\_RemoteControl(LONG IUserID, DWORD dwCommand, LPVOID lpInBuffer, DWORD dwInBufferSize)

参数: [in] IUserID 用户 ID 号, NET\_DVR\_Login\_V30 返回值  
[in] dwCommand 控制命令, 详见表 5.23  
[in] lpInBuffer 输入参数, 跟控制命令相关, 详见列表  
[in] dwInBufferSize 输入参数长度

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 不同的控制功能对应不同的命令号, 同时 lpInBuffer 对应不同的结构体, 如表 5.23 所示。

表 5.23 远程控制命令

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer 对应结构体
NET_DVR_SET_ITS_ENDEVCMO	5087	设置 ITS 终端出入口控制命令 (控制下发卡片信息)	<a href="#">NET_DVR_ENTRANCEDEV_COMMAND</a>
NET_DVR_SET_ENISSUED_DATADEL	5088	设置 ITS 终端出入口控制命令 (清除下发的卡片数据)	<a href="#">NET_DVR_ENISSUED_DATADEL</a>
NET_DVR_CONTROL_PTZ_MANUALTRACE	3316	手动定位	<a href="#">NET_DVR_PTZ_MANUALTRACE</a>

[返回目录](#)

## 5.15 录像回放和下载

### 5.15.1 按时间回放录像文件 **NET\_DVR\_PlayBackByTime\_V40**

函数: LONG NET\_DVR\_PlayBackByTime\_V40(LONG IUserID, LPNET\_DVR\_VOD\_PARA pVodPara)

参数: [in] IUserID NET\_DVR\_Login\_V30 的返回值  
[in] pVodPara 回放参数, 请参见结构体: [NET\\_DVR\\_VOD\\_PARA](#)

返回值: -1 表示失败, 其他值作为 NET\_DVR\_StopPlayBack 等函数的参数。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 该接口指定了当前要播放的录像, 调用成功后, 还必须调用 NET\_DVR\_PlayBackControl\_V40 接口的开始播放控制命令 (NET\_DVR\_PLAYSTART) 才能实现回放。

在调用该接口成功后, 可以通过接口 NET\_DVR\_SetPlayDataCallBack 注册回调函数, 捕获录像的码流数据并自行处理。

[返回目录](#)

### 5.15.2 按时间下载录像文件 **NET\_DVR\_GetFileByTime\_V40**

函数: LONG NET\_DVR\_GetFileByTime\_V40(LONG IUserID, char \*sSavedFileName, LPNET\_DVR\_PLAYCOND pDownloadCond)

参 数: [in] IUserID NET\_DVR\_Login\_V30 的返回值  
[in] sSavedFileName 下载后保存到 PC 机的文件路径, 需为绝对路径, 包括文件名  
[in] pDownloadCond 下载调节, 包括通道号、开始时间和停止时间等, 详见结构体:  
[NET\\_DVR\\_PLAYCOND](#)

返回值: -1 表示失败, 其他值作为 NET\_DVR\_StopGetFile 等函数的参数。接口返回失败请调用  
[NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明: ● 该接口指定了当前要下载的录像, 调用成功后, 还需要调用 NET\_DVR\_PlayBackControl\_V40 接口的开始播放控制命令 (NET\_DVR\_PLAYSTART) 才能实现下载。  
● V5.0.3.2 或以后版本, 通过该接口保存录像, 文件最大限制为 1024MB, 大于 1024M 时, SDK 自动新建文件进行保存, 文件开始将 40 字节头自动写入, 文件名命名规则为“在接口传入的文件名基础上增加数字标识(例如: \*\_1.mp4、\*\_2.mp4)”。

[返回目录](#)

### 5.15.3 控制录像回放的状态 NET\_DVR\_PlayBackControl\_V40

函 数: BOOL NET\_DVR\_PlayBackControl\_V40(LONG IPlayHandle, DWORD dwControlCode, LPVOID lpInBuffer = NULL, DWORD dwInLen = 0, LPVOID lpOutBuffer = NULL, DWORD \*lpOutLen = NULL)

参 数: [in] IPlayHandle 回放或者下载句柄, NET\_DVR\_PlayBackByTime\_V40 或者 NET\_DVR\_GetFileByTime\_V40 的返回值  
[in] dwControlCode 控制录像回放或者下载状态命令, 详见表 5.24  
[in] lpInBuffer 指向输入参数的指针  
[in] dwInLen 输入参数的长度  
[out] lpOutBuffer 指向输出参数的指针  
[out] lpOutLen 输出参数的长度

表 5.24 回放控制命令

dwControlCode 宏定义	宏定义值	含义
NET_DVR_PLAYSTART	1	开始播放或者下载
NET_DVR_PLAYPAUSE	3	暂停播放或者下载
NET_DVR_PLAYRESTART	4	恢复播放或者下载
NET_DVR_PLAYFAST	5	快放
NET_DVR_PLAYSLOW	6	慢放
NET_DVR_PLAYNORMAL	7	正常速度播放 (在暂停后调用将恢复暂停前的速度播放)
NET_DVR_PLAYFRAME	8	单帧放 (恢复正常回放使用 NET_DVR_PLAYNORMAL 命令)
NET_DVR_PLAYSTARTAUDIO	9	打开声音
NET_DVR_PLAYSTOPAUDIO	10	关闭声音
NET_DVR_PLAYAUDIOVOLUME	11	调节音量, 取值范围[0,0xffff]

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明: 该接口中的第三个参数是否需要输入数值与控制命令有关。当控制命令为 NET\_DVR\_PLAYSTART 时, lpInBuffer 指向一个 4 字节整型的偏移量, 智能交通设备不支持该功能, 值设为 0; 当控制命令为 NET\_DVR\_PLAYAUDIOVOLUME 指向一个 4 字节整型的音量值; 其他命令时, lpInBuffer 设为 NULL。

[返回目录](#)

## 回放录像文件时的数据捕获

### 5.15.4 捕获回放的录像数据，并保存成文件 **NET\_DVR\_PlayBackSaveData**

函数: BOOL NET\_DVR\_PlayBackSaveData(LONG IPlayHandle, char \*sFileName)

参数: [in]IPlayHandle 回放句柄, NET\_DVR\_PlayBackByTime\_V40 的返回值  
[in]sFileName 保存数据的文件路径, 绝对路径, 包括文件名

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: V5.0.3.2 或以后版本, 通过该接口保存录像, 文件最大限制为 1024MB, 大于 1024M 时, SDK 自动新建文件进行保存, 文件开始将 40 字节头自动写入, 文件名命名规则为“在接口传入的文件名基础上增加数字标识(例如: \*\_1.mp4、\*\_2.mp4)”。

[返回目录](#)

### 5.15.5 停止保存录像数据 **NET\_DVR\_StopPlayBackSave**

函数: BOOL NET\_DVR\_StopPlayBackSave(LONG IPlayHandle)

参数: [in]IPlayHandle 播放句柄, NET\_DVR\_PlayBackByTime\_V40 的返回值

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

### 5.15.6 注册回调函数，捕获录像数据 **NET\_DVR\_SetPlayDataCallBack\_V40**

函数: BOOL NET\_DVR\_SetPlayDataCallBack\_V40(LONG IPlayHandle, fPlayDataCallBack cbPlayDataCallBack, void \*pUser)

参数: [in]IPlayHandle 播放句柄, NET\_DVR\_PlayBackByName 或 NET\_DVR\_PlayBackByTime\_V40 的返回值  
[in]fPlayDataCallBack 录像数据回调函数  
[in] pUser 用户数据

```
typedef void(CALLBACK *fPlayDataCallBack_V40)(LONG IPlayHandle, DWORD dwDataType, BYTE *pBuffer, DWORD dwBufSize, void *pUser)
```

[out]IPlayHandle 当前的录像播放句柄  
[out]dwDataType 数据类型, 详见表 5.25  
[out]pBuffer 存放数据的缓冲区指针  
[out]dwBufSize 缓冲区大小  
[out]pUser 用户数据

表 5.25 回放回调数据类型

dwDataType 宏定义	宏定义值	含义
NET_DVR_SYSHEAD	1	系统头数据
NET_DVR_STREAMDATA	2	流数据 (包括复合流或音视频分开的视频流数据)

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通

过错误码判断出错原因。

- 说明：**
- 此函数包括开始和停止用户处理 SDK 捕获的数据，当回调函数 `cbPlayDataCallBack` 设为非 NULL 值时，表示回调和处理数据；当设为 NULL 时表示停止回调和处理数据。回调的第一个包是 40 个字节的文件头，供后续解码使用，之后回调的是压缩的码流。
  - `cbPlayDataCallBack` 回调函数中不能执行可能会占用时间较长的接口或操作，不建议调用该 SDK（HCNetSDK.dll）本身的接口。

[返回目录](#)

## 5.16 图片下载

### 5.16.1 获取图片数据并存放在指定的内存空间中 **NET\_DVR\_GetPicture\_V30**

**函数：** `BOOL NET_DVR_GetPicture_V30(LONG lUserID, char *sDVRFileName, char *sSavedFileBuf, DWORD dwBufLen, DWORD *lpdwRetLen)`

**参数：**

[in]lUserID	NET_DVR_Login_V30 的返回值
[in]sDVRFileName	图片名称
[in]sSavedFileName	保存图片的缓冲区
[in]dwBufLen	缓冲区大小
[out]lpdwRetLen	实际收到的数据长度指针，不能为 NULL

**返回值：** TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

**说明：** 图片为 JPEG 格式，数据保存在缓冲区 `sSavedFileName` 中，读取缓冲区的图片数据自己显示或者保存成文件。

[返回目录](#)

## 5.17 设备维护管理

### 远程升级

#### 5.17.1 设置远程升级时网络环境 **NET\_DVR\_SetNetworkEnvironment**

**函数：** `BOOL NET_DVR_SetNetworkEnvironment(DWORD dwEnvironmentLevel)`

**参数：** [in] dwEnvironmentLevel 网络环境级别

```
enum{
    LOCAL_AREA_NETWORK = 0, //局域网环境
    WIDE_AREA_NETWORK    //广域网环境
}
```

**返回值：** TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

**说明：** 接口中的网络环境级别参数分为两类：`LOCAL_AREA_NETWORK` 表示局域网环境(网络环境好，通讯流畅)；`WIDE_AREA_NETWORK` 表示广域网环境(网络环境差，易阻塞)。在调用远程升级接口之前，可以通过此接口适应不同的升级环境。

[返回目录](#)

### 5.17.2 远程升级 **NET\_DVR\_Upgrade**

函数: LONG NET\_DVR\_Upgrade(LONG IUserID, char \*sFileName)  
参数: [in] IUserID NET\_DVR\_Login\_V30 的返回值  
[in] sFileName 升级文件路径(包括文件名)。路径长度和操作系统有关, sdk 不做限制, windows 默认路径长度小于等于 256 字节(包括文件名)  
返回值: -1 表示失败, 其他值作为 NET\_DVR\_GetUpgradeState 等函数的参数。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

### 5.17.3 获取远程升级的进度 **NET\_DVR\_GetUpgradeProgress**

函数: int NET\_DVR\_GetUpgradeProgress(LONG IUpgradeHandle)  
参数: [in] IUpgradeHandle NET\_DVR\_Upgrade 的返回值  
返回值: -1 表示失败, 0~100 表示升级进度。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

### 5.17.4 获取远程升级的状态 **NET\_DVR\_GetUpgradeState**

函数: int NET\_DVR\_GetUpgradeState(LONG IUpgradeHandle)  
参数: [in] IUpgradeHandle NET\_DVR\_Upgrade 的返回值  
返回值: -1 表示失败, 其他值定义: 1- 升级成功; 2- 正在升级; 3- 升级失败; 4- 网络断开, 状态未知; 5- 升级文件语言版本不匹配。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

### 5.17.5 关闭远程升级句柄, 释放资源 **NET\_DVR\_CloseUpgradeHandle**

函数: BOOL NET\_DVR\_CloseUpgradeHandle(LONG IUpgradeHandle)  
参数: [in] IUpgradeHandle NET\_DVR\_Upgrade 的返回值  
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

## 日志查找

### 5.17.6 查找设备的日志信息 **NET\_DVR\_FindDVRLog\_V30**

**函数:** LONG NET\_DVR\_FindDVRLog\_V30(LONG IUserID, LONG ISelectMode, DWORD dwMajorType, DWORD dwMinorType, LPNET\_DVR\_TIME lpStartTime, LPNET\_DVR\_TIME lpStopTime, BOOL bOnlySmart)

**参数:**

[in]IUserID	NET_DVR_Login_V30 的返回值
[in]ISelectMode	查询方式: 0- 全部; 1- 按类型; 2- 按时间; 3- 按时间和类型
[in]dwMajorType	日志主类型 (S.M.A.R.T 搜索时无效), 0 表示全部类型, 其他类型定义请参见 <a href="#">设备日志主类型</a>
[in]dwMinorType	日志次类型 (S.M.A.R.T 搜索时无效), 0 表示全部类型, 根据不同的主类型的次类型定义请参见 <a href="#">设备日志次类型</a>
[in]lpStartTime	文件的开始时间, 详见结构体: <a href="#">NET_DVR_TIME</a>
[in]lpStopTime	文件结束时间, 详见结构体: <a href="#">NET_DVR_TIME</a>
[in]bOnlySmart	是否只搜索带 S.M.A.R.T 信息的日志

**返回值:** -1 表示失败, 其他值作为 NET\_DVR\_FindNextLog\_V30 等函数的参数。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

**说明:** 该接口如果用于搜索普通日志信息, 一般设备支持 2000 条, 而搜索带 S.M.A.R.T 信息 (硬盘运行日志记录) 的日志最大只支持 500 条。通常不需要搜索详细的 S.M.A.R.T 信息时, 置 bOnlySmart 为 FALSE 即可完成所有日志信息的搜索。

[返回目录](#)

### 5.17.7 逐条获取查找到的日志信息 **NET\_DVR\_FindNextLog\_V30**

**函数:** LONG NET\_DVR\_FindNextLog\_V30(LONG ILogHandle, LPNET\_DVR\_LOG\_V30 lpLogData)

**参数:**

[in]ILogHandle	日志查找句柄, NET_DVR_FindDVRLog_V30 的返回值
[out]lpLogData	保存日志信息的指针, 详见结构体: <a href="#">NET_DVR_LOG_V30</a>

**返回值:** -1 表示失败, 其他值表示当前的获取状态等信息。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

**说明:** 在调用该接口获取查找日志之前, 必须先调用 NET\_DVR\_FindDVRLog\_V30 得到当前的查找句柄。

[返回目录](#)

### 5.17.8 释放查找日志的资源 **NET\_DVR\_FindLogClose\_V30**

**函数:** BOOL NET\_DVR\_FindLogClose\_V30(LONG ILogHandle)

**参数:** [in]ILogHandle 日志查找句柄, NET\_DVR\_FindDVRLog\_V30 的返回值

**返回值:** TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

**说明:**

[返回目录](#)

## 恢复设备默认参数

### 5.17.9 恢复设备默认参数 **NET\_DVR\_RestoreConfig**

函数: BOOL NET\_DVR\_RestoreConfig(LONG IUserID)

参数: [in] IUserID 用户 ID 号, NET\_DVR\_Login\_V30 的返回值

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

## 导入/导出配置文件

### 5.17.10 导出配置文件 **NET\_DVR\_GetConfigFile\_V30**

函数: BOOL NET\_DVR\_GetConfigFile\_V30(LONG IUserID, char \*sOutBuffer, DWORD dwOutSize, DWORD \*pReturnSize)

参数: [in] IUserID 用户 ID 号, NET\_DVR\_Login\_V30 的返回值

[out] sOutBuffer 存放配置参数的缓冲区

[in] dwOutSize 缓冲区大小

[out] pReturnSize 实际获得的缓冲区大小

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 当 sOutBuffer = NULL、dwOutSize = 0 且 pReturnSize != NULL 时, 获取参数配置文件的所需的缓冲区长度; 当 sOutBuffer != NULL 且 dwOutSize != 0 时, 获取参数配置文件的所需的缓冲区内容。

[返回目录](#)

### 5.17.11 导出配置文件 **NET\_DVR\_GetConfigFile**

函数: BOOL NET\_DVR\_GetConfigFile(LONG IUserID, char \*sFileName)

参数: [in] IUserID 用户 ID 号, NET\_DVR\_Login\_V30 的返回值

[in] sFileName 存放保存配置文件的文件路径 (二进制文件)

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

### 5.17.12 导入配置文件 **NET\_DVR\_SetConfigFile\_EX**

函数: BOOL NET\_DVR\_SetConfigFile\_EX(LONG IUserID, char \*sInBuffer, DWORD dwInSize)

参数: [in] IUserID 用户 ID 号, NET\_DVR\_Login\_V30 的返回值

[in] sInBuffer                    存放配置参数的缓冲区

[in] dwInSize                    缓冲区大小

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

### 5.17.13 导入配置文件 **NET\_DVR\_SetConfigFile**

函 数: BOOL NET\_DVR\_SetConfigFile(LONG IUserID, char \*sFileName)

参 数: [in] IUserID                    用户 ID 号, NET\_DVR\_Login\_V30 的返回值

[in] sFileName                    存放保存配置文件的文件路径 (二进制文件)

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

## 关机和重启

### 5.17.14 重启设备 **NET\_DVR\_RebootDVR**

函 数: BOOL NET\_DVR\_RebootDVR(LONG IUserID)

参 数: [in] IUserID                    用户 ID 号, NET\_DVR\_Login\_V30 的返回值

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

### 5.17.15 关闭设备 **NET\_DVR\_ShutDownDVR**

函 数: BOOL NET\_DVR\_ShutDownDVR(LONG IUserID)

参 数: [in] IUserID                    用户 ID 号, NET\_DVR\_Login\_V30 的返回值

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

## 5.18 获取设备支持的云台协议

### 5.18.1 获取设备支持的云台协议 **NET\_DVR\_GetPTZProtocol**

函数: BOOL NET\_DVR\_GetPTZProtocol(LONG IUserID, NET\_DVR\_PTZCFG \*pPtzcfg)

参数: [in]IUserID 用户 ID 号, NET\_DVR\_Login\_V30 的返回值  
[out]pPtzcfg 设备的云台协议, 详见结构体: [NET\\_DVR\\_PTZCFG](#)

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 在配置前端云台协议时必须调用该接口获取当前设备支持的云台协议。

[返回目录](#)

## 5.19 获取 IPC 协议列表

### 5.19.1 获取设备支持的 IPC 协议表 **NET\_DVR\_GetIPCProtoList**

函数: BOOL NET\_DVR\_GetIPCProtoList(LONG IUserID, LPNET\_DVR\_IPC\_PROTO\_LIST IpProtoList)

参数: [in]IUserID NET\_DVR\_Login\_V30 的返回值  
[out]IpProtoList IPC 协议列表结构, 详见结构体: [NET\\_DVR\\_IPC\\_PROTO\\_LIST](#)

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 该接口用于获取当前设备所支持的外接 IPC 的协议。

[返回目录](#)

## 5.20 获取设备状态

### 5.20.1 获取设备状态信息 **NET\_DVR\_GetDeviceStatus**

函数: BOOL NET\_DVR\_GetDeviceStatus(LONG IUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpOutBuffer, DWORD dwOutBufferSize)

参数: [in] IUserID 用户 ID 号, NET\_DVR\_Login\_V30 返回值  
[in] dwCommand 获取设备状态的命令值  
[in] dwCount 要获取的状态个数, 设为 1  
[in] lpInBuffer 配置条件缓冲区  
[in] dwInBufferSize 缓冲区长度  
[out] lpStatusList 错误信息列表, 和要查询的监控点一一对应, 例如 lpStatusList[2] 就对应 lpInBuffer[2], 由用户分配内存, 每个错误信息为 4 个字节(1 个 32 位无符号整数值), 参数值: 0- 成功, 大于 0- 失败  
[out] lpOutBuffer 设备返回的参数内容, 和要查询的监控点一一对应。如果某个监

控点对应的 IpStatusList 信息为大于 0 值，对应 IpOutBuffer 的内容就是无效的

[in] dwOutBufferSize 输出缓冲区大小

返回值: TRUE 表示成功，但不代表每一个配置都成功，哪一个成功，对应查看 IpStatusList[n]值；FALSE 表示全部失败。接口返回失败请调用 [NET\\_DVR\\_GetLastError](#) 获取错误码，通过错误码判断出错原因。

**说明:** 该接口是带有发送数据的批量获取设备状态信息的通用接口。不同的获取功能对应不同的结构体和命令号，如下表所示。

表 5.26 获取设备状态

dwCommand 宏定义	dwCommand 含义	IpInBuffer	IpOutBuffer 对应结构体	宏定义值
NET_ITS_GET_WORKSTATE	获取智能交通终端工作状态	NULL	dwCount 个 <a href="#">NET_ITS_WORKSTATE</a>	5069
NET_ITS_GET_ECTWORKSTATE	获取出入口终端工作状态	dwCount 个 <a href="#">NET_ITS_ECT_WORKSTATE_COMMAND</a>	dwCount 个 <a href="#">NET_ITS_ECTWORKSTATE</a>	5081

[返回目录](#)

## 6 错误代码及说明

### 6.1 网络通讯库错误码

错误名称	错误值	说明
NET_DVR_NOERROR	0	没有错误。
NET_DVR_PASSWORD_ERROR	1	用户名密码错误。注册时输入的用户名或者密码错误。
NET_DVR_NOENOUGHPRI	2	权限不足。该注册用户没有权限执行当前对设备的操作，可以与远程用户参数配置做对比。
NET_DVR_NOINIT	3	SDK 未初始化。
NET_DVR_CHANNEL_ERROR	4	通道号错误。设备没有对应的通道号。
NET_DVR_OVER_MAXLINK	5	连接到设备的用户个数超过最大。
NET_DVR_VERSIONNOMATCH	6	版本不匹配。SDK 和设备的版本不匹配。
NET_DVR_NETWORK_FAIL_CONNECT	7	连接设备失败。设备不在线或网络原因引起的连接超时等。
NET_DVR_NETWORK_SEND_ERROR	8	向设备发送失败。
NET_DVR_NETWORK_RECV_ERROR	9	从设备接收数据失败。
NET_DVR_NETWORK_RECV_TIMEOUT	10	从设备接收数据超时。
NET_DVR_NETWORK_ERRORDATA	11	传送的数据有误。发送给设备或者从设备接收到的数据错误，如远程参数配置时输入设备不支持的值。
NET_DVR_ORDER_ERROR	12	调用次序错误。
NET_DVR_OPERNOPERMIT	13	无此权限。
NET_DVR_COMMANDTIMEOUT	14	设备命令执行超时。
NET_DVR_ERRORSERIALPORT	15	串口号错误。指定的设备串口号不存在。
NET_DVR_ERRORALARMPORT	16	报警端口错误。指定的设备报警输出端口不存在。
NET_DVR_PARAMETER_ERROR	17	参数错误。SDK 接口中给入的输入或输出参数为空。
NET_DVR_CHAN_EXCEPTION	18	设备通道处于错误状态
NET_DVR_NODISK	19	设备无硬盘。当设备无硬盘时，对设备的录像文件、硬盘配置等操作失败。
NET_DVR_ERRORDISKNUM	20	硬盘号错误。当对设备进行硬盘管理操作时，指定的硬盘号不存在时返回该错误。
NET_DVR_DISK_FULL	21	设备硬盘满。
NET_DVR_DISK_ERROR	22	设备硬盘出错
NET_DVR_NOSUPPORT	23	设备不支持。
NET_DVR_BUSY	24	设备忙。
NET_DVR_MODIFY_FAIL	25	设备修改不成功。
NET_DVR_PASSWORD_FORMAT_ERROR	26	密码输入格式不正确
NET_DVR_DISK_FORMATING	27	硬盘正在格式化，不能启动操作。
NET_DVR_DVRNORESOURCE	28	设备资源不足。
NET_DVR_DVROPRATEFAILED	29	设备操作失败。
NET_DVR_OPENHOSTSOUND_FAIL	30	语音对讲、语音广播操作中采集本地音频或打开音频输出失败。
NET_DVR_DVRVOICEOPENED	31	设备语音对讲被占用。
NET_DVR_TIMEINPUTERROR	32	时间输入不正确。

NET_DVR_NOSPECFILE	33	回放时设备没有指定的文件。
NET_DVR_CREATEFILE_ERROR	34	创建文件出错。本地录像、保存图片、获取配置文件和远程下载录像时创建文件失败。
NET_DVR_FILEOPENFAIL	35	打开文件出错。设置配置文件、设备升级、上传审讯文件时打开文件失败。
NET_DVR_OPERNOTFINISH	36	上次的操作还没有完成
NET_DVR_GETPLAYTIMEFAIL	37	获取当前播放的时间出错。
NET_DVR_PLAYFAIL	38	播放出错。
NET_DVR_FILEFORMAT_ERROR	39	文件格式不正确。
NET_DVR_DIR_ERROR	40	路径错误
NET_DVR_ALLOC_RESOURCE_ERROR	41	SDK 资源分配错误。
NET_DVR_AUDIO_MODE_ERROR	42	声卡模式错误。当前打开声音播放模式与实际设置的模式不符出错。
NET_DVR_NOENOUGH_BUF	43	缓冲区太小。接收设备数据的缓冲区或存放图片缓冲区不足。
NET_DVR_CREATESOCKET_ERROR	44	创建 SOCKET 出错。
NET_DVR_SETSOCKET_ERROR	45	设置 SOCKET 出错。
NET_DVR_MAX_NUM	46	个数达到最大。分配的注册连接数、预览连接数超过 SDK 支持的最大数。
NET_DVR_USERNOTEXIST	47	用户不存在。注册的用户 ID 已注销或不可用。
NET_DVR_WRITEFLASHERROR	48	写 FLASH 出错。设备升级时写 FLASH 失败。
NET_DVR_UPGRADEFAIL	49	设备升级失败。网络或升级文件语言不匹配等原因升级失败。
NET_DVR_CARDHAVEINIT	50	解码卡已经初始化过。
NET_DVR_PLAYERFAILED	51	调用播放库中某个函数失败。
NET_DVR_MAX_USERNUM	52	登录设备的用户数达到最大。
NET_DVR_GETLOCALIPANDMACFAIL	53	获得本地 PC 的 IP 地址或物理地址失败。
NET_DVR_NOENCODEING	54	设备该通道没有启动编码。
NET_DVR_IPMISMATCH	55	IP 地址不匹配。
NET_DVR_MACMISMATCH	56	MAC 地址不匹配。
NET_DVR_UPGRADELANGMISMATCH	57	升级文件语言不匹配。
NET_DVR_MAX_PLAYERPORT	58	播放器路数达到最大。
NET_DVR_NOSPACEBACKUP	59	备份设备中没有足够空间进行备份。
NET_DVR_NODEVICEBACKUP	60	没有找到指定的备份设备。
NET_DVR_PICTURE_BITS_ERROR	61	图像素位数不符，限 24 色。
NET_DVR_PICTURE_DIMENSION_ERROR	62	图片高*宽超限，限 128*256。
NET_DVR_PICTURE_SIZ_ERROR	63	图片大小超限，限 100K。
NET_DVR_LOADPLAYERSDKFAILED	64	载入当前目录下 Player Sdk 出错。
NET_DVR_LOADPLAYERSDKPROC_ERROR	65	找不到 Player Sdk 中某个函数入口。
NET_DVR_LOADDSSDKFAILED	66	载入当前目录下 DsSdk 出错。
NET_DVR_LOADDSSDKPROC_ERROR	67	找不到 DsSdk 中某个函数入口。
NET_DVR_DSSDK_ERROR	68	调用硬解码库 DsSdk 中某个函数失败。
NET_DVR_VOICEMONOPOLIZE	69	声卡被独占。

NET_DVR_JOINMULTICASTFAILED	70	加入多播组失败。
NET_DVR_CREATEDIR_ERROR	71	建立日志文件目录失败。
NET_DVR_BINDSOCKET_ERROR	72	绑定套接字失败。
NET_DVR_SOCKETCLOSE_ERROR	73	socket 连接中断，此错误通常是由于连接中断或目的地不可达。
NET_DVR_USERID_ISUSING	74	注销时用户 ID 正在进行某操作。
NET_DVR_SOCKETLISTEN_ERROR	75	监听失败。
NET_DVR_PROGRAM_EXCEPTION	76	程序异常。
NET_DVR_WRITEFILE_FAILED	77	写文件失败。本地录像、远程下载录像、下载图片等操作时写文件失败。
NET_DVR_FORMAT_READONLY	78	禁止格式化只读硬盘。
NET_DVR_WITHSAMEUSERNAME	79	远程用户配置结构中存在相同的用户名。
NET_DVR_DEVICETYPE_ERROR	80	导入参数时设备型号不匹配。
NET_DVR_LANGUAGE_ERROR	81	导入参数时语言不匹配。
NET_DVR_PARAVERSION_ERROR	82	导入参数时软件版本不匹配。
NET_DVR_IPCHAN_NOTALIVE	83	预览时外接 IP 通道不在线。
NET_DVR_RTSP_SDK_ERROR	84	加载标准协议通讯库 StreamTransClient 失败。
NET_DVR_CONVERT_SDK_ERROR	85	加载转封装库失败。
NET_DVR_IPC_COUNT_OVERFLOW	86	超出最大的 IP 接入通道数。
NET_DVR_MAX_ADD_NUM	87	添加录像标签或者其他操作超出最多支持的个数。
NET_DVR_PARAMMODE_ERROR	88	图像增强仪，参数模式错误（用于硬件设置时，客户端进行软件设置时错误值）。
NET_DVR_CODESPITTER_OFFLINE	89	码分器不在线。
NET_DVR_BACKUP_COPYING	90	设备正在备份。
NET_DVR_CHAN_NOTSUPPORT	91	通道不支持该操作。
NET_DVR_CALLINEINVALID	92	高度线位置太集中或长度线不够倾斜。
NET_DVR_CALCANCELCONFLICT	93	取消标定冲突，如果设置了规则及全局的实际大小尺寸过滤。
NET_DVR_CALPOINTOUTRANGE	94	标定点超出范围。
NET_DVR_FILTERRECTINVALID	95	尺寸过滤器不符合要求。
NET_DVR_DDNS_DEVOFFLINE	96	设备没有注册到 ddns 上。
NET_DVR_DDNS_INTER_ERROR	97	DDNS 服务器内部错误。
NET_DVR_ALIAS_DUPLICATE	150	别名重复（EasyDDNS 的配置）
NET_DVR_DEV_NET_OVERFLOW	800	网络流量超过设备能力上限
NET_DVR_STATUS_RECORDFILE_WRITING_NOT_LOCK	801	录像文件在录像，无法被锁定
NET_DVR_STATUS_CANT_FORMAT_LITTLE_DISK	802	由于硬盘太小无法格式化
<b>智能交通错误码</b>		
NET_DVR_ERR_LANENUM_EXCEED	1400	车道数超出能力。
NET_DVR_ERR_PRAREA_EXCEED	1401	牌识区域过大。
NET_DVR_ERR_LIGHT_PARAM	1402	信号灯接入参数错误。
NET_DVR_ERR_LANE_LINE_INVALID	1403	车道线配置错误。
NET_DVR_ERR_STOP_LINE_INVALID	1404	停止线配置错误。
NET_DVR_ERR_LEFTORRIGHT_LINE_INVALID	1405	左/右转分界线配置错误。
NET_DVR_ERR_LANE_NO_REPEAT	1406	叠加车道号重复。

NET_DVR_ERR_PRAREA_INVALID	1407	牌识多边形不符合要求。
NET_DVR_ERR_LIGHT_NUM_EXCEED	1408	视频检测交通灯信号灯数目超出最大值。
NET_DVR_ERR_SUBLIGHT_NUM_INVALID	1409	视频检测交通灯信号灯子灯数目不合法
NET_DVR_ERR_LIGHT_AREASIZE_INVALID	1410	视频检测交通灯输入信号灯框大小不合法。
NET_DVR_ERR_LIGHT_COLOR_INVALID	1411	视频检测交通灯输入信号灯颜色不合法。
NET_DVR_ERR_LIGHT_DIRECTION_INVALID	1412	视频检测交通灯输入灯方向属性不合法。

## 6.2 RTSP 通讯库错误码

错误名称	错误值	说明
NET_DVR_RTSP_GETPORTFAILED	407	获取 RTSP 端口错误
NET_DVR_RTSP_DESCRIBESENDTIMEOUT	411	RTSP DESCRIBE 发送超时
NET_DVR_RTSP_DESCRIBESENDERROR	412	RTSP DESCRIBE 发送失败
NET_DVR_RTSP_DESCRIBERECVTIMEOUT	413	RTSP DESCRIBE 接收超时
NET_DVR_RTSP_DESCRIBERECDATALOST	414	RTSP DESCRIBE 接收数据错误
NET_DVR_RTSP_DESCRIBERECVERROR	415	RTSP DESCRIBE 接收失败
NET_DVR_RTSP_DESCRIBESERVERERR	416	RTSP DESCRIBE 服务器返回 401,501 等错误
NET_DVR_RTSP_SETUPSENDTIMEOUT	421	RTSP SETUP 发送超时
NET_DVR_RTSP_SETUPSENDERROR	422	RTSP SETUP 发送错误
NET_DVR_RTSP_SETUPRECVTIMEOUT	423	RTSP SETUP 接收超时
NET_DVR_RTSP_SETUPRECDATALOST	424	RTSP SETUP 接收数据错误
NET_DVR_RTSP_SETUPRECVERROR	425	RTSP SETUP 接收失败
NET_DVR_RTSP_OVER_MAX_CHAN	426	设备超过最大连接数
NET_DVR_RTSP_PLAYSENDTIMEOUT	431	RTSP PLAY 发送超时
NET_DVR_RTSP_PLAYSENDERROR	432	RTSP PLAY 发送错误
NET_DVR_RTSP_PLAYRECVTIMEOUT	433	RTSP PLAT 接收超时
NET_DVR_RTSP_PLAYRECDATALOST	434	RTSP PLAY 接收数据错误
NET_DVR_RTSP_PLAYRECVERROR	435	RTSP PLAY 接收失败
NET_DVR_RTSP_PLAYSERVERERR	436	RTSP PLAY 设备返回错误状态
NET_DVR_RTSP_TEARDOWNSENDTIMEOUT	441	RTSP TEARDOWN 发送超时
NET_DVR_RTSP_TEARDOWNSENDERROR	442	RTSP TEARDOWN 发送错误
NET_DVR_RTSP_TEARDOWNRECVTIMEOUT	443	RTSP TEARDOWN 接收超时
NET_DVR_RTSP_TEARDOWNRECDATALOST	444	RTSP TEARDOWN 接收数据错误
NET_DVR_RTSP_TEARDOWNRECVERROR	445	RTSP TEARDOWN 接收失败
NET_DVR_RTSP_TEARDOWNSERVERERR	446	RTSP TEARDOWN 设备返回错误状态

## 6.3 软解码库错误码

错误名称	错误值	说明
NET_PLAYM4_NOERROR	500	没有错误

NET_PLAYM4_PARA_OVER	501	输入参数非法
NET_PLAYM4_ORDER_ERROR	502	调用顺序不对
NET_PLAYM4_TIMER_ERROR	503	多媒体时钟设置失败
NET_PLAYM4_DEC_VIDEO_ERROR	504	视频解码失败
NET_PLAYM4_DEC_AUDIO_ERROR	505	音频解码失败
NET_PLAYM4_ALLOC_MEMORY_ERROR	506	分配内存失败
NET_PLAYM4_OPEN_FILE_ERROR	507	文件操作失败
NET_PLAYM4_CREATE_OBJ_ERROR	508	创建线程事件等失败
NET_PLAYM4_CREATE_DDRAW_ERROR	509	创建 directDraw 失败
NET_PLAYM4_CREATE_OFFSCREEN_ERROR	510	创建后端缓存失败
NET_PLAYM4_BUF_OVER	511	缓冲区满，输入流失败
NET_PLAYM4_CREATE_SOUND_ERROR	512	创建音频设备失败
NET_PLAYM4_SET_VOLUME_ERROR	513	设置音量失败
NET_PLAYM4_SUPPORT_FILE_ONLY	514	只能在播放文件时才能使用此接口
NET_PLAYM4_SUPPORT_STREAM_ONLY	515	只能在播放流时才能使用此接口
NET_PLAYM4_SYS_NOT_SUPPORT	516	系统不支持，解码器只能工作在 Pentium 3 以上
NET_PLAYM4_FILEHEADER_UNKNOWN	517	没有文件头
NET_PLAYM4_VERSION_INCORRECT	518	解码器和编码器版本不对应
NET_PLAYM4_INIT_DECODER_ERROR	519	初始化解码器失败
NET_PLAYM4_CHECK_FILE_ERROR	520	文件太短或码流无法识别
NET_PLAYM4_INIT_TIMER_ERROR	521	初始化多媒体时钟失败
NET_PLAYM4_BLT_ERROR	522	位拷贝失败
NET_PLAYM4_UPDATE_ERROR	523	显示 overlay 失败
NET_PLAYM4_OPEN_FILE_ERROR_MULTI	524	打开混合流文件失败
NET_PLAYM4_OPEN_FILE_ERROR_VIDEO	525	打开视频流文件失败
NET_PLAYM4_JPEG_COMPRESS_ERROR	526	JPEG 压缩错误
NET_PLAYM4_EXTRACT_NOT_SUPPORT	527	不支持该文件版本.
NET_PLAYM4_EXTRACT_DATA_ERROR	528	提取文件数据失败

## 7 附录:结构体

### 7.1 宏定义

宏定义	宏定义值	含义
MAX_NAMELEN	16	设备本地登录名长度
MAX_RIGHT	32	设备支持的权限（1-12 表示本地权限，13-32 表示远程权限）
NAME_LEN	32	用户名长度
PASSWD_LEN	16	密码长度
SERIALNO_LEN	48	序列号长度
MACADDR_LEN	6	MAC 地址长度
MAX_LINK	6	最大视频流连接数
MAX_ANALOG_CHANNUM	32	最大 32 个模拟通道
MAX_ANALOG_ALARMOUT	32	最大 32 路模拟报警输出
MAX_ANALOG_ALARMIN	32	最大 32 路模拟报警输入
MAX_IP_CHANNEL	32	允许加入的最多 IP 通道数
MAX_IP_ALARMIN	128	允许加入的最多报警输入数
MAX_IP_ALARMOUT	64	允许加入的最多报警输出数
MAX_CHANNUM_V30	64	(MAX_ANALOG_CHANNUM + MAX_IP_CHANNEL)
MAX_ALARMOUT_V30	96	(MAX_ANALOG_ALARMOUT + MAX_IP_ALARMOUT)
MAX_ALARMIN_V30	160	(MAX_ANALOG_ALARMIN + MAX_IP_ALARMIN)
MAX_PRESET_V30	256	支持的云台预置点数
MAX_TRACK_V30	256	支持的云台花样扫描路径数
MAX_CRUISE_V30	256	支持的云台巡航数
MAX_ETHERNET	2	设备可配以太网络
MAX_DOMAIN_NAME	64	最大域名长度
MAX_EXCEPTIONNUM_V30	32	设备的最大异常处理数
MAX_SHELTERNUM	4	设备的最大遮挡区域数
MAX_STRINGNUM_V30	8	最大 OSD 字符行数
MAX_TIMESEGMENT_V30	8	最大时间段数
MAX_DAYS	7	每周的天数
MAX_DISKNUM_V30	33	最大硬盘数
MAX_LICENSE_LEN	16	车牌号最大长度
MAX_USERNUM_V30	32	设备的最大用户数
VERSION_LEN	32	版本长度
IPC_PROTOCOL_NUM	50	IPC 协议最大个数

MAX_IP_DEVICE_V40	64	允许接入的最大 IP 设备数
STREAM_ID_LEN	32	流 ID 长度
DESC_LEN	16	云台描述字符串长度
PTZ_PROTOCOL_NUM	200	最大支持的云台协议数
MONITORSITE_ID_LEN	48	监测点编号长度
DEVICE_ID_LEN	48	设备编号长度
ITS_MAX_DEVICE_NUM	32	最大设备个数
MAX_PARKING_NUM	4	一个通道最大车位个数
MAX_PARKING_STATUS	8	车位状态个数
MAX_ID_LEN	48	编号最大长度
MAX_ALARMREASON_LEN	32	报警类型描述最大长度
MAX_DEVNAME_LEN	32	设备名称最大长度
MAX_LED_INFO	256	屏幕字体显示信息最大长度
MAX_TIME_LEN	32	时间最大长度
MAX_CARD_LEN	24	卡号最大长度
MAX_OPERATORNAME_LEN	32	操作人员名称最大长度
MAX_INDEX_LED	8	LED 索引最大值
MAX_AID_RULE	8	最大事件规则数目
MAX_TPS_RULE	8	最大参数规则数目

## 7.2 NET\_DVR\_ACTIVATECFG:设备激活参数

```
struct{
    DWORD    dwSize;
    BYTE     sPassword[PASSWD_LEN];
    BYTE     byRes[108];
}NET_DVR_ACTIVATECFG,*LPNET_DVR_ACTIVATECFG;
```

### Members

#### *dwSize*

结构体大小

#### *sPassword*

初始密码，密码等级弱或者以上

#### *byRes*

保留，置为 0

### Remarks

- 出厂设备需要先激活，然后再使用激活使用的初始密码登录设备。
- 将密码输入分为数字(0~9)、小写字母(a~z)、大写字母(A~Z)、特殊符号 (:\|除外) 4 类，等级分为 4 个等级，如下所示：
  - 1) 等级 0 (风险密码): 密码长度小于 8 位，或者只包含 4 类字符中的任意一类，或者密码与用户名一样，或者密码是用户名的倒写。例如：12345、abcdef。

- 2) 等级 1 (弱密码): 包含两类字符, 且组合为 (数字+小写字母) 或 (数字+大写字母), 且长度大于等于 8 位。例如: abc12345、123ABCDEF
- 3) 等级 2 (中密码): 包含两类字符, 且组合不能为 (数字+小写字母) 和 (数字+大写字母), 且长度大于等于 8 位。例如: 12345\*\*\*++、ABCDabcd。
- 4) 等级 3 (强密码): 包含三类字符及以上, 且长度大于等于 8 位。例如: Abc12345、abc12345++。

## 7.3 NET\_DVR\_AID\_ALARM\_V41:交通事件报警信息

```
struct{
    DWORD                dwSize;
    DWORD                dwRelativeTime;
    DWORD                dwAbsTime;
    NET\_VCA\_DEV\_INFO    struDevInfo;
    NET\_DVR\_AID\_INFO    struAIDInfo;
    NET\_DVR\_SCENE\_INFO struSceneInfo;
    DWORD                dwPicDataLen;
    BYTE                *pImage;
    BYTE                byDataType;
    BYTE                byLaneNo;
    WORD                wMilliSecond;
    BYTE                byMonitoringSiteID[MONITORSITE_ID_LEN];
    BYTE                byDeviceID[DEVICE_ID_LEN];
    BYTE                byRes[28];
}NET_DVR_AID_ALARM_V41, *LPNET_DVR_AID_ALARM_V41;
```

### Members

#### *dwSize*

结构体大小

#### *dwRelativeTime*

相对时标

#### *dwAbsTime*

绝对时标

#### *struDevInfo*

前端设备信息

#### *struAIDInfo*

交通事件信息

#### *struSceneInfo*

场景信息

#### *dwPicDataLen*

图片长度

#### *pImage*

图片长度指向图片的指针

#### *byDataType*

数据上传方式: 0-数据直接上传; 1-云存储服务 URL (原先的图片数据变成 URL 数据, 图片长度变成 URL 长度)

*byLaneNo*

关联车道号

*wMilliSecond*

时标毫秒

*byMonitoringSiteID*

监测点编号（路口编号、内部编号）

*byDeviceID*

设备编号

*byRes*

保留

### Remarks

- 从 DWORD 时间值得到年月日时分秒的算法如下所示：
 

```
#define GET_YEAR(_time_)      (((_time_)>>26) + 2000)
#define GET_MONTH(_time_)    (((_time_)>>22) & 15)
#define GET_DAY(_time_)      (((_time_)>>17) & 31)
#define GET_HOUR(_time_)     (((_time_)>>12) & 31)
#define GET_MINUTE(_time_)   (((_time_)>>6) & 63)
#define GET_SECOND(_time_)   (((_time_)>>0) & 63)
```
- 通过 [NET\\_DVR\\_CLOUDSTORAGE\\_CFG](#) 配置可以启用云存储功能，则上传的图片信息将变成获取图片信息的 URL 地址，平台通过该 URL 地址去云存储服务器上获取数据
- 图片云存储 URL 格式：
 

```
http://CVMIP:Port/pic?did=DevID&bid=BlkID&pid=PictureID&ptime=PicTime
```

CVMIP: CVM(云存储服务器)的 IP 地址  
 Port: CVM(云存储服务器)对外提供 http 服务的端口（固定 8009）  
 DevID: CVS(云存储服务器)中设备 ID 号  
 BlkID: CVS(云存储服务器)中设备的块号  
 PictureID: CVS(云存储服务器)为图片生成的编号  
 PicTime: 图片的时间戳

示例：  
 http://10.192.65.140:8009/pic?did=35b9cbd0-8ffa-1031-87e6-0025903c6a50&bid=387&pid=2952790009&ptime=1378106185

## 7.4 NET\_DVR\_AID\_INFO:交通事件信息

```
struct{
    BYTE                byRuleID;
    BYTE                byRes1[3];
    BYTE                byRuleName[NAME_LEN];
    DWORD              dwAIDType;
    NET_DVR_DIRECTION  struDirect;
    BYTE                byRes2[40];
}NET_DVR_AID_INFO, *LPNET_DVR_AID_INFO;
```

### Members

*byRuleID*

规则序号，取值范围：1~8

*byRes1*

保留

*byRuleName*

规则名称

*dwAIDType*

交通事件类型，具体定义如下：

```
enum _TRAFFIC_AID_TYPE_{
    CONGESTION                = 0x1,
    PARKING                   = 0x2,
    INVERSE                   = 0x4,
    PEDESTRIAN               = 0x8,
    DEBRIS                    = 0x10,
    SMOG                      = 0x20,
    OVERLINE                  = 0x40,
    VEHICLE_CONTROL_LIST     = 0x80,
    SPEED                     = 0x100,
    LANECHANGE                = 0x200,
    TURNAROUND                = 0x400,
    VEHICLE_OCCUPANCY_NONVEHICLE = 0x800,
    GASSER                    = 0x1000
}TRAFFIC_AID_TYPE
```

*CONGESTION*

拥堵

*PARKING*

停车

*INVERSE*

逆行

*PEDESTRIAN*

行人

*DEBRIS*

遗留物，抛洒物碎片

*SMOG*

烟雾

*OVERLINE*

压线

*VEHICLE\_CONTROL\_LIST*

名单数据

*SPEED*

超速

*LANECHANGE*

变道

*TURNAROUND*

掉头

*VEHICLE\_OCCUPANCY\_NONVEHICLE*

机动车占用非机动车位

*GASSER*

加塞

*struDirect*

报警指向区域

*byRes2*

保留

## 7.5 NET\_DVR\_AID\_PARAM:交通事件参数

```
struct{
    WORD    wParkingDuration;
    WORD    wPedestrianDuration;
    WORD    wDebrisDuration;
    WORD    wCongestionLength;
    WORD    wCongestionDuration;
    WORD    wInverseDuration;
    WORD    wInverseDistance;
    WORD    wInverseAngleTolerance;
    WORD    wIllegalParkingTime;
    WORD    wIllegalParkingPicNum;
    BYTE    byMergePic;
    BYTE    byRes1[23];
}NET_DVR_AID_PARAM, *LPNET_DVR_AID_PARAM;
```

### Members

*wParkingDuration*

停车持续时间, 范围: 5~120s

*wPedestrianDuration*

行人持续时间, 范围: 1~120s

*wDebrisDuration*

抛洒物持续时间, 范围: 5~120s

*wCongestionLength*

拥堵长度阈值, 范围: 5~200 米

*wCongestionDuration*

拥堵持续时间, 范围: 5~120s

*wInverseDuration*

逆行持续时间, 范围: 1~120s

*wInverseDistance*

逆行距离阈值, 范围: 2~100 米, 默认为 10 米

*wInverseAngleTolerance*

允许角度偏差, 范围: 90~180 度。车流与逆行允许的夹角

*wIllegalParkingTime*

违停时间, 取值范围: [4,60], 单位: 分钟, TFS (交通违章取证) 城市模式下有效

***willegalParkingPicNum***

违停图片数量，取值范围：[1,6]，TFS（交通违章取证）城市模式下有效

***byMergePic***

图片拼接：0- 不拼接，1- 拼接，TFS（交通违章取证）城市模式下有效

***byRes1***

保留

**Remarks**

- **wParkingDuration** 是算法库灵敏度，指停车多少秒后算法库给出停车报警，接着分析仪根据这个停车报警作为起始报警点对该目标进行持续性的处理，超过违停时间后就整个取证报警事件结束。  
**willegalParkingTime** 是指违停时间限制，比如城市违章停车需要持续停车 20 分钟或几分钟后才能上报违停报警，在规定时间内是允许停车的。
- 对于不同设备，以上参数的取值范围可能不同。可以调用 [NET\\_DVR\\_GetDeviceAbility](#) 获取智能通道分析能力集（VcaChanAbility，能力集类型：DEVICE\_ABILITY\_INFO）而得到实际取值范围。

**7.6 NET\_DVR\_AID\_RULECFG\_V41:交通事件规则**

```
struct{
    DWORD                dwSize;
    BYTE                 byPicProType;
    BYTE                 byRes1[3];
    NET\_DVR\_JPEGPARA    struPictureParam;
    NET_DVR_ONE_AID_RULE_V41  struAIDRule[MAX_AID_RULE];
    BYTE                 byRes2[128];
}NET_DVR_AID_RULECFG_V41,*LPNET_DVR_AID_RULECFG_V41;
```

**Members*****dwSize***

结构体大小

***byPicProType***

报警时图片处理方式：0-不处理，非 0-上传

***byRes1***

保留，置为 0

***struPictureParam***

图片规格结构

***struAIDRule***

单条交通事件规则参数

***byRes2***

保留

**7.7 NET\_DVR\_ALARMER:报警设备信息**

```
struct{
    BYTE    byUserIDValid;
    BYTE    bySerialValid;
```

```

BYTE    byVersionValid;
BYTE    byDeviceNameValid;
BYTE    byMacAddrValid;
BYTE    byLinkPortValid;
BYTE    byDeviceIPValid;
BYTE    bySocketIPValid;
LONG    lUserID;
BYTE    sSerialNumber[SERIALNO_LEN];
DWORD   dwDeviceVersion;
char    sDeviceName[NAME_LEN];
BYTE    byMacAddr[MACADDR_LEN];
WORD    wLinkPort;
char    sDeviceIP[128];
char    sSocketIP[128];
BYTE    byIpProtocol;
BYTE    byRes2[11];
}NET_DVR_ALARMER,*LPNET_DVR_ALARMER;

```

### Members

#### *byUserIDValid*

userid 是否有效：0—无效；1—有效

#### *bySerialValid*

序列号是否有效：0—无效；1—有效

#### *byVersionValid*

版本号是否有效：0—无效；1—有效

#### *byDeviceNameValid*

设备名字是否有效：0—无效；1—有效

#### *byMacAddrValid*

MAC 地址是否有效：0—无效；1—有效

#### *byLinkPortValid*

Login 端口是否有效：0—无效；1—有效

#### *byDeviceIPValid*

设备 IP 是否有效：0—无效；1—有效

#### *bySocketIPValid*

Socket IP 是否有效：0-无效；1-有效

#### *lUserID*

NET\_DVR\_Login 或 NET\_DVR\_Login\_V30 返回值，布防时有效

#### *sSerialNumber*

序列号

#### *dwDeviceVersion*

版本信息：V3.0 以上版本支持的设备最高 8 位为主版本号，次高 8 位为次版本号，低 16 位为修复版本号；V3.0 以下版本支持的设备高 16 位表示主版本，低 16 位表示次版本

#### *sDeviceName*

设备名称

#### *byMacAddr*

MAC 地址

*wLinkPort*

设备通讯端口

*sDeviceIP*

设备 IP 地址

*sSocketIP*

报警主动上传时的 Socket IP 地址

*byIpProtocol*

IP 协议：0—IPV4；1—IPV6

*byRes2*

保留，置为 0

## 7.8 NET\_DVR\_BACKLIGHT:背光补偿参数

```
struct{
    BYTE        byBacklightMode;
    BYTE        byBacklightLevel;
    BYTE        byRes1[2];
    DWORD       dwPositionX1;
    DWORD       dwPositionY1;
    DWORD       dwPositionX2;
    DWORD       dwPositionY2;
    BYTE        byRes2[4];
}NET_DVR_BACKLIGHT, *LPNET_DVR_BACKLIGHT;
```

### Members

*byBacklightMode*

背光补偿模式：0-off、1-UP、2-DOWN、3-LEFT、4-RIGHT、5-MIDDLE、6-自定义

*byBacklightLevel*

背光补偿等级：0x0-0xF

*byRes1*

保留

*dwPositionX1*

X 坐标 1

*dwPositionY1*

Y 坐标 1

*dwPositionX2*

X 坐标 2

*dwPositionY2*

Y 坐标 2

*byRes2*

保留

## 7.9 NET\_DVR\_CAMERAPARAMCFG\_EX:前端参数配置(扩展)

```

struct{
    DWORD
    NET\_DVR\_VIDEOEFFECT          dwSize;
    NET\_DVR\_GAIN                 struVideoEffect;
    NET\_DVR\_WHITEBALANCE        struGain;
    NET\_DVR\_EXPOSURE           struWhiteBalance;
    NET\_DVR\_GAMMACORRECT       struExposure;
    NET\_DVR\_WDR                struGammaCorrect;
    NET\_DVR\_DAYNIGHT          struWdr;
    NET\_DVR\_BACKLIGHT         struDayNight;
    NET\_DVR\_NOISEREMOVE       struBackLight;
    BYTE                          struNoiseRemove;
    BYTE                          byPowerLineFrequencyMode;
    BYTE                          byIrisMode;
    BYTE                          byMirror;
    BYTE                          byDigitalZoom;
    BYTE                          byDeadPixelDetect;
    BYTE                          byBlackPwl;
    BYTE                          byEptzGate;
    BYTE                          byLocalOutputGate;
    BYTE                          byCoderOutputMode;
    BYTE                          byLineCoding;
    BYTE                          byDimmerMode;
    BYTE                          byPaletteMode;
    BYTE                          byEnhancedMode;
    BYTE                          byDynamicContrastEN;
    BYTE                          byDynamicContrast;
    BYTE                          byJPEGQuality;
    NET\_DVR\_CMOSMODECFG        struCmosModeCfg;
    BYTE                          byFilterSwitch;
    BYTE                          byFocusSpeed;
    BYTE                          byAutoCompensationInterval;
    BYTE                          bySceneMode;
    NET\_DVR\_DEFOGCFG          struDefogCfg;
    NET\_DVR\_ELECTRONICSTABILIZATION struElectronicStabilization;
    NET\_DVR\_CORRIDOR\_MODE\_CCD struCorridorMode;
    BYTE                          byExposureSegmentEnable;
    BYTE                          byBrightCompensate;
    BYTE                          byRes[310];
}NET_DVR_CAMERAPARAMCFG_EX, *LPNET_DVR_CAMERAPARAMCFG_EX;

```

### Members

*dwSize*

结构体大小

*struVideoEffect*

视频效果参数

*struGain*

增益参数

*struWhiteBalance*

白平衡参数

*struExposure*

曝光参数

*struGammaCorrect*

Gamma 校正参数

*struWdr*

宽动态参数

*struDayNight*

日夜转换功能参数

*struBackLight*

背光补偿参数

*struNoiseRemove*

数字降噪参数

*byPowerLineFrequencyMode*

0-50HZ; 1-60HZ

*byIrisMode*

0- 自动光圈; 1- 手动光圈

*byMirror*

镜像: 0- 关闭, 1- 左右, 2- 上下, 3- 中间

*byDigitalZoom*

数字缩放: 0- 不启用, 1- 启用。对于热成像仪, 表示数字倍率: 0- 关闭, 1-x2, 2-x4

*byDeadPixelDetect*

坏点检测是否启用, 0-不启用, 1-启用

*byBlackPwl*

黑电平补偿, 0-255

*byEptzGate*

EPTZ 开关变量: 0-关闭电子云台, 1-开启电子云台

*byLocalOutputGate*

本地输出开关变量:

0-本地输出关闭

1-本地 BNC 输出打开

2-HDMI 输出关闭

11-缩放输出

12-裁剪输出

13-裁剪缩放输出

20-HDMI\_720P50 输出开

21-HDMI\_720P60 输出开

22-HDMI\_1080I60 输出开

23-HDMI\_1080I50 输出开  
 24-HDMI\_1080P24 输出开  
 25-HDMI\_1080P25 输出开  
 26-HDMI\_1080P30 输出开  
 27-HDMI\_1080P50 输出开  
 28-HDMI\_1080P60 输出开

**byCoderOutputMode**

编码器 fpga 输出模式：0-直通；3-像素搬家

**byLineCoding**

是否开启行编码：0- 否，1- 是

**byDimmerMode**

调光模式：0- 半自动，1- 自动，适用于热成像仪

**byPaletteMode**

调色板：0- 白热，1- 黑热，2- 调色板 2，...，8- 调色板 8，适用于热成像仪

**byEnhancedMode**

增强方式（探测物体周边）：0- 不增强，1- 1，2- 2，3- 3，4- 4，适用于热成像仪

**byDynamicContrastEN**

动态对比度增强：0-不增强；1-增强

**byDynamicContrast**

动态对比度：0~100

**byJPEGQuality**

JPEG 图像质量：0~100

**struCmosModeCfg**

CMOS 模式下前端参数配置，镜头模式从能力集获取

**byFilterSwitch**

滤波开关：0- 不启用，1- 启用，适用于热成像仪

**byFocusSpeed**

镜头调焦速度，取值范围 0~10，适用于热成像仪

**byAutoCompensationInterval**

定时自动快门补偿，取值范围 1~120，单位：分钟，适用于热成像仪

**bySceneMode**

场景模式：0- 室外，1- 室内

**struDefogCfg**

透雾参数

**struElectronicStabilization**

电子防抖

**struCorridorMode**

旋转功能

**byExposureSegmentEnable**

曝光时间和增益呈阶梯状调整：0- 不启用，1- 启用，比如曝光往上调整时，先提高曝光时间到中间值，然后提高增益到中间值，再提高曝光到最大值，最后提高增益到最大值

**byBrightCompensate**

夜晚亮度增强，取值范围：[0,100]

**byRes**

保留，置为 0

## 7.10 NET\_DVR\_CHANNELSTATE\_V30:通道状态信息

```
struct{
    BYTE                byRecordStatic;
    BYTE                bySignalStatic;
    BYTE                byHardwareStatic;
    BYTE                byRes1;
    DWORD               dwBitRate;
    DWORD               dwLinkNum;
    NET\_DVR\_IPADDR    struClientIP[MAX_LINK];
    DWORD               dwIPLinkNum;
    BYTE                byExceedMaxLink;
    BYTE                byRes[11];
}NET_DVR_CHANNELSTATE_V30,*LPNET_DVR_CHANNELSTATE_V30;
```

### Members

#### *byRecordStatic*

通道是否在录像：0—不录像；1—录像

#### *bySignalStatic*

连接的信号状态：0—正常，1—信号丢失

#### *byHardwareStatic*

通道硬件状态：0—正常，1—异常（例如 DSP 异常）

#### *byRes1*

保留，置为 0

#### *dwBitRate*

实际码率

#### *dwLinkNum*

连接的客户端个数

#### *struClientIP*

连接的客户端 IP 地址

#### *dwIPLinkNum*

如果该通道为 IP 接入，表示 IP 接入当前的连接数

#### *byExceedMaxLink*

是否超出了单路 6 路连接数 0 - 未超出, 1-超出

#### *byRes*

保留，置为 0

## 7.11 NET\_DVR\_CLOUDSTORAGE\_CFG:云存储配置

```
struct{
    DWORD               dwSize;
    BYTE                byEnable;
```

```

BYTE                byRes[3];
NET_DVR_IPADDR     struIP;
WORD                wPort;
BYTE                byRes1[2];
char                szUser[CLOUD_NAME_LEN];
char                szPassword[CLOUD_PASSWD_LEN];
NET_DVR_POOLPARAM  struPoolInfo[16];
BYTE                byRes2[128];
}NET_DVR_CLOUDSTORAGE_CFG,*LPNET_DVR_CLOUDSTORAGE_CFG;

```

### Members

*dwSize*

结构体大小

*byEnable*

是否启用：0- 不启用，1- 启用

*byRes*

保留，置为 0

*struIP*

云存储服务器 IP 地址

*wPort*

云存储服务器端口号

*byRes1*

保留，置为 0

*szUser*

服务器用户名

*szPassword*

服务器密码

*struPoolInfo*

云存储数据池参数

*byRes2*

保留，置为 0

## 7.12 NET\_DVR\_CLOUDSTORAGE\_COND:云存储配置条件

```

struct{
    DWORD  dwSize;
    DWORD  dwChannel;
    BYTE   byRes1[64];
}NET_DVR_CLOUDSTORAGE_COND,*LPNET_DVR_CLOUDSTORAGE_COND;

```

### Members

*dwSize*

结构体大小

*dwChannel*

通道号

*byRes1*

保留，置为 0

### 7.13 NET\_DVR\_CMOSMODECFG:CMOS 模式下前端镜头配置

```
struct{
    BYTE    byCaptureMod;
    BYTE    byBrightnessGate;
    BYTE    byCaptureGain1;
    BYTE    byCaptureGain2;
    DWORD   dwCaptureShutterSpeed1;
    DWORD   dwCaptureShutterSpeed2;
    BYTE    byRes[4];
}NET_DVR_CMOSMODECFG, *LPNET_DVR_CMOSMODECFG;
```

#### Members

##### *byCaptureMod*

抓拍模式：0-抓拍模式 1；1-抓拍模式 2

##### *byBrightnessGate*

亮度阈值

##### *byCaptureGain1*

抓拍增益 1，0-100

##### *byCaptureGain2*

抓拍增益 2，0-100

##### *dwCaptureShutterSpeed1*

抓拍快门速度 1

##### *dwCaptureShutterSpeed2*

抓拍快门速度 2

##### *byRes*

保留

### 7.14 NET\_DVR\_COMPEL\_CAPTURE:车位强制抓图参数

```
struct{
    DWORD   dwSize;
    BYTE    byParkIndex;
    BYTE    byRes[63];
}NET_DVR_COMPEL_CAPTURE, *LPNET_DVR_COMPEL_CAPTURE;
```

#### Members

##### *dwSize*

结构体大小

##### *byParkIndex*

车位序号，从左到右：1、2、3、4

##### *byRes*

保留，置为 0

## 7.15 NET\_DVR\_COMPRESSIONCFG\_V30:通道压缩参数

```
struct{
    DWORD                dwSize;
    NET_DVR_COMPRESSION_INFO_V30  struNormHighRecordPara;
    NET_DVR_COMPRESSION_INFO_V30  struRes;
    NET_DVR_COMPRESSION_INFO_V30  struEventRecordPara;
    NET_DVR_COMPRESSION_INFO_V30  struNetPara;
}NET_DVR_COMPRESSIONCFG_V30, *LPNET_DVR_COMPRESSIONCFG_V30;
```

### Members

#### *dwSize*

结构体大小

#### *struNormHighRecordPara*

录像的码流压缩参数（即主码流的压缩参数）

#### *struRes*

保留，置为 0

#### *struEventRecordPara*

事件触发压缩参数，智能交通摄像机不支持，保留

#### *struNetPara*

网传的码流压缩参数（即子码流的压缩参数）

## 7.16 NET\_DVR\_COMPRESSION\_INFO\_V30:码流压缩参数

```
struct{
    BYTE    byStreamType;
    BYTE    byResolution;
    BYTE    byBitrateType;
    BYTE    byPicQuality;
    DWORD   dwVideoBitrate;
    DWORD   dwVideoFrameRate;
    WORD    wIntervalFrameI;
    BYTE    byIntervalBPFrame;
    BYTE    byres1;
    BYTE    byVideoEncType;
    BYTE    byAudioEncType;
    BYTE    byVideoEncComplexity;
    BYTE    byres[9];
}NET_DVR_COMPRESSION_INFO_V30, *LPNET_DVR_COMPRESSION_INFO_V30;
```

### Members

#### *byStreamType*

码流类型：0-视频流，1-复合流

#### *byResolution*

分辨率：0-DCIF(528\*384/528\*320)，1-CIF(352\*288/352\*240)，2-QCIF(176\*144/176\*120)，

3-4CIF(704\*576/704\*480)或 D1(720\*576/720\*486), 4-2CIF(704\*288/704\*240), 6-QVGA(320\*240), 7-QQVGA(160\*120), 12-384\*288, 13-576\*576, 16-VGA(640\*480), 17-UXGA(1600\*1200), 18-SVGA(800\*600), 19-HD720P(1280\*720), 20-XVGA(1280\*960), 21-HD900P(1600\*900), 23-1536\*1536, 24-1920\*1920, 27-1920\*1080p, 28-2560\*1920, 29-1600\*304, 30-2048\*1536, 31-2448\*2048, 32-2448\*1200, 33-2448\*800, 34-XGA(1024\*768), 35-SXGA(1280\*1024), 36-WD1(960\*576/960\*480), 37-1080i(1920\*1080), 38-WXGA(1440\*900), 39-HD\_F(1920\*1080/1280\*720), 40-HD\_H(1920\*540/1280\*360), 41-HD\_Q(960\*540/630\*360), 42-2336\*1744, 43-1920\*1456, 44-2592\*2048, 45-3296\*2472, 46-1376\*768, 47-1366\*768, 48-1360\*768, 49-WSXGA+, 50-720\*720, 51-1280\*1280, 52-2048\*768, 53-2048\*2048, 54-2560\*2048, 55-3072\*2048, 56-2304\*1296, 57-WXGA(1280\*800), 58-1600\*600, 59-1600\*900, 0xff-Auto(使用当前码流分辨率)

#### *byBitrateType*

码率类型: 0-变码率, 1-定码率

#### *byPicQuality*

图象质量: 0-最好, 1-次好, 2-较好, 3-一般, 4-较差, 5-差

#### *dwVideoBitrate*

码率: 0-保留, 1-16K(保留), 2-32K, 3-48k, 4-64K, 5-80K, 6-96K, 7-128K, 8-160k, 9-192K, 10-224K, 11-256K, 12-320K, 13-384K, 14-448K, 15-512K, 16-640K, 17-768K, 18-896K, 19-1024K, 20-1280K, 21-1536K, 22-1792K, 23-2048K, 24-3072K, 25-4096K, 26-8192K, 27-16384K。

最高位(31 位)置成 1 表示是自定义码流, 0~30 位表示码流值, 最小值 16k

#### *dwVideoFrameRate*

帧率: 0-全部, 1-1/16, 2-1/8, 3-1/4, 4-1/2, 5-1, 6-2, 7-4, 8-6, 9-8, 10-10, 11-12, 12-16, 13-20, 14-15, 15-18, 16-22

#### *wIntervalFrameI*

I 帧间隔, 0xffff-无效

#### *byIntervalBPFrame*

帧格式: 0-BBP 帧, 1-BP 帧, 2-单 P 帧, 0xff-无效

#### *byres1*

保留, 置为 0

#### *byVideoEncType*

视频编码类型: 0-私有 264, 1-标准 h264, 2-标准 mpeg4, 7-M-JPEG, 0xff-无效

#### *byAudioEncType*

音频编码类型: 0-OggVorbis, 1-G711\_U, 2-G711\_A, 6-G726, 0xff-无效

#### *byVideoEncComplexity*

视频编码复杂度: 0- 低, 1- 中, 2- 高

#### *byres*

保留, 置为 0

#### Remarks

当修改设备视频编码类型时, 需要重启设备生效。

## 7.17 NET\_DVR\_CORRIDOR\_MODE\_CCD:旋转功能参数

```
struct{
    BYTE    byEnableCorridorMode;
    BYTE    byRes[11];
```

```
}NET_DVR_CORRIDOR_MODE_CCD, *LPNET_DVR_CORRIDOR_MODE_CCD;
```

### Members

*byEnableCorridorMode*

是否启用旋转功能：0- 不启用，1- 启用

*byRes*

保留

### Remarks

开启旋转功能后，视频编码将逆时针旋转 90°，例如 1280\*720 旋转为 720\*1280。摄像机旋转安装时，启用该功能，可以提高垂直方向的监控有效范围，适用于走廊、道路等场所。

## 7.18 NET\_DVR\_DAYNIGHT: 日夜转换功能参数

```
struct{
    BYTE        byDayNightFilterType;
    BYTE        bySwitchScheduleEnabled;
    BYTE        byBeginTime;
    BYTE        byEndTime;
    BYTE        byDayToNightFilterLevel;
    BYTE        byNightToDayFilterLevel;
    BYTE        byDayNightFilterTime;
    BYTE        byBeginTimeMin;
    BYTE        byBeginTimeSec;
    BYTE        byEndTimeMin;
    BYTE        byEndTimeSec;
    BYTE        byAlarmTrigState;
}NET_DVR_DAYNIGHT, *LPNET_DVR_DAYNIGHT;
```

### Members

*byDayNightFilterType*

日夜切换：0-白天，1-夜晚，2-自动，3-定时，4-报警输入触发

*bySwitchScheduleEnabled*

0- 启动， 1- 禁用。（保留）

*byBeginTime*

定时模式开始时间（小时），取值范围：0~23

*byEndTime*

定时模式结束时间（小时），取值范围：0~23

*byDayToNightFilterLevel*

0~7

*byNightToDayFilterLevel*

0~7

*byDayNightFilterTime*

60 秒

*byBeginTimeMin*

定时模式开始时间（分），取值范围：0~59

*byBeginTimeSec*

定时模式开始时间（秒），取值范围：0~59

*byEndTimeMin*

定时模式结束时间（分），取值范围：0~59

*byEndTimeSec*

定时模式结束时间（秒），取值范围：0~59

*byAlarmTrigState*

报警输入触发状态：0-白天，1-夜晚

## 7.19 NET\_DVR\_DECODERCFG\_V30:云台解码器(RS485)参数

```
struct{
    DWORD    dwSize;
    DWORD    dwBaudRate;
    BYTE     byDataBit;
    BYTE     byStopBit;
    BYTE     byParity;
    BYTE     byFlowcontrol;
    WORD     wDecoderType;
    WORD     wDecoderAddress;
    BYTE     bySetPreset[MAX_PRESET_V30];
    BYTE     bySetCruise[MAX_CRUISE_V30];
    BYTE     bySetTrack[MAX_TRACK_V30];
}NET_DVR_DECODERCFG_V30, *LPNET_DVR_DECODERCFG_V30;
```

### Members

*dwSize*

结构体大小

*dwBaudRate*

波特率(bps), 0-50, 1-75, 2-110, 3-150, 4-300, 5-600, 6-1200, 7-2400, 8-4800, 9-9600, 10-19200, 11-38400, 12-57600, 13-76800, 14-115.2k

*byDataBit*

数据有几位：0-5 位，1-6 位，2-7 位，3-8 位

*byStopBit*

停止位：0-1 位，1-2 位

*byParity*

是否校验：0-无校验，1-奇校验，2-偶校验

*byFlowcontrol*

是否流控：0-无，1-软流控,2-硬流控

*wDecoderType*

解码器类型，通过 [NET\\_DVR\\_GetPTZProtocol](#) 获取，该值对应于结构 NET\_DVR\_PTZ\_PROTOCOL 中的 dwType。

*wDecoderAddress*

解码器地址：[0,255]

*bySetPreset*

预置点是否设置：0-没有设置，1-设置

*bySetCruise*

巡航是否设置：0-没有设置，1-设置

*bySetTrack*

花样扫描是否设置：0-没有设置，1-设置

**Remarks**

在早前的设备中规定了一系列云台协议，但在后期的设备仅保留一部分常用的协议，所以在配置解码器类型时必须调用 [NET\\_DVR\\_GetPTZProtocol](#) 获取当前设备支持的云台协议。

**7.20 NET\_DVR\_DEFOGCFG:透雾参数**

```
struct{
    BYTE    byMode;
    BYTE    byLevel;
    BYTE    byRes[6];
}NET_DVR_DEFOGCFG, *LPNET_DVR_DEFOGCFG;
```

**Members***byMode*

透雾模式：0-不启用，1-自动模式，2-常开模式

*byLevel*

透雾等级，取值范围：0~100

*byRes*

保留

**7.21 NET\_DVR\_DEVICECFG\_V40:设备参数**

```
struct{
    DWORD    dwSize;
    BYTE    sDVRName[NAME_LEN];
    DWORD    dwDVRID;
    DWORD    dwRecycleRecord;
    BYTE    sSerialNumber[SERIALNO_LEN];
    DWORD    dwSoftwareVersion;
    DWORD    dwSoftwareBuildDate;
    DWORD    dwDSPSoftwareVersion;
    DWORD    dwDSPSoftwareBuildDate;
    DWORD    dwPanelVersion;
    DWORD    dwHardwareVersion;
    BYTE    byAlarmInPortNum;
    BYTE    byAlarmOutPortNum;
    BYTE    byRS232Num;
    BYTE    byRS485Num;
    BYTE    byNetworkPortNum;
    BYTE    byDiskCtrlNum;
```

```

BYTE    byDiskNum;
BYTE    byDVRType;
BYTE    byChanNum;
BYTE    byStartChan;
BYTE    byDecordChans;
BYTE    byVGANum;
BYTE    byUSBNum;
BYTE    byAuxoutNum;
BYTE    byAudioNum;
BYTE    byIPChanNum;
BYTE    byZeroChanNum;
BYTE    bySupport;
BYTE    byEsataUseage;
BYTE    byIPCPlug;
BYTE    byStorageMode;
BYTE    bySupport1;
WORD    wDevType;
BYTE    byDevTypeName[24];
BYTE    byRes2[16];
}NET_DVR_DEVICECFG_V40,*LPNET_DVR_DEVICECFG_V40;

```

### Members

#### *dwSize*

结构体大小

#### *sDVRName*

设备名称

#### *dwDVRID*

设备 ID 号，用于遥控器，v1.4 的设备号范围为(0-99), v1.5 及以上版本的设备号为(0-255)

#### *dwRecycleRecord*

是否循环录像：0—不是；1—是

### 以下参数不可更改

#### *sSerialNumber*

设备序列号

#### *dwSoftwareVersion*

软件版本号，V3.0 以上版本支持的设备最高 8 位为主版本号，次高 8 位为次版本号，低 16 位为修复版本号；V3.0 以下版本支持的设备高 16 位表示主版本，低 16 位表示次版本

#### *dwSoftwareBuildDate*

软件生成日期，0xYYYYMMDD

#### *dwDSPSoftwareVersion*

DSP 软件版本，高 16 位是主版本，低 16 位是次版本

#### *dwDSPSoftwareBuildDate*

DSP 软件生成日期，0xYYYYMMDD

#### *dwPanelVersion*

前面板版本，高 16 位是主版本，低 16 位是次版本

#### *dwHardwareVersion*

硬件版本，高 16 位是主版本，低 16 位是次版本

*byAlarmInPortNum*

设备报警输入个数

*byAlarmOutPortNum*

设备报警输出个数

*byRS232Num*

设备 232 串口个数

*byRS485Num*

设备 485 串口个数

*byNetworkPortNum*

网络口个数

*byDiskCtrlNum*

硬盘控制器个数

*byDiskNum*

硬盘个数

*byDVRType*

设备类型

宏定义	宏定义值	设备类型
<b>TRAFFIC_TS_SERVER</b>	<b>210</b>	终端服务器
<b>TRAFFIC_VAR</b>	<b>211</b>	视频分析记录仪
<b>TRAFFIC_ECT</b>	<b>1400</b>	出入口终端服务器
<b>TRAFFIC_PARKING_SERVER</b>	<b>1401</b>	停车场服务器

*byChanNum*

设备模拟通道个数

*byStartChan*

起始通道号

*byDecordChans*

设备解码路数

*byVGANum*

VGA 口的个数

*byUSBNum*

USB 口的个数

*byAuxoutNum*

辅口的个数

*byAudioNum*

语音口的个数

*byIPChanNum*

最大数字通道

*byZeroChanNum*

零通道编码个数

*bySupport*

能力，位与结果为 0 表示不支持，1 表示支持

*bySupport & 0x1*，表示是否支持智能搜索

bySupport & 0x2, 表示是否支持备份  
 bySupport & 0x4, 表示是否支持压缩参数能力获取  
 bySupport & 0x8, 表示是否支持双网卡  
 bySupport & 0x10, 表示支持远程 SADP  
 bySupport & 0x20, 表示支持 Raid 卡功能  
 bySupport & 0x40, 表示支持 IPSAN 搜索  
 bySupport & 0x80, 表示支持 rtp over rtsp

**byEsataUseage**

Esata 的默认用途, 0-默认备份, 1-默认录像

**byIPCPlug**

0-不支持即插即用, 1-支持即插即用

**byStorageMode**

0-盘组模式, 1-磁盘配额

**bySupport1**

能力集扩充, 位与结果为 0 表示不支持, 1 表示支持

bySupport1 & 0x1, 表示是否支持 snmp v30

bySupport1 & 0x2, 支持区分回放和下载

**wDevType**

设备型号

**byDevTypeName**

设备型号名称

**byRes2**

保留, 置为 0

**Remarks**

如果 byDVRType 是 0, 则接口中解析 wDevType 作为设备型号, 设备端同时将设备型号的名称传过来。

如果 byDVRType 不是 0, 则接口中将不解析 wDevType 及 byDevTypeName, 使用已有的设备型号及名称对 byDVRType、wDevType、byDevTypeName 进行填充, 其中 byDVRType=wDevType。

建议开发时使用 wDevType、byDevTypeName, 而不要使用 byDVRType, sdk 内部兼容。

## 7.22 NET\_DVR\_DEVICEINFO\_V30:设备参数

```
struct{
  BYTE      sSerialNumber[SERIALNO_LEN];
  BYTE      byAlarmInPortNum;
  BYTE      byAlarmOutPortNum;
  BYTE      byDiskNum;
  BYTE      byDVRType;
  BYTE      byChanNum;
  BYTE      byStartChan;
  BYTE      byAudioChanNum;
  BYTE      byIPChanNum;
  BYTE      byZeroChanNum;
  BYTE      byMainProto;
  BYTE      bySubProto;
```

```

BYTE    bySupport;
BYTE    bySupport1;
BYTE    bySupport2;
WORD    wDevType;
BYTE    bySupport3;
BYTE    byMultiStreamProto;
BYTE    byStartDChan;
BYTE    byStartDTalkChan;
BYTE    byHighDChanNum;
BYTE    byRes2[11];

```

```
}NET_DVR_DEVICEINFO_V30,*LPNET_DVR_DEVICEINFO_V30;
```

## Members

*sSerialNumber*

序列号

*byAlarmInPortNum*

报警输入个数

*byAlarmOutPortNum*

报警输出个数

*byDiskNum*

硬盘个数

*byDVRType*

设备类型，详见“Remarks”说明

*byChanNum*

设备模拟通道个数，数字（IP）通道最大个数为  $\text{byIPChanNum} + \text{byHighDChanNum} * 256$

*byStartChan*

模拟通道的起始通道号，从 1 开始。IP 通道的起始通道号见下面参数 *byStartDChan*

*byAudioChanNum*

设备语音通道数

*byIPChanNum*

设备最大数字通道个数

*byZeroChanNum*

零通道编码个数

*byMainProto*

主码流传输协议类型：0-private，1-rtsp

*bySubProto*

子码流传输协议类型：0-private，1-rtsp

*bySupport*

能力，位与结果为 0 表示不支持，1 表示支持

*bySupport & 0x1*，表示是否支持智能搜索

*bySupport & 0x2*，表示是否支持备份

*bySupport & 0x4*，表示是否支持压缩参数能力获取

*bySupport & 0x8*，表示是否支持双网卡

*bySupport & 0x10*，表示支持远程 SADP

*bySupport & 0x20*，表示支持 Raid 卡功能

bySupport & 0x40, 表示支持 IPSAN 目录查找

bySupport & 0x80, 表示支持 rtp over rtsp

#### bySupport1

能力集扩充, 位与结果为 0 表示不支持, 1 表示支持

bySupport1 & 0x1, 表示是否支持 snmp v30

bySupport1 & 0x2, 表示是否支持区分回放和下载

bySupport1 & 0x4, 表示是否支持布防优先级

bySupport1 & 0x8, 表示智能设备是否支持布防时间段扩展

bySupport1 & 0x10, 表示是否支持多磁盘数 (超过 33 个)

bySupport1 & 0x20, 表示是否支持 rtsp over http

bySupport1 & 0x40, 表示是否支持延时预览

bySupport1 & 0x80, 表示是否支持车牌新报警信息

#### bySupport2

能力集扩充, 位与结果为 0 表示不支持, 1 表示支持

bySupport2 & 0x1, 表示解码器是否支持通过 URL 取流解码

#### wDevType

设备型号, 详见“Remarks”说明

#### bySupport3

能力集扩展, 位与结果: 0- 不支持, 1- 支持

bySupport3 & 0x1, 表示是否支持多码流

bySupport3 & 0x4, 表示是否支持按组配置, 具体包含通道图像参数、报警输入参数、IP 报警输入/输出接入参数、用户参数、设备工作状态、JPEG 抓图、定时和时间抓图、硬盘盘组管理等

bySupport3 & 0x20, 表示是否支持通过 DDNS 域名解析取流

#### byMultiStreamProto

是否支持多码流, 按位表示, 位与结果: 0-不支持, 1-支持

byMultiStreamProto & 0x1, 表示是否支持码流 3

byMultiStreamProto & 0x2, 表示是否支持码流 4

byMultiStreamProto & 0x40, 表示是否支持主码流

byMultiStreamProto & 0x80, 表示是否支持子码流

#### byStartDChan

起始数字通道号, 0 表示无数字通道, 比如 DVR 或 IPC

#### byStartDTalkChan

起始数字对讲通道号, 区别于模拟对讲通道号, 0 表示无数字对讲通道

#### byHighDChanNum

数字通道个数, 高 8 位

#### byRes2

保留, 置为 0

### Remarks

如果 byDVRTYPE 是 0, 则接口中解析 wDevType 作为设备型号; 如果 byDVRTYPE 非 0, 则接口中 byDVRTYPE 和 wDevType 值相等, 都是 byDVRTYPE。推荐使用 wDevType 作为设备类型。设备类型定义如下所示:

宏定义	宏定义值	设备类型
TRAFFIC_TS_SERVER	210	终端服务器
TRAFFIC_VAR	211	视频分析记录仪

TRAFFIC_ECT	1400	出入口终端服务器
TRAFFIC_PARKING_SERVER	1401	停车场服务器

## 7.23 NET\_DVR\_DEVICEINFO\_V40:设备参数

```

struct{
    NET_DVR_DEVICEINFO_V30    struDeviceV30;
    BYTE                      bySupportLock;
    BYTE                      byRetryLoginTime;
    BYTE                      byPasswordLevel;
    BYTE                      byRes1;
    DWORD                    dwSurplusLockTime;
    BYTE                      byCharEncodeType;
    BYTE                      byRes2[255];
}NET_DVR_DEVICEINFO_V40,*LPNET_DVR_DEVICEINFO_V40;

```

### Members

#### *struDeviceV30*

设备参数

#### *bySupportLock*

设备是否支持锁定功能，*bySupportLock* 为 1 时，*dwSurplusLockTime* 和 *byRetryLoginTime* 有效

#### *byRetryLoginTime*

剩余可尝试登陆的次数，用户名、密码错误时，此参数有效

#### *byPasswordLevel*

密码安全等级：0- 无效，1- 默认密码，2- 有效密码，3- 风险较高的密码，当管理员用户的密码为出厂默认密码（12345）或者风险较高的密码时，建议上层客户端提示用户更改密码

#### *byRes1*

保留，置为 0

#### *dwSurplusLockTime*

剩余时间，单位：秒，用户锁定时此参数有效。在锁定期间，用户尝试登陆，不管用户名密码输入对错，设备锁定剩余时间重新恢复到 30 分钟

#### *byCharEncodeType*

字符编码类型（SDK 所有接口返回的字符串编码类型，透传接口除外）：0- 无字符编码信息(老设备)，1- GB2312(简体中文)，2- GBK，3- BIG5(繁体中文)，4- Shift\_JIS(日文)，5- EUC-KR(韩文)，6- UTF-8，7- ISO8859-1，8- ISO8859-2，9- ISO8859-3，...，依次类推，21- ISO8859-15(西欧)

#### *byRes2*

保留，置为 0

### Remarks

将密码输入分为数字(0~9)、小写字母(a~z)、大写字母(A~Z)、特殊符号 (:\除外) 4 类，等级分为 4 个等级，如下所示：

- 等级 0（风险密码）：密码长度小于 8 位，或者只包含 4 类字符中的任意一类，或者密码与用户名一样，或者密码是用户名的倒写。例如：12345、abcdef。
- 等级 1（弱密码）：包含两类字符，且组合为（数字+小写字母）或（数字+大写字母），且长度大于

等于 8 位。例如：abc12345、123ABCDEF。

- 等级 2（中密码）：包含两类字符，且组合不能为（数字+小写字母）和（数字+大写字母），且长度大于等于 8 位。例如：12345\*\*\*++、ABCDabcd。
- 等级 3（强密码）：包含三类字符及以上，且长度大于等于 8 位。例如：Abc12345、abc12345++。

## 7.24 NET\_DVR\_DIRECTION:方向信息

```
struct{
    NET_VCA_POINT    struStartPoint;
    NET_VCA_POINT    struEndPoint;
}NET_DVR_DIRECTION,*LPNET_DVR_DIRECTION;
```

### Members

*struStartPoint*

方向起始点

*struEndPoint*

方向结束点

## 7.25 NET\_DVR\_DISKSTATE:硬盘信息

```
struct{
    DWORD    dwVolume;
    DWORD    dwFreeSpace;
    DWORD    dwHardDiskStatic;
}NET_DVR_DISKSTATE,*LPNET_DVR_DISKSTATE;
```

### Members

*dwVolume*

硬盘容量，单位：MB

*dwFreeSpace*

硬盘剩余空间，单位：MB

*dwHardDiskStatic*

硬盘的状态：0-活动，1-休眠，2-异常，3-休眠硬盘出错，4-未格式化，5-未连接状态(网络硬盘)，6-硬盘正在格式化

## 7.26 NET\_DVR\_ELECTRONICSTABILIZATION:电子防抖参数

```
struct{
    BYTE    byEnable;
    BYTE    byLevel;
    BYTE    byRes[6];
}NET_DVR_ELECTRONICSTABILIZATION,*LPNET_DVR_ELECTRONICSTABILIZATION;
```

### Members

*byMode*

电子防抖使能：0- 不启用，1- 启用

*byLevel*

电子防抖等级，取值范围：0~100

*byRes*

保留

## 7.27 NET\_DVR\_ENISSUED\_DATADEL:终端出入口控制(清空下发卡片信息)结构体

```
struct{
    DWORD    dwSize;
    BYTE     byDevCtrlCode;
    BYTE     byRes[27];
}NET_DVR_ENISSUED_DATADEL, *LPNET_DVR_ENISSUED_DATADEL;
```

### Members

*dwSize*

结构体大小

*byDevCtrlCode*

出入口控制机拨码地址，用于区分设备，取值范围：1~255

*byRes*

保留，置为 0

## 7.28 NET\_DVR\_ENTRANCEDEV\_COMMAND:终端出入口控制（数据下发）结构体

```
struct{
    DWORD    dwSize;
    BYTE     byDevCtrlCode;
    BYTE     byManualIssuedData;
    BYTE     byRes[62];
}NET_DVR_ENTRANCEDEV_COMMAND, *LPNET_DVR_ENTRANCEDEV_COMMAND;
```

### Members

*dwSize*

结构体大小

*byDevCtrlCode*

出入口控制机拨码地址，用于区分设备，取值范围：1~255

*byManualIssuedData*

是否启动手动数据下发：0-关闭，1-开启

*byRes*

保留，置为 0

## 7.29 NET\_DVR\_ETHERNET\_V30:以太网配置

```
struct{
    NET_DVR_IPADDR    struDVRIP;
    NET_DVR_IPADDR    struDVRIPMask;
    DWORD             dwNetInterface;
    WORD              wDVRPort;
    WORD              wMTU;
    BYTE              byMACAddr[MACADDR_LEN];
    BYTE              byRes[2];
}NET_DVR_ETHERNET_V30, *LPNET_DVR_ETHERNET_V30;
```

### Members

#### *struDVRIP*

设备 IP 地址

#### *struDVRIPMask*

设备 IP 地址掩码

#### *dwNetInterface*

网络接口：1-10MBase-T；2-10MBase-T 全双工；3-100MBase-TX；4-100M 全双工；5-10M/100M/1000M 自适应；6-1000M 全双工

#### *wDVRPort*

设备端口号

#### *wMTU*

MTU 设置，默认 1500

#### *byMACAddr*

设备物理地址

#### *byRes*

保留

### Remarks

MTU 的设置范围为 500-9676，若 MTU 设置过小客户端将无法注册到设备，并且客户端预览、回放、配置参数也会失败。

## 7.30 NET\_DVR\_EXCEPTION\_V30:异常参数

```
struct{
    DWORD             dwSize;
    NET_DVR_HANDLEEXCEPTION_V30 struExceptionHandleType[MAX_EXCEPTIONNUM_V30];
}NET_DVR_EXCEPTION_V30,*LPNET_DVR_EXCEPTION_V30;
```

### Members

#### *dwSize*

结构体大小

#### *struExceptionHandleType*

异常信息处理方式：

数组 0—硬盘满

- 数组 1—硬盘出错
- 数组 2—网线断
- 数组 3—IP 地址冲突
- 数组 4—非法访问
- 数组 5—输入/输出视频制式不匹配
- 数组 6—视频信号异常
- 数组 7—录像异常

### 7.31 NET\_DVR\_EXPOSURE:CCD 曝光控制参数

```
struct{
    BYTE        byExposureMode;
    BYTE        byAutoApertureLevel;
    BYTE        byRes[2];
    DWORD       dwVideoExposureSet;
    DWORD       dwExposureUserSet;
    DWORD       dwRes;
}NET_DVR_EXPOSURE, *LPNET_DVR_EXPOSURE;
```

#### Members

##### *byExposureMode*

0-手动曝光, 1-自动曝光

##### *byAutoApertureLevel*

自动光圈灵敏度, 取值范围: 0~10

##### *byRes*

保留

##### *dwVideoExposureSet*

自定义视频曝光时间 (单位 us), 自动曝光时该值为曝光最慢值

##### *dwExposureUserSet*

自定义曝光时间。在智能高清网络摄像机上应用及 CCD 模式时, 是指抓拍快门速度, (单位 us)

##### *dwRes*

保留

### 7.32 NET\_DVR\_EXTERNAL\_CONTROL\_ALARM:指示灯外控报警信息

```
struct{
    DWORD       dwSize;
    DWORD       dwChannelNo;
    NET\_DVR\_LAMP\_STATE struLampStateCtrl;
    NET\_DVR\_TIME struExternalBeginTime;
    BYTE        byRes1[64];
}NET_DVR_EXTERNAL_CONTROL_ALARM, *LPNET_DVR_EXTERNAL_CONTROL_ALARM;
```

#### Members

##### *dwSize*

结构体大小

*dwChannelNo*

通道号

*struLampStateCtrl*

指示灯外控状态

*struExternalBeginTime*

外控开始时间

*byRes*

保留，置为 0

### 7.33 NET\_DVR\_EXTERNAL\_DEVCFG:ITS 外接设备信息

```

struct{
    DWORD    dwSize;
    BYTE     byExternalDevTpye;
    BYTE     byRes2[3];
    char     sDevName[MAX_DEVNAME_LEN];
    union{
        BYTE     byUnionLen[656];
        struct{
            NET\_DVR\_IPADDR    struDevIP;
            WORD     wDevPort;
            BYTE     byRes[510];
        }struRrReader;
        struct{
            BYTE     byRs485No;
            BYTE     byDevCtrlCode;
            BYTE     byAutoIssuedData;
            BYTE     byOfflineDetEnable;
            BYTE     byDetCycle;
            BYTE     byRes[651];
        }struGateway;
        struct{
            BYTE     byExternalMode;
            BYTE     byRes[3];
            NET\_DVR\_IPADDR    struDevIP;
            WORD     wDevPort;
            BYTE     byRs485No;
            BYTE     byDevCtrlCode;
            BYTE     byCtrlCardType;
            BYTE     byLedScreenType;
            BYTE     byLedScreenUse;
            BYTE     byLedDisplayMode;
            char     sLedCustomInfo[MAX_LED_INFO];
        }
    }
}

```

```

        DWORD          dwLedScreenH;
        DWORD          dwLedScreenW;
        BYTE           byRes1[236];
    }struLed;
}uExternalDevInfo;
BYTE    byRes[128];
}NET_DVR_EXTERNAL_DEVCFG, *LPNET_DVR_EXTERNAL_DEVCFG;

```

## Members

*dwSize*

结构体大小

*byExternalDevType*

外接设备类型：0- 其他(联合体 *unionAddr* 可不赋值)，1- 远距离读头设备(对应联合体里的 *struRrReader*)，2- 出入口控制机(对应联合体里的 *struGateway*)，3- LED 屏(对应联合体里的 *struLed*)

*byRes2*

保留

*sDevName*

设备名称

***unionAddr* 为外接设备信息联合体**

*byUnionLen*

联合体大小为 656 字节

***struRrReader* 为远距离读头设备信息结构体**

*struDevIP*

设备 IP 地址

*wDevPort*

设备端口

*byRes*

保留

***struGateway* 为出入口控制机信息结构体**

*byRs485No*

RS485 编号(外接设备类型一样，该编号可以重复；类型不一样，该编号互斥)

*byDevCtrlCode*

拨码地址，用于区分设备，取值范围：1~255

*byAutoIssuedData*

是否自动下发卡片数据：0-不自动下发，1-自动下发

*byOfflineDetEnable*

离线检测：0-关闭，1-开启。离线检测开启时，如果检测到设备在线，则由终端匹配卡片信息，并发送道闸控制指令给出入口控制机；否则由出入口控制机自身匹配控制道闸

*byDetCycle*

检测周期，取值范围：0~60，单位：s

*byRes*

保留

***struLed* 为 LED 屏信息结构体**

*byExternalMode*

LED 屏接口类型：0-RS485，1-网络

*byRes*

保留

*struDevIP*

DEV IP 地址 144 (在 *byExternalMode* == 1 时生效)

*wDevPort*

设备端口(在 *byExternalMode* == 1 时生效)

*byRs485No*

RS485 编号(外接设备类型一样, 该编号可以重复; 类型不一样, 该编号互斥)

*byDevCtrlCode*

拨码地址, 用于区分设备, 取值范围: 1~255

*byCtrlCardType*

控制卡类型: 0-保留(无效值), 1- EQ2013, LED 屏通过控制卡控制显示的, 不同的控制卡类型支持的显示方式不同

*byLedScreenType*

屏类型: 0-单色屏, 1-双色屏

*byLedScreenUse*

屏用途: 0-其它, 1-区位屏, 2-提示屏, 3-收费屏

*byLedDisplayMode*

屏幕字体显示方式: 1- 立即显示, 2- 左移, 3- 右移, 4- 上移, 5- 下移

*sLedContomInfo*

屏幕字体显示信息

*dwLedScreenH*

显示屏组成高度上需要的 LED 点数

*dwLedScreenW*

显示屏组成宽度上需要的 LED 点数

*byRes1*

保留

*byRes*

保留

### Remarks

LED 屏是由控制卡控制的, LED 屏可以根据项目实际使用更换大小, 此时需要终端通知控制卡目前使用的屏宽(*dwLedScreenH*)和屏高(*dwLedScreenW*)。

## 7.34 NET\_DVR\_EXTERNAL\_DEVCOND:ITS 外接设备获取信息条件

```

struct{
    DWORD    dwSize;
    BYTE     byExternalDevTpye;
    BYTE     byRelativeIndex;
    BYTE     byRes[30];
}NET_DVR_EXTERNAL_DEVCOND, *LPNET_DVR_EXTERNAL_DEVCOND;

```

### Members

*dwSize*

结构体大小

**byExternalDevTpye**

外接设备类型：0- 保留，1- 远距离读头设备，2- 出入口控制机，3- LED 屏，0xff- 全部信息（长连接获取时使用）

**byRelativeIndex**

相对索引：0-保留；针对 LED 屏的类型，相对索引值为 1~255，针对其他类型（远距离读头设备、出入口控制机）相对索引值为 1~8；0xff-全部信息（长连接获取时使用）

**byRes**

保留，置为

**7.35 NET\_DVR\_EXTERNAL\_DEVSTATUS:外接设备状态信息**

```
struct{
    DWORD    dwSize;
    char     sDevName[MAX_DEVNAME_LEN];
    BYTE     byExternalDevTpye;
    BYTE     byRelativeIndex;
    BYTE     byOnline;
    BYTE     byRes[125];
}NET_DVR_EXTERNAL_DEVSTATUS, *LPNET_DVR_EXTERNAL_DEVSTATUS;
```

**Members****dwSize**

结构体大小

**sDevName**

设备名称

**byExternalDevTpye**

外接设备类型：0- 保留，1- 远距离读头设备，2- 出入口控制机，3- LED 屏

**byRelativeIndex**

相对索引：0-保留；针对 LED 屏的类型，相对索引值为 1~255，针对其他类型（远距离读头设备、出入口控制机）相对索引值为 1~8

**byRes**

保留，置为 0

**7.36 NET\_DVR\_EXTERNAL\_LAMP\_CTRL\_MODE:外接灯控制模式**

```
struct{
    NET\_DVR\_PARKINFO    struParkInfo[MAX_PARKING_NUM];
    BYTE                byRes[32];
}NET_DVR_EXTERNAL_LAMP_CTRL_MODE, *LPNET_DVR_EXTERNAL_LAMP_CTRL_MODE;
```

**Members****struParkInfo**

外接灯的情况下车位信息，一个通道最大 4 个车位（从左到右车位对应数组 0~3）

**byRes**

保留

## 7.37 NET\_DVR\_FORENSICS\_MODE: 交通违章取证方式配置

```
struct{
    DWORD    dwSize;
    BYTE     byMode;
    BYTE     byRes[23];
}NET_DVR_FORENSICS_MODE, *LPNET_DVR_FORENSICS_MODE;
```

### Members

#### *dwSize*

结构体大小

#### *byMode*

取证方式：0- 手动取证，1- 自动取证

#### *byRes*

保留

### Remarks

设备必须接入球机才支持违停取证分析。交通违章取证模式分为两种：

- 自动模式：分析仪对视频进行分析，如果检测到目标则驱动球机转动，如果符合条件则上传报警。也可以使用手动定位控制命令指定目标。
- 手动模式：分析仪对视频进行分析，即使检测到目标也不会驱动球机转动，除非使用手动定位控制命令 NET\_DVR\_CONTROL\_PTZ\_MANUALTRACE（对应接口：[NET\\_DVR\\_RemoteControl](#)）指定目标，让分析仪对目标进行分析。使用手动模式，主要是为了降低误报。

## 7.38 NET\_DVR\_GAIN: 增益参数

```
struct{
    BYTE     byGainLevel;
    BYTE     byGainUserSet;
    BYTE     byRes[2];
    DWORD    dwMaxGainValue;
}NET_DVR_GAIN, *LPNET_DVR_GAIN;
```

### Members

#### *byGainLevel*

增益，单位 dB，取值范围[0,100]

#### *byGainUserSet*

用户自定义增益，单位 dB，取值范围[0,100]，对于智能高清网络摄像机，是 CCD 模式下的抓拍增益

#### *byRes*

保留，置为 0

#### *dwMaxGainValue*

最大增益值，单位 dB

### 7.39 NET\_DVR\_GAMMACORRECT:Gamma 校正配置参数

```
struct{
    BYTE        byGammaCorrectionEnabled;
    BYTE        byGammaCorrectionLevel;
    BYTE        byRes[6];
}NET_DVR_GAMMACORRECT, *LPNET_DVR_GAMMACORRECT;
```

#### Members

##### *byGammaCorrectionEnabled*

Gamma 校正是否启用，0-不启用，1-启用

##### *byGammaCorrectionLevel*

0-100

##### *byRes*

保留，置为 0

### 7.40 NET\_DVR\_GET\_STREAM\_UNION:取流方式联合体

```
union{
    NET\_DVR\_IPCHANINFO        struChanInfo;
    NET_DVR_PU_STREAM_CFG    struPUStream;
    NET_DVR_IPSERVER_STREAM  struIPServerStream;
    NET_DVR_DDNS_STREAM_CFG  struDDNSStream;
    NET_DVR_PU_STREAM_URL    struStreamUrl;
    NET_DVR_HKDDNS_STREAM    struHkDDNSStream;
}NET_DVR_GET_STREAM_UNION,*LPNET_DVR_GET_STREAM_UNION;
```

#### Members

##### *struChanInfo*

直接从设备取流的 IP 通道信息

##### *struPUStream*

通过流媒体从设备取流

##### *struIPServerStream*

通过 IPServer 获得 IP 地址后取流

##### *struDDNSStream*

通过 IPServer 找到设备，再通过流媒体取设备的码流

##### *struStreamUrl*

通过 URL 从流媒体取流

##### *struHkDDNSStream*

通过 hiDDNS 连接设备然后从设备取流

#### Remarks

智能交通设备仅支持直接从设备取流(*struChanInfo*)的模式。其他模式不支持，其相关结构体在本文档里不做介绍，请参见头文件。

## 7.41 NET\_DVR\_HANDLEEXCEPTION\_V30:报警和异常处理

```
struct{
    DWORD    dwHandleType;
    BYTE     byRelAlarmOut[MAX_ALARMOUT_V30];
}NET_DVR_HANDLEEXCEPTION_V30, *LPNET_DVR_HANDLEEXCEPTION_V30;
```

### Members

#### *dwHandleType*

处理方式:

0x00: 无响应

0x01: 监视器上警告

0x02: 声音警告

0x04: 上传中心

0x08: 触发报警输出

0x10: Jpeg 抓图并上传 Email

0x20: 无线声光报警器联动

0x200: 抓图并上传 ftp

#### *byRelAlarmOut*

报警触发的输出通道, 0-不触发, 1-触发输出, 按位表示输出通道, 例如 `byRelAlarmOut[0]==1` 表示触发输出通道 1, `byRelAlarmOut[1]==1` 表示触发输出通道 2, 依此类推

## 7.42 NET\_DVR\_HDCFG:设备硬盘配置结构体

```
struct{
    DWORD                dwSize;
    DWORD                dwHDCount;
    NET\_DVR\_SINGLE\_HD    struHDInfo[MAX_DISKNUM_V30];
}NET_DVR_HDCFG, *LPNET_DVR_HDCFG;
```

### Members

#### *dwSize*

结构体大小

#### *dwHDCount*

硬盘数, 该参数只能获取, 不支持设置

#### *struHDInfo*

硬盘信息参数

### Remarks

本结构体中的 `dwHDCount` 参数是指设备本地的硬盘数, 因此只能获取该信息, 不能设置。对硬盘的信息进行设置后需要重启设备才生效。

## 7.43 NET\_DVR\_INLAY\_LAMP\_CTRL\_MODE:内置灯控制模式

```
struct{
```

```

NET\_DVR\_PARK\_INLAY\_SUBINFO    struLampStateCtrl[MAX_PARKING_STATUS];
BYTE                               byRes[96];
}NET_DVR_INLAY_LAMP_CTRL_MODE, *LPNET_DVR_INLAY_LAMP_CTRL_MODE;

```

### Members

#### *struLampStateCtrl*

内置灯的情况下车位状态，数组 0- 无车，数组 1- 有车，数组 2- 压线(优先级最高)，数组 3- 特殊车位

#### *byRes*

保留

## 7.44 NET\_DVR\_IPADDR:IP 地址

```

struct{
    char    slpV4[16];
    BYTE    slpV6[128];
}NET_DVR_IPADDR, *LPNET_DVR_IPADDR;

```

### Members

#### *slpV4*

设备 IPv4 地址

#### *slpV6*

设备 IPv6 地址

## 7.45 NET\_DVR\_IPC\_PROTO\_LIST:IPC 协议列表

```

struct{
    DWORD    dwSize;
    DWORD    dwProtoNum;
    NET\_DVR\_PROTO\_TYPE    struProto[IPC_PROTOCOL_NUM];
    BYTE     byRes[8];
}NET_DVR_IPC_PROTO_LIST,*LPNET_DVR_IPC_PROTO_LIST;

```

### Members

#### *dwSize*

结构体大小

#### *dwProtoNum*

有效的 IPC 协议个数

#### *struProto*

有效的 IPC 协议的参数结构

#### *byRes*

保留，置为 0

## 7.46 NET\_DVR\_IPCHANINFO:IP 通道信息

```
struct{
    BYTE    byEnable;
    BYTE    byIPID;
    BYTE    byChannel;
    BYTE    byIPIDHigh;
    BYTE    byRes[32];
}NET_DVR_IPCHANINFO, *LPNET_DVR_IPCHANINFO;
```

### Members

#### byEnable

IP 通道在线状态，是一个只读的属性；0 表示 HDVR 或者 NVR 设备的数字通道连接对应的 IP 设备失败，该通道不在线；1 表示连接成功，该通道在线

#### byIPID

IP 设备 ID 的低 8 位， $byIPID = iDevID \% 256$

#### byChannel

IP 设备的通道号，例如设备 A（HDVR 或者 NVR 设备）的 IP 通道 01，对应的是设备 B 里的通道 04，则  $byChannel=4$ 。

#### byIPIDHigh

IP 设备 ID 的高 8 位， $byIPIDHigh = iDevID / 256$

#### byRes

保留，置为 0

### Remarks

iDevID 为设备 ID 号， $iDevID = byIPIDHigh * 256 + byIPID$ 。通过 iDevID 值查找具体的设备信息 `struIPDevInfo`（结构体 `NET_DVR_IPPARACFG_V40` 的数组参数），与设备信息数组下标（`iDevInfoIndex`）换算关系为： $iDevID = iDevInfoIndex + iGroupNO * 64 + 1$ 。

## 7.47 NET\_DVR\_IPDEVINFO\_V31:IP 设备信息

```
struct{
    BYTE    byEnable;
    BYTE    byProType;
    BYTE    byEnableQuickAdd;
    BYTE    byRes1;
    BYTE    sUserName[NAME_LEN];
    BYTE    sPassword[PASSWD_LEN];
    BYTE    byDomain[MAX_DOMAIN_NAME];
    NET\_DVR\_IPADDR struIP;
    WORD    wDVRPort;
    BYTE    byRes2[34];
}NET_DVR_IPDEVINFO_V31, *LPNET_DVR_IPDEVINFO_V31;
```

### Members

#### byEnable

该 IP 设备是否启用

*byProType*

协议类型(默认为私有协议), 0- 私有协议, 1- 松下协议, 2- 索尼, 更多协议通过 [NET\\_DVR\\_GetIPCProtoList](#) 获取。

*byEnableQuickAdd*

0-不支持快速添加; 1-使用快速添加

*byRes1*

保留, 置为 0

*sUserName*

用户名

*sPassword*

密码

*byDomain*

设备域名

*struIP*

IP 地址

*wDVRPort*

端口号

*byRes2*

保留, 置为 0

#### Remarks

当某个 IP 设备参数对应的所有 IP 通道被删除, 即 IP 通道资源的中所有 IP 通道参数的 IPID 减 1 没有与该 IP 设备参数的下标值相对应的时候, 设备本地的该 IP 设备参数将被删除。

在该结构体中, 设备域名为空, ipv4 地址有效时, 使用 ipv4 地址去连接设备; ipv4 和设备域名都为空, ipv6 地址有效时, 使用 ipv6 去连接设备。

## 7.48 NET\_DVR\_IPPARACFG\_V40:IP 设备资源及 IP 通道资源配置参数

```
struct{
    DWORD                dwSize;
    DWORD                dwGroupNum;
    DWORD                dwAChanNum;
    DWORD                dwDChanNum;
    DWORD                dwStartDChan;
    BYTE                 byAnalogChanEnable[MAX_CHANNUM_V30];
    NET_DVR_IPDEVINFO_V31 struIPDevInfo[MAX_IP_DEVICE_V40];
    NET_DVR_STREAM_MODE struStreamMode[MAX_CHANNUM_V30];
    BYTE                 byRes2[20];
}NET_DVR_IPPARACFG_V40, *LPNET_DVR_IPPARACFG_V40;
```

#### Members

*dwSize*

结构体大小

*dwGroupNum*

设备支持的总组数 (只读)。若设备支持的组数大于 1, NET\_DVR\_GetDVRConfig (或者

NET\_DVR\_SetDVRConfig) 获取 (或设置) 相关通道参数需要按照组数调用多次命令分别获取 (或设置) 各组通道参数, 此时接口中 IChannel 对应组号。

*dwAChanNum*

最大模拟通道个数 (只读)

*dwDChanNum*

数字通道个数 (只读)

*dwStartDChan*

起始数字通道 (只读)

*byAnalogChanEnable*

模拟通道资源是否启用, 从低到高表示 1-64 通道: 0-禁用, 1-启用。

*struIPDevInfo*

IP 设备信息, 下标 0 对应设备 IP ID 为 1

*struStreamMode*

取流模式

*byRes2*

保留, 置为 0

## 7.49 NET\_DVR\_JPEGPARA:JPEG 图像信息

```
struct{
    WORD    wPicSize;
    WORD    wPicQuality;
}NET_DVR_JPEGPARA,*LPNET_DVR_JPEGPARA;
```

### Members

*wPicSize*

图片分辨率: 0-CIF(352\*288/352\*240), 1-QCIF(176\*144/176\*120), 2-4CIF(704\*576/704\*480)或 D1(720\*576/720\*486), 3-UXGA(1600\*1200), 4-SVGA(800\*600), 5-HD720P(1280\*720), 6-VGA(640\*480), 7-XVGA(1280\*960), 8-HD900P(1600\*900), 9-HD1080P(1920\*1080), 10-2560\*1920, 11-1600\*304, 12-2048\*1536, 13-2448\*2048, 14-2448\*1200, 15-2448\*800, 16-XGA(1024\*768), 17-SXGA(1280\*1024), 18-WD1(960\*576/960\*480), 19-1080I(1920\*1080), 20-576\*576, 21-1536\*1536, 22-1920\*1920, 0xff-Auto(使用当前码流分辨率)

*wPicQuality*

图片质量系数: 0-最好, 1-较好, 2-一般

### Remarks

图片分辨率需要设备支持, 建议设置为设备当前主码流编码分辨率。

## 7.50 NET\_DVR\_LAMP\_CTRL\_INFO:内外置灯参数配置

```
struct{
    DWORD    dwSize;
    BYTE    byLampCtrlMode;
    BYTE    byCtrlChannellIndex;
    BYTE    byRes[2];
}
```

```

NET\_DVR\_LAMP\_CTRL\_MODE\_UNION    uLampCtrlMode;
BYTE                                byRes2[32];
}NET_DVR_LAMP_CTRL_INFO, *LPNET_DVR_LAMP_CTRL_INFO;

```

**Members***dwSize*

结构体大小

*byLampCtrlMode*

控灯模式：1- 内置灯，2- 外接灯

*byCtrlChannellIndex*

交替控制通道号（IP 通道接入，取值范围：33~48）

*byRes*

保留，置为 0

*uLampCtrlMode*

灯控制模式联合体

*byRes2*

保留，置为 0

**7.51 NET\_DVR\_LAMP\_CTRL\_MODE\_UNION:灯控制模式联合体**

```

union{
    BYTE                                uLen[288];
    NET\_DVR\_INLAY\_LAMP\_CTRL\_MODE        struInlayLampCtrlMode;
    NET\_DVR\_EXTERNAL\_LAMP\_CTRL\_MODE     struExternalLampCtrlMode;
}NET_DVR_LAMP_CTRL_MODE_UNION, *LPNET_DVR_LAMP_CTRL_MODE_UNION;

```

**Members***uLen*

联合体大小，288 字节

*struInlayLampCtrlMode*

内置灯控制模式参数

*struExternalLampCtrlMode*

外接灯控制模式参数

**7.52 NET\_DVR\_LAMP\_EXTERNAL\_CFG:指示灯外控配置**

```

struct{
    DWORD                                dwSize;
    BYTE                                byEnable;
    BYTE                                byRes1[3];
    NET\_DVR\_LAMP\_STATE                    struLampStateCtrl;
    BYTE                                byRes2[32];
}NET_DVR_LAMP_EXTERNAL_CFG, *LPNET_DVR_LAMP_EXTERNAL_CFG;

```

**Members***dwSize*

结构体大小

*byEnable*

外控使能：0- 不启用，1- 启用

*byRes1*

保留，置为 0

*struLampStateCtrl*

指示灯外控状态信息

*byRes2*

保留，置为 0

## 7.53 NET\_DVR\_LAMP\_STATE: 指示灯外控状态信息

```
struct{
    BYTE    byFlicker;
    BYTE    byRes1[3];
    DWORD   dwIONo;
    BYTE    byRes2[8];
}NET_DVR_LAMP_STATE, *LPNET_DVR_LAMP_STATE;
```

### Members

*byFlicker*

0- 不闪烁，1- 闪烁

*byRes1*

保留，置为 0

*dwIONo*

按位表示，支持复选，bit1- IO1，bit2- IO2，bit4- IO3，取值：1- 启用，0- 不启用

*byRes*

保留，置为 0

## 7.54 NET\_DVR\_LANE\_PARAM\_V41: 车道参数

```
struct{
    BYTE    byRuleName[NAME_LEN];
    BYTE    byRuleID;
    BYTE    byLaneType;
    BYTE    byTrafficState;
    BYTE    byLaneNo;
    DWORD   dwVaryType;
    DWORD   dwTpsType;
    DWORD   dwLaneVolume;
    DWORD   dwLaneVelocity;
    DWORD   dwTimeHeadway ;
    DWORD   dwSpaceHeadway;
    float   fSpaceOccupyRation ;
}
```

```

float          fTimeOccupyRation;
DWORD         dwLightVehicle ;
DWORD         dwMidVehicle;
DWORD         dwHeavyVehicle ;
NET_DVR_LANE_QUEUE struLaneQueue;
NET_VCA_POINT struRuleLocation;
DWORD         dwOversizeVehicle ;
BYTE         byRes2[60];
}NET_DVR_LANE_PARAM_V41, *LPNET_DVR_LANE_PARAM_V41;

```

### Members

#### *byRuleName*

车道规则名称

#### *byRuleID*

规则序号，取值范围：1~8

#### *byLaneType*

车道上行或下行

#### *byTrafficState*

车道的交通状态，0-无效，1-畅通，2-拥挤，3-堵塞

#### *byLaneNo*

车道号

#### *dwVaryType*

车道交通参数变化类型参照，按位区分

```

enum _TRAFFIC_DATA_VARY_TYPE_EX_ENUM_{
    ENUM_TRAFFIC_VARY_NO           = 0x00,
    ENUM_TRAFFIC_VARY_VEHICLE_ENTER = 0x01,
    ENUM_TRAFFIC_VARY_VEHICLE_LEAVE = 0x02,
    ENUM_TRAFFIC_VARY_QUEUE       = 0x04,
    ENUM_TRAFFIC_VARY_STATISTIC    = 0x08,
}TRAFFIC_DATA_VARY_TYPE_EX_ENUM

```

#### *ENUM\_TRAFFIC\_VARY\_NO*

无变化

#### *ENUM\_TRAFFIC\_VARY\_VEHICLE\_ENTER*

车辆进入虚拟线圈

#### *ENUM\_TRAFFIC\_VARY\_VEHICLE\_LEAVE*

车辆离开虚拟线圈

#### *ENUM\_TRAFFIC\_VARY\_QUEUE*

队列变化

#### *ENUM\_TRAFFIC\_VARY\_STATISTIC*

统计数据变化（每分钟变化一次包括平均速度，车道空间/时间占有率，交通状态）

#### *dwTpsType*

数据变化类型标志，表示当前上传的统计参数中，哪些数据有效，按位区分

```

enum _ITS_TPS_TYPE_{
    LANE_VOLUME           = 0x1,
    LANE_VELOCITY        = 0x2,
}

```

```

TIME_HEADWAY          = 0x4,
SPACE_HEADWAY         = 0x8,
TIME_OCCUPANCY_RATIO = 0x10,
SPACE_OCCUPANCY_RATIO = 0x20,
QUEUE                 = 0x40,
VEHICLE_TYPE         = 0x80,
TRAFFIC_STATE        = 0x100
}ITS_TPS_TYPE

```

**LANE\_VOLUME**

车道流量

**LANE\_VELOCITY**

车道速度

**TIME\_HEADWAY**

车头时距

**SPACE\_HEADWAY**

车头间距

**TIME\_OCCUPANCY\_RATIO**

车道占有率(时间上)

**SPACE\_OCCUPANCY\_RATIO**

车道占有率, 百分比计算(空间上)

**QUEUE**

排队长度

**VEHICLE\_TYPE**

车辆类型

**TRAFFIC\_STATE**

交通状态

**dwLaneVolume**

车道流量, 统计有多少车子通过

**dwLaneVelocity**

车道内车辆的平均速度, 以千米/小时表示

**dwTimeHeadway**

车头时距, 以秒计算

**dwSpaceHeadway**

车头间距, 以米来计算

**fSpaceOccupyRation**

车道占有率, 百分比计算 (空间上, 车辆长度与监控路段总长度的比值)

**fTimeOccupyRation**

时间占有率, 百分比计算

**dwLightVehicle**

小型车数量

**dwMidVehicle**

中型车数量

**dwHeavyVehicle**

重型车数量

*struLaneQueue*

车道队列长度

*struRuleLocation*

规则位置虚拟线圈的中心

*dwOversizeVehicle*

大型车数量

*byRes2*

保留

## 7.55 NET\_DVR\_LANE\_QUEUE:车道队列

```
struct{
    NET_VCA_POINT    struHead;
    NET_VCA_POINT    struTail;
    DWORD            dwLength;
}NET_DVR_LANE_QUEUE,*LPNET_DVR_LANE_QUEUE;
```

### Members

*struHead*

队列头

*struTail*

队列尾

*dwLength*

实际队列长度，单位：米，范围[0-500]

## 7.56 NET\_DVR\_LOCAL\_ABILITY\_PARSE\_CFG:能力集解析库配置

```
struct{
    BYTE    byEnableAbilityParse;
    BYTE    byRes[127];
}NET_DVR_LOCAL_ABILITY_PARSE_CFG,*LPNET_DVR_LOCAL_ABILITY_PARSE_CFG;
```

### Members

*byEnableAbilityParse*

使用能力集解析库：0-不使用，1-使用，默认不使用

*byRes*

保留，置为 0

### Remarks

模拟能力集默认禁用，调用该接口可以启用模拟能力集，支持获取设备各种能力。如果需要获取能力集（NET\_DVR\_GetDeviceAbility），可以调用此接口来启用模拟能力集，并且需要加载 LocalXml.zip（要求和 SDK 库文件放在同一个目录下）。

## 7.57 NET\_DVR\_LOCAL\_BYTE\_ENCODE\_CONVERT: 字符编码转换参数

```
struct{
    CHAR_ENCODE_CONVERT    fnCharConvertCallBack;
    BYTE                    byRes[256];
}NET_DVR_LOCAL_BYTE_ENCODE_CONVERT,*LPNET_DVR_LOCAL_BYTE_ENCODE_CONVERT;
```

### Members

#### *fnCharConvertCallBack*

字符编码转换回调函数

#### *byRes*

保留，置为 0

### Callback Function

```
typedef int(CALLBACK *CHAR_ENCODE_CONVERT)(
    char        *pInput,
    DWORD       dwInputLen,
    DWORD       dwInEncodeType,
    char        *pOutput,
    DWORD       dwOutputLen,
    DWORD       dwOutEncodeType
);
```

### Callback Function Parameters

#### *pInput*

[in] 输入字符串，内存由 SDK 申请，字符串数据也由 SDK 提供

#### *dwInputLen*

[in] 输入字符串缓冲区大小

#### *dwInEncodeType*

[in] 输入的字符编码格式：0- 无字符编码信息(老设备)，1- GB2312(简体中文)，2- GBK，3- BIG5(繁体中文)，4- Shift\_JIS(日文)，5- EUC-KR(韩文)，6- UTF-8，7- ISO8859-1，8- ISO8859-2，9- ISO8859-3，...，依次类推，21- ISO8859-15(西欧)

#### *pOutput*

[out] 输出字符串，内存由 SDK 申请，存放使用用户字符编码接口转换之后的字符串

#### *dwOutputLen*

[out] 输出字符串缓冲区大小

#### *dwOutEncodeType*

[out] 输出字符编码格式：0- 无字符编码信息(老设备)，1- GB2312(简体中文)，2- GBK，3- BIG5(繁体中文)，4- Shift\_JIS(日文)，5- EUC-KR(韩文)，6- UTF-8，7- ISO8859-1，8- ISO8859-2，9- ISO8859-3，...，依次类推，21- ISO8859-15(西欧)

### Remarks

- 回调函数的返回值：-1 表示失败，0 表示成功（内存足够存放转换以后的字符串）。
- 设备的字符编码类型在登录接口返回，对应 [NET\\_DVR\\_DEVICEINFO\\_V40](#) 结构体中的参数 `byCharEncodeType`。SDK 内部需要字符编码转换时，SDK 默认使用 `libiconv` 库进行类型转换。如果不想使用 `libiconv` 编码库，可以调用 [NET\\_DVR\\_SetSDKLocalCfg](#) (类型: `NET_SDK_LOCAL_CFG_TYPE_BYTE_ENCODE`) 设置字符转码回调函数，告知 SDK 用户自己的字符编码接口，然后 SDK 将使用用户提供的字符编码接

口进行字符串处理。

## 7.58 NET\_DVR\_LOCAL\_CHECK\_DEV:设备在线巡检参数

```
struct{
    DWORD    dwCheckOnlineTimeout;
    DWORD    dwCheckOnlineNetFailMax;
    BYTE     byRes[256];
}NET_DVR_LOCAL_CHECK_DEV, *LPNET_DVR_LOCAL_CHECK_DEV;
```

### Members

#### *dwCheckOnlineTimeout*

巡检时间间隔，单位：ms，取值范围：30s~120s，0 表示用默认值(120s)，推荐设置 30s

#### *dwCheckOnlineNetFailMax*

由于网络原因失败的最大累加次数，达到该次数，SDK 才回调用户异常消息，0 表示使用默认值 1，推荐设置 3 次

#### *byRes*

保留，置为 0

### Remarks

- SDK 按照该结构体中的时间间隔对设备进行自动巡检，巡检过程中如果连失败或者重连成功在 [NET\\_DVR\\_SetExceptionCallBack\\_V30](#) 设置的异常消息回调函数中返回，对应异常消息类型为：EXCEPTION\_EXCHANGE、RESUME\_EXCHANGE。
- 推荐设置 30s 时间间隔、3 次，即心跳间隔为 1.5 分钟。

## 7.59 NET\_DVR\_LOCAL\_MEM\_POOL\_CFG:内存池本地配置

```
struct{
    DWORD    dwAlarmMaxBlockNum;
    DWORD    dwAlarmReleaseInterval;
    BYTE     byRes[60];
}NET_DVR_LOCAL_MEM_POOL_CFG, *LPNET_DVR_LOCAL_MEM_POOL_CFG;
```

### Members

#### *dwAlarmMaxBlockNum*

报警模块内存池最多向系统申请的内存块（block）个数，每个 block 为 64MB，超过这个上限则不向系统申请，0 表示无上限

#### *dwAlarmReleaseInterval*

报警模块空闲内存释放的间隔，单位：秒，为 0 表示不释放空闲的内存

#### *byRes*

保留，置为 0

## 7.60 NET\_DVR\_LOCAL\_MODULE\_RECV\_TIMEOUT\_CFG:按模块配置超时时间

```
struct{
    DWORD    dwPreviewTime;
    DWORD    dwAlarmTime;
    DWORD    dwVodTime;
    DWORD    dwElse;
    BYTE     byRes[512];
}NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG,*LPNET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG;
```

### Members

#### *dwPreviewTime*

预览模块超时时间，单位：毫秒，取值范围：0~3000,000

#### *dwAlarmTime*

报警模块超时时间，单位：毫秒，取值范围：0~3000,000

#### *dwVodTime*

回放模块超时时间，单位：毫秒，取值范围：0~3000,000

#### *dwElse*

其他模块超时时间，单位：毫秒，取值范围：0~3000,000

#### *byRes*

保留，置为 0

## 7.61 NET\_DVR\_LOCAL\_PROTECT\_KEY\_CFG:密钥配置

```
struct{
    BYTE     byProtectKey[128];
    BYTE     byRes[128];
}NET_DVR_LOCAL_PROTECT_KEY_CFG,*LPNET_DVR_LOCAL_PROTECT_KEY_CFG;
```

### Members

#### *byProtectKey*

密钥，默认设置为 0

#### *byRes*

保留，置为 0

## 7.62 NET\_DVR\_LOCAL\_TALK\_MODE\_CFG:对讲模式配置

```
struct{
    BYTE     byEnableAbilityParse;
    BYTE     byRes[127];
}NET_DVR_LOCAL_TALK_MODE_CFG,*LPNET_DVR_LOCAL_TALK_MODE_CFG;
```

### Members

**byTalkMode**

对讲模式：0- 使用对讲库（默认），1- 使用 windows api 模式

**byRes**

保留，置为 0

**Remarks**

V4.2.2.5 及以前版本 SDK 均采用 windows API 实现相关功能。之后版本默认使用语音对讲库的方式，语音对讲库模式下必须加载 AudioIntercom.dll 和 OpenAL32.dll。

**7.63 NET\_DVR\_LOCAL\_TCP\_PORT\_BIND\_CFG:本地 TCP 端口绑定配置**

```
struct{
    WORD    wLocalBindTcpMinPort;
    WORD    wLocalBindTcpMaxPort;
    BYTE    byRes[60];
}NET_DVR_LOCAL_TCP_PORT_BIND_CFG,*LPNET_DVR_LOCAL_TCP_PORT_BIND_CFG;
```

**Members****wLocalBindTcpMinPort**

本地绑定 TCP 最小端口

**wLocalBindTcpMaxPort**

本地绑定 TCP 最大端口

**byRes**

保留，置为 0

**Remarks**

端口绑定的策略是：给一个端口段，可以保证使用的端口都是在这个段里（多播除外），但不能保证每一个段内的端口都用到，因为是循环利用的；端口池中取出的端口会去尝试绑定，如果被占用了，将取下一个，如果段内每一个都绑定不了，则连接操作返回失败。建议最好不要设置系统预留的端口（1-1024），比如 80 等。

设置的最大端口应该大于等于最小端口，[0, 0]表示清除绑定，[0, 非 0]将设置失败，因为 0 不能进行绑定。

**7.64 NET\_DVR\_LOCAL\_UDP\_PORT\_BIND\_CFG:本地 UDP 端口绑定配置**

```
struct{
    WORD    wLocalBindUdpMinPort;
    WORD    wLocalBindUdpMaxPort;
    BYTE    byRes[60];
}NET_DVR_LOCAL_UDP_PORT_BIND_CFG,*LPNET_DVR_LOCAL_UDP_PORT_BIND_CFG;
```

**Members****wLocalBindUdpMinPort**

本地绑定 UDP 最小端口

**wLocalBindUdpMaxPort**

本地绑定 UDP 最大端口

**byRes**

保留，置为 0

### Remarks

端口绑定的策略是：给一个端口段，可以保证使用的端口都是在这个段里（多播除外），但不能保证每一个段内的端口都用到，因为是循环利用的；端口池中取出的端口会去尝试绑定，如果被占用了，将取下一个，如果段内每一个都绑定不了，则连接操作返回失败。建议最好不要设置系统预留的端口（1-1024），比如 80 等。

设置的最大端口应该大于等于最小端口，[0, 0]表示清除绑定，[0, 非 0]将设置失败，因为 0 不能进行绑定。

## 7.65 NET\_DVR\_LOG\_V30:日志信息

```
struct{
    NET\_DVR\_TIME      strLogTime;
    DWORD            dwMajorType;
    DWORD            dwMinorType;
    BYTE             sPanelUser[MAX_NAMELEN];
    BYTE             sNetUser[MAX_NAMELEN];
    NET\_DVR\_IPADDR    struRemoteHostAddr;
    DWORD            dwParaType;
    DWORD            dwChannel;
    DWORD            dwDiskNumber;
    DWORD            dwAlarmInPort;
    DWORD            dwAlarmOutPort;
    DWORD            dwInfoLen;
    char             sInfo[LOG_INFO_LEN];
}NET_DVR_LOG_V30,*LPNET_DVR_LOG_V30;
```

### Members

#### *strLogTime*

日志时间

#### *dwMajorType*

报警主类型，定义请参见[设备日志主类型](#)

#### *dwMinorType*

报警次类型，根据不同的主类型的次类型定义请参见[设备日志次类型](#)

#### *sPanelUser*

操作面板的用户名

#### *sNetUser*

网络操作的用户名

#### *struRemoteHostAddr*

远程主机地址

#### *dwParaType*

对于 DS-90xx 设备，当日志次类型为 MINOR\_START\_VT 或者 MINOR\_STOP\_VT 时，表示语音对讲端口号。

当日志的主类型为 MAJOR\_OPERATION=03（操作），且次类型为 MINOR\_LOCAL\_CFG\_PARM=0x52（本地配置参数）或 MINOR\_REMOTE\_GET\_PARM=0x76（远程获得参数）或

MINOR\_REMOTE\_CFG\_PARM=0x77（远程配置参数）时，该参数类型有效，其含义如下：

宏定义	宏定义值	含义
PARA_VIDEOOUT	0x1	视频输出结构配置
PARA_IMAGE	0x2	图像参数结构配置
PARA_ENCODE	0x4	压缩参数结构配置
PARA_NETWORK	0x8	网络参数结构配置
PARA_ALARM	0x10	报警参数结构配置
PARA_EXCEPTION	0x20	异常参数结构配置
PARA_DECODER	0x40	解码器参数结构配置
PARA_RS232	0x80	RS232 参数结构配置
PARA_PREVIEW	0x100	本地预览参数结构配置
PARA_SECURITY	0x200	用户权限参数结构配置
PARA_DATETIME	0x400	本地系统配置
PARA_FRAME TYPE	0x800	帧信息参数结构配置

*dwChannel*

通道号

*dwDiskNumber*

硬盘号

*dwAlarmInPort*

报警输入端口

*dwAlarmOutPort*

报警输出端口

*dwInfoLen*

日志附加信息长度

*sInfo*

日志附加信息

## 7.66 NET\_DVR\_NETCFG\_V30:网络配置

```

struct{
    DWORD                dwSize;
    NET\_DVR\_ETHERNET\_V30 struEtherNet[MAX_ETHERNET];
    NET\_DVR\_IPADDR       struRes1[2];
    NET\_DVR\_IPADDR       struAlarmHostIpAddr;
    WORD                 wRes2[2];
    WORD                 wAlarmHostIpPort;
    BYTE                 byUseDhcp;
    BYTE                 byRes3;
    NET\_DVR\_IPADDR       struDnsServer1IpAddr;
    NET\_DVR\_IPADDR       struDnsServer2IpAddr;
}

```

```

BYTE          byIpResolver[MAX_DOMAIN_NAME];
WORD          wIpResolverPort;
WORD          wHttpPortNo;
NET\_DVR\_IPADDR struMulticastIpAddr;
NET\_DVR\_IPADDR struGatewayIpAddr;
NET\_DVR\_PPPOECFG struPPPoE;
BYTE          byRes[64];
}NET_DVR_NETCFG_V30,*LPNET_DVR_NETCFG_V30;

```

## Members

### *dwSize*

结构体大小

### *struEtherNet*

以太网口

### *struRes1*

保留，置为 0

### *struAlarmHostIpAddr*

报警主机 IP 地址

### *wRes2*

保留，置为 0

### *wAlarmHostIpPort*

报警主机端口号

### *byUseDhcp*

是否启用 DHCP: 0xff-无效; 0-不启用; 1-启用

### *byRes3*

保留，置为 0

### *struDnsServer1IpAddr*

域名服务器 1 的 IP 地址

### *struDnsServer2IpAddr*

域名服务器 2 的 IP 地址

### *byIpResolver*

IP 解析服务器域名或 IP 地址 (8000 设备不支持域名)

### *wIpResolverPort*

IP 解析服务器端口号

### *wHttpPortNo*

HTTP 端口号

### *struMulticastIpAddr*

多播组地址

### *struGatewayIpAddr*

网关地址

### *struPPPoE*

PPPoE 参数

### *byRes*

保留，置为 0

## 7.67 NET\_DVR\_NOISEREMOVE:数字降噪功能参数

```
struct{
    BYTE        byDigitalNoiseRemoveEnable;
    BYTE        byDigitalNoiseRemoveLevel;
    BYTE        bySpectralLevel;
    BYTE        byTemporalLevel;
    BYTE        byRes[4];
}NET_DVR_NOISEREMOVE, *LPNET_DVR_NOISEREMOVE;
```

### Members

#### *byDigitalNoiseRemoveEnable*

数字去噪是否启用，0-不启用，1-普通模式数字降噪，2-专家模式数字降噪

#### *byDigitalNoiseRemoveLevel*

普通模式数字降噪级别：0x0-0xF

#### *bySpectralLevel*

专家模式下空域强度：0-100

#### *byTemporalLevel*

专家模式下时域强度：0-100

#### *byRes*

保留，置为 0

## 7.68 NET\_DVR\_NTTPARA:网络应用参数(NTP)

```
struct{
    BYTE        sNTPServer[64];
    WORD        wInterval;
    BYTE        byEnableNTP;
    signed char cTimeDifferenceH;
    signed char cTimeDifferenceM;
    BYTE        res1;
    WORD        wNtpPort;
    BYTE        res2[8];
}NET_DVR_NTTPARA, *LPNET_DVR_NTTPARA;
```

### Members

#### *sNTPServer*

NTP 服务器域名或者 IP 地址

#### *wInterval*

校时间隔时间（以小时为单位）

#### *byEnableNTP*

NTP 校时是否启用：0—否，1—是

#### *cTimeDifferenceH*

与国际标准时间的时差（小时），-12 ... +13

#### *cTimeDifferenceM*

与国际标准时间的时差（分钟），0, 30, 45

*res1*

保留，置为 0

*wNtpPort*

NTP 服务器端口，设备默认为 123

*res2*

保留，置为 0

## 7.69 NET\_DVR\_ONE\_AID\_RULE\_V41:单条交通事件规则

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNo;
    BYTE                byRes1[2];
    BYTE                byRuleName[NAME_LEN];
    DWORD              dwEventType;
    NET_VCA_SIZE_FILTER struSizeFilter;
    NET_VCA_POLYGON    struPolygon;
    NET_DVR_AID_PARAM  struAIDParam;
    NET_DVR_SCHEDTIME  struAlarmTime[MAX_DAYS][MAX_TIMESEGMENT_V30];
    NET_DVR_HANDLEEXCEPTION_V30 struHandleType;
    BYTE                byRelRecordChan[MAX_CHANNUM_V30];
    BYTE                byRes2[60];
}NET_DVR_ONE_AID_RULE_V41,*LPNET_DVR_ONE_AID_RULE_V41;
```

### Members

*byEnable*

是否启用事件规则：0-否；非 0-是

*byLaneNo*

关联车道号

*byRes1*

保留，置为 0

*byRuleName*

规则名称

*dwEventType*

交通事件类型，具体定义如下：

```
enum _TRAFFIC_AID_TYPE_{
    CONGESTION                = 0x1,
    PARKING                   = 0x2,
    INVERSE                   = 0x4,
    PEDESTRIAN               = 0x8,
    DEBRIS                   = 0x10,
    SMOG                     = 0x20,
    OVERLINE                 = 0x40,
    VEHICLE_CONTROL_LIST     = 0x80,
```

```

SPEED                = 0x100,
LANECHANGE           = 0x200,
TURNAROUND           = 0x400,
VEHICLE_OCCUPANCY_NONVEHICLE = 0x800,
GASSER               = 0x1000
}TRAFFIC_AID_TYPE

```

**CONGESTION**

拥堵

**PARKING**

停车

**INVERSE**

逆行

**PEDESTRIAN**

行人

**DEBRIS**

遗留物，抛洒物碎片

**SMOG**

烟雾

**OVERLINE**

压线

**VEHICLE\_CONTROL\_LIST**

名单数据

**SPEED**

超速

**LANECHANGE**

变道

**TURNAROUND**

掉头

**VEHICLE\_OCCUPANCY\_NONVEHICLE**

机动车占用非机动车位

**GASSER**

加塞

**struSizeFilter**

尺寸过滤器

**struPolygon**

规则区域

**struAIDParam**

交通事件参数

**struAlarmTime**

布防时间

**struHandleType**

处理方式

**byRelRecordChan**

报警触发的录象通道：1 表示触发该通道；0 表示不触发

*byRes2*

保留，置为 0

## 7.70 NET\_DVR\_ONE\_TPS\_RULE\_V41:单条交通事件规则

```
struct{
    BYTE                byEnable;
    BYTE                byLaneID;
    BYTE                byRes1[2];
    DWORD              dwCalcType;
    NET\_VCA\_SIZE\_FILTER   struSizeFilter;
    NET\_VCA\_POLYGON     struVirtualLoop;
    NET\_DVR\_SCHEDTIME   struAlarmTime[MAX_DAYS][MAX_TIMESEGMENT_V30];
    NET\_DVR\_HANDLEEXCEPTION\_V30 struHandleType;
    BYTE                byRes2[60];
}NET_DVR_ONE_TPS_RULE_V41,*LPNET_DVR_ONE_TPS_RULE_V41;
```

### Members

*byEnable*

是否启用车道交通规则

*byLaneID*

车道 ID

*byRes1*

保留，置为 0

*dwCalcType*

统计参数类型

```
enum_ITS_TPS_TYPE_{
    LANE_VOLUME           = 0x1,
    LANE_VELOCITY        = 0x2,
    TIME_HEADWAY         = 0x4,
    SPACE_HEADWAY        = 0x8,
    TIME_OCCUPANCY_RATIO = 0x10,
    SPACE_OCCUPANCY_RATIO = 0x20,
    QUEUE                = 0x40,
    VEHICLE_TYPE         = 0x80,
    TRAFFIC_STATE        = 0x100
}ITS_TPS_TYPE
```

*LANE\_VOLUME*

车道流量

*LANE\_VELOCITY*

车道速度

*TIME\_HEADWAY*

车头时距

*SPACE\_HEADWAY*

车头间距

*TIME\_OCCUPANCY\_RATIO*

车道占有率(时间上)

*SPACE\_OCCUPANCY\_RATIO*

车道占有率, 百分比计算(空间上)

*QUEUE*

排队长度

*VEHICLE\_TYPE*

车辆类型

*TRAFFIC\_STATE*

交通状态

*struSizeFilter*

尺寸过滤器

*struVirtualLoop*

虚拟线圈, 类似于地感线圈, 在实际场景中大于车子一半小于车子大小最适合

*struAlarmTime*

布防时间

*struHandleType*

处理方式。一般为处理是否上传中心, 其他功能不需要

*byRes2*

保留, 置为 0

## 7.71 NET\_DVR\_PARK\_EXTERNAL\_SUBINFO:外接灯模式车位信息子结构

```
struct{
    BYTE    byEnable;
    BYTE    byFlicker;
    BYTE    byIOState;
    BYTE    byRes[5];
}NET_DVR_PARK_EXTERNAL_SUBINFO, *LPNET_DVR_PARK_EXTERNAL_SUBINFO;
```

### Members

*byEnable*

使能: 0- 不开启, 1- 开启

*byFlicker*

0- 不闪烁, 1- 闪烁

*byIOState*

外置灯起效的 IO 状态: 0- 低电平, 1- 高电平

*byRes*

保留

## 7.72 NET\_DVR\_PARK\_INLAY\_SUBINFO:内置灯模式车位信息

```
struct{
    BYTE    byEnable;
```

```

BYTE    byFlicker;
BYTE    byLampColor;
BYTE    byRes[21];
}NET_DVR_PARK_INLAY_SUBINFO, *LPNET_DVR_PARK_INLAY_SUBINFO;

```

### Members

#### *byEnable*

使能：0- 不开启，1- 开启

#### *byFlicker*

0- 不闪烁，1- 闪烁

#### *byLampColor*

外置灯起效的颜色状态：0- 灭，1- 红，2- 绿，3- 黄，4- 蓝，5- 品红，6- 青，7- 白色

#### *byRes*

保留

## 7.73 NET\_DVR\_PARKINFO:外接灯模式车位信息

```

struct{
    NET_DVR_PARK_EXTERNAL_SUBINFO    struNormalParkIOState;
    NET_DVR_PARK_EXTERNAL_SUBINFO    struNormalNoParkIOState;
    NET_DVR_PARK_EXTERNAL_SUBINFO    struSpecialParkIOState;
    NET_DVR_PARK_EXTERNAL_SUBINFO    struSpecialNoParkIOState;
    BYTE                               byRes[32];
}NET_DVR_PARKINFO, *LPNET_DVR_PARKINFO;

```

### Members

#### *struNormalParkIOState*

有车 IO 电平状态

#### *struNormalNoParkIOState*

无车 IO 电平状态

#### *struSpecialParkIOState*

特殊车位有车 IO 电平状态

#### *struSpecialNoParkIOState*

特殊车位无车 IO 电平状态

#### *byRes*

保留

## 7.74 NET\_DVR\_PARKSPACE\_ATTRIBUTE:车位属性配置

```

struct{
    DWORD                dwSize;
    NET_DVR_PARKSPACE_INFO    struParkSpaceInfo[MAX_PARKING_NUM];
    BYTE                 byRes[64];
}NET_DVR_PARKSPACE_ATTRIBUTE, *LPNET_DVR_PARKSPACE_ATTRIBUTE;

```

### Members

*dwSize*

结构体大小

*struParkSpaceInfo*

车位属性，一个通道最大 4 个车位 (从左到右车位对应数组 0~3)

*byRes*

保留，置为 0

## 7.75 NET\_DVR\_PARKSPACE\_INFO:车位属性信息

```
struct{
    BYTE    byParkSpaceAttribute;
    BYTE    byRes[23];
}NET_DVR_PARKSPACE_INFO, *LPNET_DVR_PARKSPACE_INFO;
```

### Members

*byParkSpaceAttribute*

0- 普通车位(默认), 1- 特殊车位

*byRes*

保留，置为 0

## 7.76 NET\_DVR\_PICCFG\_V30:通道图像

```
struct{
    DWORD                dwSize;
    BYTE                 sChanName[NAME_LEN];
    DWORD                dwVideoFormat;
    char                 reservedData[64];
    DWORD                dwShowChanName;
    WORD                 wShowNameTopLeftX;
    WORD                 wShowNameTopLeftY;
    NET_DVR_VILOST_V30  struVILost;
    NET_DVR_VILOST_V30  struRes;
    NET_DVR_MOTION_V30  struMotion;
    NET_DVR_HIDEALARM_V30 struHideAlarm;
    DWORD                dwEnableHide;
    NET_DVR_SHELTER     struShelter[MAX_SHELTERNUM];
    DWORD                dwShowOsd;
    WORD                 wOSDTopLeftX;
    WORD                 wOSDTopLeftY;
    BYTE                 byOSDType;
    BYTE                 byDispWeek;
    BYTE                 byOSDAttrib;
    BYTE                 byHourOsdType;
    BYTE                 byFontSize;
```

```

    BYTE                byRes[63];
}NET_DVR_PICCFG_V30,*LPNET_DVR_PICCFG_V30;

```

### Members

#### *dwSize*

结构体大小

#### *sChanName*

通道名称

#### *dwVideoFormat*

视频制式：1-NTSC，2-PAL

#### *reservedData*

保留，置为 0

#### *dwShowChanName*

预览的图象上是否显示通道名称：0-不显示，1-显示（区域大小 704\*576）

#### *wShowNameTopLeftX*

通道名称显示位置的 x 坐标

#### *wShowNameTopLeftY*

通道名称显示位置的 y 坐标

#### *struVILost*

视频信号丢失报警参数，智能交通设备不支持，保留

#### *struRes*

保留，置为 0

#### *struMotion*

移动侦测报警参数，智能交通设备不支持，保留

#### *struHideAlarm*

遮挡报警参数，智能交通设备不支持，保留

#### *dwEnableHide*

是否启动遮挡：0-否，1-是，智能交通设备不支持，保留

#### *struShelter*

遮挡区域参数，智能交通设备不支持，保留

#### *dwShowOsd*

预览的图象上是否显示 OSD：0-不显示，1-显示（区域大小 704\*576）

#### *wOSDTopLeftX*

OSD 的 x 坐标

#### *wOSDTopLeftY*

OSD 的 y 坐标

#### *byOSDType*

OSD 类型(年月日格式):

0—XXXX-XX-XX 年月日

1—XX-XX-XXXX 月日年

2—XXXX 年 XX 月 XX 日

3—XX 月 XX 日 XXXX 年

4—XX-XX-XXXX 日月年

5—XX 日 XX 月 XXXX 年

#### *byDispWeek*

是否显示星期：0-不显示，1-显示

#### *byOSDAtrib*

OSD 属性（透明/闪烁）：

1—透明，闪烁

2—透明，不闪烁

3—闪烁，不透明

4—不透明，不闪烁

#### *byHourOsdType*

小时制：0 表示 24 小时制，1 表示 12 小时制或 am/pm

#### *byFontSize*

字体大小：0- 小，1- 中，2- 大，智能交通设备不支持，保留

#### *byRes*

保留，置为 0

## 7.77 NET\_DVR\_PLATE\_INFO:车牌识别结果

```
struct{
    BYTE                byPlateType;
    BYTE                byColor;
    BYTE                byBright;
    BYTE                byLicenseLen;
    BYTE                byEntireBelieve;
    BYTE                byRes[35];
    NET_VCA_RECT        struPlateRect;
    char                sLicense[MAX_LICENSE_LEN];
    BYTE                byBelieve[MAX_LICENSE_LEN];
}NET_DVR_PLATE_INFO, *LPNET_DVR_PLATE_INFO;
```

### Members

#### *byPlateType*

车牌类型，具体定义如下：

```
enum _VCA_PLATE_TYPE_{
    VCA_STANDARD92_PLATE = 0,
    VCA_STANDARD02_PLATE,
    VCA_WJPOLICE_PLATE,
    VCA_JINGCHE_PLATE,
    STANDARD92_BACK_PLATE,
    VCA_SHIGUAN_PLATE,
    VCA_NONGYONG_PLATE,
    VCA_MOTO_PLATE
}VCA_PLATE_TYPE
```

VCA\_STANDARD92\_PLATE

标准民用车与军车车牌

VCA\_STANDARD02\_PLATE

02 式民用车牌

*VCA\_WJPOLICE\_PLATE*

武警车车牌

*VCA\_JINGCHE\_PLATE*

警车车牌

*STANDARD92\_BACK\_PLATE*

民用车双行尾牌

*VCA\_SHIGUAN\_PLATE*

使馆车牌

*VCA\_NONGYONG\_PLATE*

农用车车牌

*VCA\_MOTO\_PLATE*

摩托车车牌

#### *byColor*

车牌颜色，具体定义如下：

```
enum _VCA_PLATE_COLOR_{
    VCA_BLUE_PLATE = 0,
    VCA_YELLOW_PLATE,
    VCA_WHITE_PLATE,
    VCA_BLACK_PLATE
}VCA_PLATE_COLOR
```

*VCA\_BLUE\_PLATE*

蓝色车牌

*VCA\_YELLOW\_PLATE*

黄色车牌

*VCA\_WHITE\_PLATE*

白色车牌

*VCA\_BLACK\_PLATE*

黑色车牌

#### *byBright*

车牌亮度

#### *byLicenseLen*

车牌字符个数

#### *byEntireBelieve*

整个车牌的置信度，0-100

#### *byRes*

保留

#### *struPlateRect*

车牌位置

#### *sLicense*

车牌号码

#### *byBelieve*

各个识别字符的置信度，如检测到车牌"浙 A12345"，置信度为 20,30,40,50,60,70，则表示"浙"字正确的可能性是 20%，"A"字的正确的可能性是 30%

## 7.78 NET\_DVR\_PLAYCOND:回放或者下载信息

```
struct{
    DWORD          dwChannel;
    NET\_DVR\_TIME   struStartTime;
    NET\_DVR\_TIME   struStopTime;
    BYTE           byDrawFrame;
    BYTE           byRes[63];
}NET_DVR_PLAYCOND, *LPNET_DVR_PLAYCOND;
```

### Members

*dwChannel*

通道号

*struStartTime*

开始时间

*struStopTime*

结束时间

*byDrawFrame*

是否抽帧：0- 不抽帧，1- 抽帧

*byRes*

保留

## 7.79 NET\_DVR\_POOLPARAM:云存储数据池参数

```
struct{
    DWORD   dwPoolID;
    BYTE    byRes[4];
}NET_DVR_POOLPARAM,*LPNET_DVR_POOLPARAM;
```

### Members

*dwPoolID*

云存储分配节点号

*byRes*

保留，置为 0

## 7.80 NET\_DVR\_PPPOECFG:PPPoE 配置

```
struct{
    DWORD          dwPPPOE;
    BYTE           sPPPoEUser[NAME_LEN];
    char           sPPPoEPassword[PASSWD_LEN];
    NET\_DVR\_IPADDR struPPPoEIP;
}NET_DVR_PPPOECFG, *LPNET_DVR_PPPOECFG;
```

### Members

**dwPPPOE**

是否启用 PPPoE: 0-不启用, 1-启用

**sPPPoEUser**

PPPoE 用户名

**sPPPoEPassword**

PPPoE 密码

**struPPPoEIP**

PPPoE IP 地址

## 7.81 NET\_DVR\_PREVIEWINFO:预览参数

```
struct{
    LONG      IChannel;
    DWORD     dwStreamType;
    DWORD     dwLinkMode;
    HWND      hPlayWnd;
    BOOL      bBlocked;
    BOOL      bPassbackRecord;
    BYTE      byPreviewMode;
    BYTE      byStreamID[STREAM_ID_LEN];
    BYTE      byProtoType;
    BYTE      byRes[222];
}NET_DVR_PREVIEWINFO, *LPNET_DVR_PREVIEWINFO;
```

### Members

**IChannel**

通道号, 数字通道号一般从 33 开始, 实际取值可通过 [NET\\_DVR\\_GetDVRConfig](#) (配置命令 NET\_DVR\_GET\_IPPARACFG\_V40) 获取 (dwStartDChan)。

**dwStreamType**

码流类型: 0-主码流, 1-子码流, 2-码流 3, 3-码流 4, 以此类推

**dwLinkMode**

连接方式: 0- TCP 方式, 1- UDP 方式, 2- 多播方式, 3- RTP 方式, 4-RTP/RTSP, 5-RSTP/HTTP

**hPlayWnd**

播放窗口的句柄, 为 NULL 表示不解码显示

**bBlocked**

0- 非阻塞取流, 1- 阻塞取流。如果阻塞取流, SDK 内部 connect 失败将会有 5s 的超时才能够返回, 不适合于轮询取流操作。

**bPassbackRecord**

0-不启用录像回传, 1-启用录像回传。录像回传即断网录像功能, 暂不支持, 置为 0

**byPreviewMode**

预览模式: 0- 正常预览, 1- 延迟预览

**byStreamID**

流 ID, IChannel 为 0xffffffff 时启用此参数

**byProtoType**

应用层取流协议: 0- 私有协议, 1- RTSP 协议

*byRes*

保留

### Remarks

- 该结构体中可以设置当前预览操作是否阻塞（通过 `bBlocked` 参数设置）。若设为不阻塞，表示发起与设备的连接就认为连接成功，如果发生码流接收失败、播放失败等情况以预览异常的方式通知上层。在循环播放的时候可以减短停顿的时间，与 `NET_DVR_RealPlay` 处理一致。若设为阻塞，表示直到播放操作完成才返回成功与否。
- `dwStreamType`、`bPassbackRecord`、`byPreviewMode`、`byStreamID` 这些参数的取值需要设备支持。
- `NET_DVR_RealPlay_V40` 支持多播方式预览（`dwLinkMode` 设为 2），不需要传多播组地址，底层自动从设备获取已配置的多播组地址（`NET_DVR_NETCFG_V30->struMulticastIpAddr`）并以该多播组地址实现多播。

## 7.82 NET\_DVR\_PROTO\_TYPE:协议参数

```
struct{
    DWORD   dwType;
    BYTE    byDescribe[DESC_LEN];
}NET_DVR_PROTO_TYPE,*LPNET_DVR_PROTO_TYPE;
```

### Members

*dwType*

协议值

*byDescribe*

协议描述

### Remarks

`dwType` 取值对应的协议类型可以见 `byDescribe` 协议描述，也可以参考下列枚举类型：

```
enum _NET_DVR_IPC_ENUM_{
    ENUM_BUSINESS_INVALID = -1,
    ENUM_BUSINESS_PRIVATE = 0,
    ENUM_BUSINESS_PANASONIC,
    ENUM_BUSINESS_SONY,
    ENUM_BUSINESS_AXIS,
    ENUM_BUSINESS_SANYO,
    ENUM_BUSINESS_BOSCH,
    ENUM_BUSINESS_ZAVIO,
    ENUM_BUSINESS_GRANDEYE,
    ENUM_BUSINESS_PROVIDEO,
    ENUM_BUSINESS_ARECONT,          //9
    ENUM_BUSINESS_ACTI,
    ENUM_BUSINESS_PELCO,
    ENUM_BUSINESS_VIVOTEK,
    ENUM_BUSINESS_INFINOVA,
    ENUM_BUSINESS_DAHUA,          //14
    ENUM_BUSINESS_PRIVT_STD_H264 = 0x20,
    ENUM_BUSINESS_PRIVT_STD_MPEG4,
    ENUM_BUSINESS_SUNELL,        //景阳
```

```

ENUM_BUSINESS_ATEME,
ENUM_BUSINESS_LAUNCH,      //朗驰
ENUM_BUSINESS_YAAN,        //雅安
ENUM_BUSINESS_BLUESKY,     //蓝色星际
ENUM_BUSINESS_BLUESKYLIMIT, //蓝色星际
ENUM_BUSINESS_TDWY,        //天地伟业
ENUM_BUSINESS_HBGK,        //汉邦高科
ENUM_BUSINESS_SANTACHI,    //金三立
ENUM_BUSINESS_HIGHEASY,    //恒忆
ENUM_BUSINESS_SAMSUNG,
ENUM_BUSINESS_URL_RTSP = 0x40, //url 类型取流
ENUM_BUSINESS_ONVIF,
ENUM_MAX_BUSINESS_TYPE,    //最大厂商类型
}NET_DVR_IPC_ENUM

```

## 7.83 NET\_DVR\_PTZCFG:云台协议

```

struct{
    DWORD          dwSize;
    NET\_DVR\_PTZ\_PROTOCOL struPtz[PTZ_PROTOCOL_NUM];
    DWORD          dwPtzNum;
    BYTE           byRes[8];
}NET_DVR_PTZCFG, *LPNET_DVR_PTZCFG;

```

### Members

#### *dwSize*

结构体大小

#### *struPtz*

协议信息，最多 200 种

#### *dwPtzNum*

有效的 PTZ 协议数目，从 0 开始（即总数为该值加 1）

#### *byRes*

保留，置为 0

## 7.84 NET\_DVR\_PTZPOS:球机位置信息

```

struct{
    WORD          wAction;
    WORD          wPanPos;
    WORD          wTiltPos;
    WORD          wZoomPos;
}NET_DVR_PTZPOS, *LPNET_DVR_PTZPOS;

```

### Members

**wAction**

操作类型，仅在设置时有效。1-定位 PTZ 参数，2-定位 P 参数，3-定位 T 参数，4-定位 Z 参数，5-定位 PT 参数

**wPanPos**

P 参数（水平参数）

**wTiltPos**

T 参数（垂直参数）

**wZoomPos**

Z 参数（变倍参数）

**Remarks**

本结构体中的 **wAction** 参数是设置时的操作类型，因此获取时该参数无效。实际显示的 PTZ 值是获取到的十六进制值的十分之一，如获取的水平参数 P 的值是 0x1750，实际显示的 P 值为 175 度；获取到的垂直参数 T 的值是 0x0789，实际显示的 T 值为 78.9 度；获取到的变倍参数 Z 的值是 0x1100，实际显示的 Z 值为 110 度。

**7.85 NET\_DVR\_PTZ\_MANUALTRACE:手动定位参数**

```
struct{
    DWORD          dwSize;
    DWORD          dwChannel;
    NET_VCA_POINT  struPoint;
    BYTE           byTrackType;
    BYTE           byRes[63];
}NET_DVR_PTZ_MANUALTRACE,*LPNET_DVR_PTZ_MANUALTRACE;
```

**Members****dwSize**

结构体大小

**dwChannel**

通道号

**struPoint**

定位坐标

**byTrackType**

类型：0- 普通，1- 高速公路，2- 城市道路

**byRes**

保留

**7.86 NET\_DVR\_PTZ\_PROTOCOL:云台协议信息**

```
struct{
    DWORD  dwType;
    BYTE   byDescribe[DESC_LEN];
}NET_DVR_PTZ_PROTOCOL,*LPNET_DVR_PTZ_PROTOCOL;
```

**Members**

*dwType*

协议类型值

*byDescribe*

协议描述符

## 7.87 NET\_DVR\_RECORDDAY:全天录像参数

```
struct{
    WORD    wAllDayRecord;
    BYTE    byRecordType;
    char    reservedData;
}NET_DVR_RECORDDAY, *LPNET_DVR_RECORDDAY;
```

### Members

*wAllDayRecord*

是否全天录像：0-否，1-是

*byRecordType*

录像类型：0-定时录像

*reservedData*

保留，置为 0

## 7.88 NET\_DVR\_RECORDSCHED:时间段录像参数

```
struct{
    NET\_DVR\_SCHEDTIME    struRecordTime;
    BYTE                  byRecordType;
    char                  reservedData[3];
}NET_DVR_RECORDSCHED, *LPNET_DVR_RECORDSCHED;
```

### Members

*struRecordTime*

录像时间

*byRecordType*

录像类型：0-定时录像

*reservedData*

保留，置为 0

## 7.89 NET\_DVR\_RECORD\_V30:通道录像参数

```
struct{
    DWORD                dwSize;
    DWORD                dwRecord;
    NET\_DVR\_RECORDDAY    struRecAllDay[MAX_DAYS];
}
```

```

NET_DVR_RECORDSCHED   struRecordSched[MAX_DAYS][MAX_TIMESEGMENT_V30];
DWORD                   dwRecordTime;
DWORD                   dwPreRecordTime;
DWORD                   dwRecorderDuration;
BYTE                    byRedundancyRec;
BYTE                    byAudioRec;
BYTE                    byReserve[10];
}NET_DVR_RECORD_V30, *LPNET_DVR_RECORD_V30;

```

### Members

*dwSize*

结构体大小

*dwRecord*

是否录像：0-否，1-是

*struRecAllDay*

全天录像布防参数

*struRecordSched*

时间段录像布防参数

*dwRecordTime*

录像延时时间，0-5 秒， 1-10 秒， 2-30 秒， 3-1 分钟， 4-2 分钟， 5-5 分钟， 6-10 分钟

*dwPreRecordTime*

预录时间：0-不预录， 1-5 秒， 2-10 秒， 3-15 秒， 4-20 秒， 5-25 秒， 6-30 秒， 7-0xffffffff(尽可能预录)

*dwRecorderDuration*

录像保存的最长时间，单位：天，0xffffffff 表示该值无效

*byRedundancyRec*

是否冗余录像（重要数据双备份）：0-不录像，1-录像

*byAudioRec*

录像时复合流编码时是否记录音频数据：0-不记录，1-记录

*byReserve*

保留，置为 0

## 7.90 NET\_DVR\_SCENE\_INFO:场景信息

```

struct{
    DWORD           dwSceneID;
    BYTE            bySceneName[NAME_LEN];
    BYTE            byDirection;
    BYTE            byRes1[3];
    NET_DVR_PTZPOS struPtzPos;
    BYTE            byRes2[64];
}NET_DVR_SCENE_INFO, *LPNET_DVR_SCENE_INFO;

```

### Members

*dwSceneID*

场景 ID，0- 表示该场景无效

*bySceneName*

场景名称

*byDirection*

监测方向：1-上行，2-下行，3-双向，4-由东向西，5-由南向北，6-由西向东，7-由北向南，8-其它

*byRes1*

保留

*struPtzPos*

PTZ 坐标

*byRes2*

保留

## 7.91 NET\_DVR\_SCHEDULETIME:起止时间段参数

```
struct{
    BYTE    byStartHour;
    BYTE    byStartMin;
    BYTE    byStopHour;
    BYTE    byStopMin;
}NET_DVR_SCHEDULETIME, *LPNET_DVR_SCHEDULETIME;
```

### Members

*byStartHour*

开始时间：时

*byStartMin*

开始时间：分

*byStopHour*

结束时间：时

*byStopMin*

结束时间：分

## 7.92 NET\_DVR\_SDKSTATE:SDK 状态信息

```
struct{
    DWORD    dwTotalLoginNum;
    DWORD    dwTotalRealPlayNum;
    DWORD    dwTotalPlayBackNum;
    DWORD    dwTotalAlarmChanNum;
    DWORD    dwTotalFormatNum;
    DWORD    dwTotalFileSearchNum;
    DWORD    dwTotalLogSearchNum;
    DWORD    dwTotalSerialNum;
    DWORD    dwTotalUpgradeNum;
    DWORD    dwTotalVoiceComNum;
    DWORD    dwTotalBroadCastNum;
    DWORD    dwRes[10];
}
```

```
}NET_DVR_SDKSTATE,*LPNET_DVR_SDKSTATE;
```

### Members

*dwTotalLoginNum*

当前注册的用户数

*dwTotalRealPlayNum*

当前实时预览的路数

*dwTotalPlayBackNum*

当前回放或下载的路数

*dwTotalAlarmChanNum*

当前建立报警通道的路数

*dwTotalFormatNum*

当前硬盘格式化的路数

*dwTotalFileSearchNum*

当前文件搜索的路数

*dwTotalLogSearchNum*

当前日志搜索的路数

*dwTotalSerialNum*

当前建立透明通道的路数

*dwTotalUpgradeNum*

当前升级的路数

*dwTotalVoiceComNum*

当前语音转发的路数

*dwTotalBroadCastNum*

当前语音广播的路数

*dwRes*

保留，置为 0

## 7.93 NET\_DVR\_SDKABL:SDK 功能信息

```
struct{
    DWORD    dwMaxLoginNum;
    DWORD    dwMaxRealPlayNum;
    DWORD    dwMaxPlayBackNum;
    DWORD    dwMaxAlarmChanNum;
    DWORD    dwMaxFormatNum;
    DWORD    dwMaxFileSearchNum;
    DWORD    dwMaxLogSearchNum;
    DWORD    dwMaxSerialNum;
    DWORD    dwMaxUpgradeNum;
    DWORD    dwMaxVoiceComNum;
    DWORD    dwMaxBroadCastNum;
    DWORD    dwRes[10];
}NET_DVR_SDKABL,*LPNET_DVR_SDKABL;
```

### Members

<i>dwMaxLoginNum</i>	最大注册用户数
<i>dwMaxRealPlayNum</i>	最大实时预览的路数
<i>dwMaxPlayBackNum</i>	最大回放或下载的路数
<i>dwMaxAlarmChanNum</i>	最大建立报警通道的路数
<i>dwMaxFormatNum</i>	最大硬盘格式化的路数
<i>dwMaxFileSearchNum</i>	最大文件搜索的路数
<i>dwMaxLogSearchNum</i>	最大日志搜索的路数
<i>dwMaxSerialNum</i>	最大建立透明通道的路数
<i>dwMaxUpgradeNum</i>	最大升级的路数
<i>dwMaxVoiceComNum</i>	最大语音转发的路数
<i>dwMaxBroadCastNum</i>	最大语音广播的路数
<i>dwRes</i>	保留，置为 0

## 7.94 NET\_DVR\_SETUPALARM\_PARAM:报警布防参数

```

struct{
    DWORD    dwSize;
    BYTE     byLevel;
    BYTE     byAlarmInfoType;
    BYTE     byRetAlarmTypeV40;
    BYTE     byRetDevInfoVersion;
    BYTE     byRetVQDAAlarmType;
    BYTE     byFaceAlarmDetection;
    BYTE     bySupport;
    BYTE     byRes;
    WORD     wTaskNo;
    BYTE     byRes1[5];
    BYTE     byCustomCtrl;
}NET_DVR_SETUPALARM_PARAM, *LPNET_DVR_SETUPALARM_PARAM;

```

### Members

*dwSize*  
结构体大小

**byLevel**

布防优先级：0- 一等级（高），1- 二等级（中），2- 三等级（低，保留）

**byAlarmInfoType**

智能交通报警信息上传类型：0- 老报警信息（NET\_DVR\_PLATE\_RESULT），1- 新报警信息（NET\_ITS\_PLATE\_RESULT）

**byRetAlarmTypeV40**

ITS 终端服务器不支持，保留，置为 0

**byRetDevInfoVersion**

ITS 终端服务器不支持，保留，置为 0

**byRetVQDAAlarmType**

ITS 终端服务器不支持，保留，置为 0

**byFaceAlarmDetection**

ITS 终端服务器不支持，保留，置为 0

**bySupport**

ITS 终端服务器不支持，保留，置为 0

**byRes**

保留，置为 0

**wTaskNo**

任务处理号

**byRes1**

保留，置为 0

**byCustomCtrl**

ITS 终端服务器不支持，保留，置为 0

**Remarks**

- **byLevel** 和 **byAlarmInfoType** 针对智能交通设备（抓拍机）：一级布防最大连接数为 1 个，二级最大连接数为 3 个，设备支持一级和二级布防同时进行，一级布防优先上传信息；**byAlarmInfoType** 是否支持新报警信息可从注册返回的能力获知，详见 [NET\\_DVR\\_DEVICEINFO\\_V30](#) 结构中 **bySupport1**（表示是否支持车牌新报警信息），如果注册返回能力不支持，设备仅支持老报警信息上传。
- **wTaskNo** 针对车辆二次检测设备，用于区分不同布防链接，现在上传机制为：
  - 1) 如果 **wTaskNo** 为 0，所有的处理结果都需要从这个链接上传。
  - 2) 如果两个布防链接中 **wTaskNo** 的值相同（0 除外），返回布防链接错误。
  - 3) 布防链接后，Client 端下发任务单号 **wTaskNo**，和车辆二次识别任务和上传的结果中的 **wTaskNo** 都对应的。例如：布防链接中 **wTaskNo**==1，任务 A 中 **wTaskNo**==1，结果信息回调 **wTaskNo**==1（该信息回调只在布防中 **wTaskNo** == 1 的链接中回调）。

**7.95 NET\_DVR\_SHOWSTRINGINFO: 字符叠加参数子结构**

```
struct{
    WORD        wShowString;
    WORD        wStringSize;
    WORD        wShowStringTopLeftX;
    WORD        wShowStringTopLeftY;
    char        sString[44];
}
```

```
}NET_DVR_SHOWSTRINGINFO,*LPNET_DVR_SHOWSTRINGINFO;
```

### Members

#### *wShowString*

预览的图象上是否显示字符：0—不显示，1—显示（显示区域范围为 704\*576，单个字符的大小为 32\*32）

#### *wStringSize*

该行字符的长度，不能大于 44 个字符

#### *wShowStringTopLeftX*

字符显示位置的 x 坐标

#### *wShowStringTopLeftY*

字符显示位置的 y 坐标

#### *sString*

要显示的字符内容

## 7.96 NET\_DVR\_SHOWSTRING\_V30:字符叠加参数结构体

```
struct{
    DWORD                dwSize;
    NET_DVR_SHOWSTRINGINFO struStringInfo[MAX_STRINGNUM_V30];
}NET_DVR_SHOWSTRING_V30,*LPNET_DVR_SHOWSTRING_V30;
```

### Members

#### *dwSize*

结构体大小

#### *struStringInfo*

要显示的字符内容，支持 4 行，取数组 struStringInfo [0]~ struStringInfo [3]

## 7.97 NET\_DVR\_SINGLE\_HD:设备硬盘信息配置结构体

```
struct{
    DWORD    dwHDNo;
    DWORD    dwCapacity;
    DWORD    dwFreeSpace;
    DWORD    dwHdStatus;
    BYTE     byHDAttr;
    BYTE     byHDType;
    BYTE     byDiskDriver;
    BYTE     byRes1[1];
    DWORD    dwHdGroup;
    BYTE     byRecycling;
    BYTE     byRes2[3];
    DWORD    dwStorageType;
    DWORD    dwPictureCapacity;
    DWORD    dwFreePictureSpace;
```

```

    BYTE        byRes3[104];
}NET_DVR_SINGLE_HD, *LPNET_DVR_SINGLE_HD;

```

### Members

#### *dwHDNo*

硬盘号，取值范围[0,MAX\_DISKNUM\_V30-1]，其中#define MAX\_DISKNUM\_V30 33

#### *dwCapacity*

硬盘容量，该参数只能获取，不支持设置

#### *dwFreeSpace*

硬盘剩余空间，该参数只能获取，不支持设置

#### *dwHdStatus*

硬盘状态，该参数只能获取，不支持设置：0—正常，1—未格式化，2—错误，3—S.M.A.R.T 状态，4—不匹配，5—休眠，6-未连接状态（网络硬盘），7-虚拟磁盘正常且支持扩容，8-虚拟磁盘次正常(未使用)，9-虚拟磁盘降级(未使用)，10-硬盘正在修复(9000v2.0)，11-硬盘正在格式化(9000v2.0)

#### *byHDAttr*

硬盘属性：0—默认，1—冗余（备份重要数据），2—只读

#### *byHDType*

硬盘类型：0-本地硬盘，1-ESATA 硬盘，2-NFS 硬盘，3-iSCSI 硬盘，4-RAID 虚拟磁盘

#### *byDiskDriver*

硬盘盘值，代表其 ASCII 字符

#### *byRes1*

保留，置为 0

#### *dwHdGroup*

该硬盘属于哪个盘组,取值范围[1,MAX\_HD\_GROUP],其中#define MAX\_HD\_GROUP 16

#### *byRecycling*

是否循环利用：0- 不循环利用，1- 循环利用

#### *byRes2*

保留，置为 0

#### *dwStorageType*

按位表示：0-不支持，非 0-支持，

*dwStorageType* & 0x1 表示是否是普通录像专用存储盘

*dwStorageType* & 0x2 表示是否是抽帧录像专用存储盘

*dwStorageType* & 0x4 表示是否是图片录像专用存储盘

#### *dwPictureCapacity*

硬盘图片容量(不可设置)，单位：MB

#### *dwFreePictureSpace*

剩余图片空间(不可设置)，单位：MB

#### *byRes3*

保留，置为 0

### Remarks

- 本结构体中的 *dwCapacity*、*dwFreeSpace* 和 *dwHdStatus* 参数是关于设备本地硬盘的属性，因此只能获取该信息，不能设置。
- *dwCapacity* 和 *dwFreeSpace* 表示录像和图片的总容量；*dwPictureCapacity* 是指预留给图片的总容量，其值非 0 有效，为 0 表示无效。IPC 等设备该值有效，DVR、NVR 等设备不区分录像和图片容量，统一分配。

## 7.98 NET\_DVR\_STREAM\_INFO:流 ID 信息

```
struct{
    DWORD    dwSize;
    BYTE     byID[STREAM_ID_LEN];
    DWORD    dwChannel;
    BYTE     byRes[32];
}NET_DVR_STREAM_INFO,*LPNET_DVR_STREAM_INFO;
```

### Members

#### *dwSize*

结构体大小

#### *byID*

流 ID，全部为 0 时，无效

#### *dwChannel*

关联的设备通道。等于 0xffffffff 时，如果是设置流的来源信息(NET\_DVR\_SET\_STREAM\_SRC\_INFO)，表示不关联；如果是作为其他如 NET\_DVR\_SET\_STREAM\_RECORD\_INFO、NET\_DVR\_SET\_STREAM\_RECORD\_STATUS、NET\_DVR\_SET\_MONITOR\_VQDCFG 等配置时的输入条件参数时，表示无效。

#### *byRes*

保留，置为 0

### Remarks

如果设备不支持流 ID 标识功能，例如 DVR 设备，byID 值设为 0。

当作为 SDK 接口的输入条件参数时，如果 byID 和 dwChannel 都无效时，网络 SDK 将返回参数错误（错误代码：17）；如果两个都有效，而输入的 byID 和 dwChannel 不匹配，则设备可能会返回失败，因此，建议只输入 byID（dwChannel=0xffffffff）或者 dwChannel（byID=0）。

## 7.99 NET\_DVR\_STREAM\_MODE:取流方式配置参数

```
struct{
    BYTE     byGetStreamType;
    BYTE     byRes[3];
    NET\_DVR\_GET\_STREAM\_UNION uGetStream;
}NET_DVR_STREAM_MODE,*LPNET_DVR_STREAM_MODE;
```

### Members

#### *byGetStreamType*

取流方式：0- 直接从设备取流；1- 从流媒体取流；2- 通过 IP Server 获得 IP 地址后取流；3- 通过 IP Server 找到设备，再通过流媒体取设备的流；4- 通过流媒体由 URL 去取流；5- 通过 hiDDNS 域名连接设备然后从设备取流

#### *byRes*

保留，置为 0

#### *uGetStream*

不同取流方式联合体

## 7.100 NET\_DVR\_SYSTEM\_TIME:时间信息

```

struct{
    WORD    wYear;
    WORD    wMonth;
    WORD    wDay;
    WORD    wHour;
    WORD    wMinute;
    WORD    wSecond;
    WORD    wMilliSec;
    BYTE    byRes[2];
}NET_DVR_SYSTEM_TIME,*LPNET_DVR_SYSTEM_TIME;

```

### Members

*wYear*  
年

*wMonth*  
月

*wDay*  
日

*wHour*  
时

*wMinute*  
分

*wSecond*  
秒

*wMilliSec*  
毫秒

*byRes*  
保留

## 7.101 NET\_DVR\_TFS\_ALARM:违章取证报警信息

```

struct{
    DWORD    dwSize;
    DWORD    dwRelativeTime;
    DWORD    dwAbsTime;
    DWORD    dwIllegalType;
    DWORD    dwIllegalDuration;
    BYTE    byMonitoringSiteID[MONITORSITE_ID_LEN];
    BYTE    byDeviceID[DEVICE_ID_LEN];
    NET\_VCA\_DEV\_INFO    struDevInfo;
    NET\_DVR\_SCENE\_INFO    struSceneInfo;
    NET\_DVR\_TIME\_EX    struBeginRecTime;

```

```

NET_DVR_TIME_EX          struEndRecTime;
NET_DVR_AID_INFO         struAIDInfo;
NET_DVR_PLATE_INFO       struPlateInfo;
NET_DVR_VEHICLE_INFO     struVehicleInfo;
DWORD                    dwPicNum;
NET_ITS_PICTURE_INFO     struPicInfo[8];
BYTE                     bySpecificVehicleType;
BYTE                     byLaneNo;
BYTE                     byRes1[2];
NET_DVR_TIME_V30        struTime;
DWORD                    dwSerialNo;
BYTE                     byVehicleAttribute;
BYTE                     byPilotSafebelt;
BYTE                     byCopilotSafebelt;
BYTE                     byPilotSunVisor;
BYTE                     byCopilotSunVisor;
BYTE                     byPilotCall;
BYTE                     byRes[102];
}NET_DVR_TFS_ALARM, *LPNET_DVR_TFS_ALARM;

```

### Members

*dwSize*

结构体大小

*dwRelativeTime*

相对时标

*dwAbsTime*

绝对时标

*dwIllegalType*

违章类型，采用国标定义，详细定义说明见 [7.161](#) 违章类型国标定义

*dwIllegalDuration*

违法持续时间（单位：秒） = 抓拍最后一张图片的时间 - 抓拍第一张图片的时间

*byMonitoringSiteID*

监测点编号（路口编号、内部编号）

*byDeviceID*

设备编号

*struDevInfo*

前端设备信息

*struSceneInfo*

场景信息

*struBeginRecTime*

录像开始时间

*struEndRecTime*

录像结束时间

*struAIDInfo*

交通事件信息

**struPlateInfo**

车牌信息

**struVehicleInfo**

车辆信息

**dwPicNum**

图片数量

**struPicInfo**

图片信息，最多 8 张

**bySpecificVehicleType**

具体车辆类型：0- 未知，1- 客车，2- 货车，3- 轿车，4- 面包车，5- 小货车，6- 行人，7- 二轮车，8- 三轮车

**byLaneNo**

关联车道号

**byRes1**

保留

**struTime**

手动定位的当前时间，目前仅取证球机支持，和 dwSerialNo 一起用于平台录像和手动取证图片匹配使用

**dwSerialNo**

抓拍序号，目前仅取证球机支持，和 struTime 一起用于平台录像和手动取证图片匹配使用

**byVehicleAttribute**

车辆属性，按位表示，0- 无附加属性(普通车辆)，bit1- 黄标车(类似年检的标志)，bit2- 危险品车辆，值：0- 否，1- 是

**byPilotSafebelt**

主驾驶员是否系安全带：0- 未知，1- 系安全带，2- 未系安全带

**byCopilotSafebelt**

副驾驶员是否系安全带：0- 未知，1- 系安全带，2- 未系安全带

**byPilotSunVisor**

主驾驶是否打开遮阳板：0- 未知，1- 未打开遮阳板，2- 打开遮阳板

**byCopilotSunVisor**

副驾驶是否打开遮阳板：0- 未知，1- 未打开遮阳板，2- 打开遮阳板

**byPilotCall**

主驾驶员是否在打电话：0- 未知，1- 未打电话，2- 打电话

**byRes**

保留

**Remarks**

从 DWORD 时间值得到年月日时分秒的算法如下所示：

```
#define GET_YEAR(_time_)    (((_time_)>>26) + 2000)
#define GET_MONTH(_time_)  (((_time_)>>22) & 15)
#define GET_DAY(_time_)    (((_time_)>>17) & 31)
#define GET_HOUR(_time_)   (((_time_)>>12) & 31)
#define GET_MINUTE(_time_) (((_time_)>>6) & 63)
#define GET_SECOND(_time_) (((_time_)>>0) & 63)
```

## 7.102 NET\_DVR\_TIME:时间参数

```
struct{
    DWORD    dwYear;
    DWORD    dwMonth;
    DWORD    dwDay;
    DWORD    dwHour;
    DWORD    dwMinute;
    DWORD    dwSecond;
}NET_DVR_TIME, *LPNET_DVR_TIME;
```

### Members

*dwYear*  
年

*dwMonth*  
月

*dwDay*  
日

*dwHour*  
时

*dwMinute*  
分

*dwSecond*  
秒

## 7.103 NET\_DVR\_TIME\_EX:时间参数

```
struct{
    WORD     wYear;
    BYTE     byMonth;
    BYTE     byDay;
    BYTE     byHour;
    BYTE     byMinute;
    BYTE     bySecond;
    BYTE     byRes;
}NET_DVR_TIME_EX, *LPNET_DVR_TIME_EX;
```

### Members

*wYear*  
年

*byMonth*  
月

*byDay*  
日

*byHour*

时  
*byMinute*  
 分  
*bySecond*  
 秒  
*byRes*  
 保留

## 7.104 NET\_DVR\_TIME\_V30:时间参数

```

struct{
  WORD    wYear;
  BYTE    byMonth;
  BYTE    byDay;
  BYTE    byHour;
  BYTE    byMinute;
  BYTE    bySecond;
  BYTE    byRes;
  WORD    wMilliSec;
  BYTE    byRes1[2];
}NET_DVR_TIME_V30, *LPNET_DVR_TIME_V30;
  
```

### Members

*wYear*  
 年  
*byMonth*  
 月  
*byDay*  
 日  
*byHour*  
 时  
*byMinute*  
 分  
*bySecond*  
 秒  
*byRes*  
 保留  
*wMilliSec*  
 毫秒  
*byRes1*  
 保留

## 7.105 NET\_DVR\_TPS\_ALARM\_V41:交通统计上传报警信息

```

struct{
    DWORD                dwSize;
    DWORD                dwRelativeTime;
    DWORD                dwAbsTime;
    NET_VCA_DEV_INFO    struDevInfo;
    NET_DVR_TPS_INFO_V41 struTPSInfo;
    BYTE                byMonitoringSiteID[MONITORSITE_ID_LEN];
    BYTE                byDeviceID[DEVICE_ID_LEN];
    DWORD                dwStartTime;
    DWORD                dwStopTime;
    BYTE                byRes[24];
}NET_DVR_TPS_ALARM_V41, *LPNET_DVR_TPS_ALARM_V41;

```

### Members

*dwSize*

结构体大小

*dwRelativeTime*

相对时标

*dwAbsTime*

绝对时标

*struDevInfo*

前端设备信息

*struTPSInfo*

交通参数统计信息

*byMonitoringSiteID*

监测点编号（路口编号、内部编号）

*byDeviceID*

设备编号

*dwStartTime*

开始统计时间

*dwStopTime*

结束统计时间

*byRes1*

保留

### Remarks

从绝对时标 *dwAbsTime* 解析得到“年月日时分秒”的算法如下所示：

```

#define GET_YEAR(_time_)    (((_time_)>>26) + 2000)
#define GET_MONTH(_time_)  (((_time_)>>22) & 15)
#define GET_DAY(_time_)    (((_time_)>>17) & 31)
#define GET_HOUR(_time_)   (((_time_)>>12) & 31)
#define GET_MINUTE(_time_) (((_time_)>>6) & 63)
#define GET_SECOND(_time_) (((_time_)>>0) & 63)

```

## 7.106 NET\_DVR\_TPS\_INFO\_V41:车道统计参数信息

```
struct{
    DWORD                dwLanNum;
    NET_DVR_LANE_PARAM_V41  struLaneParam[MAX_TPS_RULE];
    BYTE                 byRes[32];
}NET_DVR_TPS_INFO_V41, *LPNET_DVR_TPS_INFO_V41;
```

### Members

*dwLanNum*

交通参数的车道数目

*struLaneParam*

车道参数

*byRes*

保留

## 7.107 NET\_DVR\_TPS\_RULECFG\_V41:车道统计参数信息

```
struct{
    DWORD                dwSize;
    NET_DVR_ONE_TPS_RULE_V41  struOneTpsRule[MAX_TPS_RULE];
    BYTE                 byRes[128];
}NET_DVR_TPS_RULECFG_V41, *LPNET_DVR_TPS_RULECFG_V41;
```

### Members

*dwSize*

结构体大小

*struOneTpsRule*

单条交通参数统计规则参数，下标对应交通参数 ID

*byRes*

保留

## 7.108 NET\_DVR\_TRAFFIC\_DATA\_QUERY\_COND:交通数据查询条件

```
struct{
    DWORD                dwSize;
    DWORD                dwQueryCond;
    DWORD                dwChannel;
    NET_DVR_TIME_V30    struStartTime;
    NET_DVR_TIME_V30    struEndTime;
    char                 sLicense[MAX_LICENSE_LEN];
    DWORD                dwPlateType;
    DWORD                dwPlateColor;
    DWORD                dwVehicleColor;
```

```

DWORD          dwVehicleType;
DWORD          dwIllegalType;
DWORD          dwEventType;
DWORD          dwForensiceType;
WORD           wVehicleLogoRecog;
BYTE           byLaneNo;
BYTE           byDirection;
WORD           wMinSpeed;
WORD           wMaxSpeed;
BYTE           byDataType;
BYTE           byExecuteCtrl;
BYTE           byRes[254];
}NET_DVR_TRAFFIC_DATA_QUERY_COND,*LPNET_DVR_TRAFFIC_DATA_QUERY_COND;

```

## Members

### *dwSize*

结构体大小

### *dwQueryCond*

查询条件，按位表示，取值：0- 无效，1- 有效，定义如下所示：

bit0- 通道，bit1- 时间，bit2- 车牌号，bit3- 车牌类型，bit4- 车牌颜色，bit5- 车身颜色，bit6- 车辆类型，bit7- 车辆品牌，bit8- 车道号，bit9- 监测方向，bit10- 最低速度，bit11- 最高速度，bit12- 数据类型，bit13- 布控方式类型，bit14- 违法取证，bit15- 事件类型，bit16- 取证类型

### *dwChannel*

查询通道，按位表示，bit0 表示数字通道 01（通道号 33），bit1 表示数字通道 02（通道号 34），依次类推，取值：0- 无效，1- 有效。例如：

`dwChannel & 0x1==1` 表示查询数字通道 01

`dwChannel & 0x8==1` 表示查询数字通道 04

### *struStartTime*

开始时间

### *struEndTime*

结束时间

### *sLicense*

车牌号码，支持模糊查询（通配符是“\*”，GB2312 编码

### *dwPlateType*

车牌类型，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 未知（其他），bit1- 标准民用与军车，bit2- 02 式民用车牌，bit3- 武警车，bit4- 警车，bit5- 民用双行尾牌，bit6- 使馆车牌，bit7- 农用车，bit8- 摩托车

### *dwPlateColor*

车牌颜色，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 未知（其他），bit1- 黄色，bit2- 白色，bit3- 黑色，bit4- 绿色，bit5- 蓝色

### *dwVehicleColor*

车身颜色，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 未知（其他），bit1- 白色，bit2- 银色，bit3- 灰色，bit4- 黑色，bit5- 红色，bit6- 深蓝色，bit7- 蓝色，bit8- 黄色，bit9- 绿色，bit10- 棕色，bit11- 粉色，bit12- 紫色，bit13- 深灰色

### *dwVehicleType*

车辆类型，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

Bit0- 未知（其他），Bit1- 客车，Bit2- 大货车，Bit3- 轿车，Bit4- 面包车，Bit5- 小货车，Bit6- 行人，Bit7- 二轮车，Bit8- 三轮车，Bit9- SUV/MPV，Bit10- 中型客车

#### *dwIllegalType*

违法类型，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 其他（保留），bit1- 低速，bit2- 超速，bit3- 逆行，bit4- 闯红灯，bit5- 压车道线，bit6- 不按导向，bit7- 路口滞留，bit8- 机占非，bit9- 违法变道，bit10- 不按车道，bit11- 违反禁令，bit12- 路口停车，bit13- 绿灯停车，bit14- 未礼让行人，bit15- 违章停车，bit16- 违章掉头，bit17- 占用应急车道，bit18- 未系安全带

#### *dwEventType*

事件类型，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 其他（保留），bit1- 拥堵，bit2- 停车，bit3- 逆行，bit4- 行人，bit5- 抛洒物，bit6- 烟雾，bit7- 压线，bit8- 禁止名单，bit9- 超速，bit10- 变道，bit11- 掉头，bit12- 机占非，bit13- 加塞

#### *dwForensicType*

取证类型，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 其他（保留），bit1- 城市公路违法停车，bit2- 高速公路违法停车，bit3- 压线，bit4- 逆行，bit5- 违法变道，bit6- 机占非

#### *wVehicleLogoRecog*

车辆主品牌（单选）

#### *byLaneNo*

车道号（0~255，0 表示车道号未知）

#### *byDirection*

监测方向：1- 上行，2- 下行，3- 双向，4- 由东向西，5- 由南向北，6- 由西向东，7- 由北向南

#### *wMinSpeed*

最低速度，取值范围：0~999，单位：km/h

#### *wMaxSpeed*

最高速度，取值范围：0~999，单位：km/h

#### *byDataType*

数据类型（单选）：0- 卡口数据，1- 违法数据，2- 交通事件，3- 取证数据

#### *byExecuteCtrl*

布控：0- 允许名单，1- 禁止名单，0xff- 其他

#### *byRes*

保留，置为 0

## 7.109 NET\_DVR\_TRAFFIC\_DATA\_QUERY\_RESULT:交通数据查询结果

```
struct{
    DWORD                dwSize;
    DWORD                dwChannel;
    char                 sLicense[MAX_LICENSE_LEN];
    DWORD                dwPlateType;
    DWORD                dwPlateColor;
    DWORD                dwVehicleColor;
    DWORD                dwVehicleType;
```

```

DWORD          dwIllegalType;
DWORD          dwEventType;
DWORD          dwForensiceType;
WORD           wVehicleLogoRecog;
BYTE           byLaneNo;
BYTE           byDirection;
WORD           wSpeed;
BYTE           byDataType;
BYTE           byRes[253];
NET_DVR_TRAFFIC_PICTURE_PARAM struTrafficPic[MAX_TRAFFIC_PICTURE_NUM];
}NET_DVR_TRAFFIC_DATA_QUERY_RESULT,*LPNET_DVR_TRAFFIC_DATA_QUERY_RESULT;

```

## Members

### *dwSize*

结构体大小

### *dwChannel*

查询通道，按位表示，bit0 表示数字通道 01（通道号 33），bit1 表示数字通道 02（通道号 34），依次类推，取值：0- 无效，1- 有效。例如：

dwChannel & 0x1==1 表示数字通道 01 的查询结果

dwChannel & 0x8==1 表示数字通道 04 的查询结果

### *sLicense*

车牌号码，GB2312 编码

### *dwPlateType*

车牌类型，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 未知（其他），bit1- 标准民用车与军车，bit2- 02 式民用车牌，bit3- 武警车，bit4- 警车，bit5- 民用车双行尾牌，bit6- 使馆车牌，bit7- 农用车，bit8- 摩托车

### *dwPlateColor*

车牌颜色，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 未知（其他），bit1- 黄色，bit2- 白色，bit3- 黑色，bit4- 绿色，bit5- 蓝色

### *dwVehicleColor*

车身颜色，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 未知（其他），bit1- 白色，bit2- 银色，bit3- 灰色，bit4- 黑色，bit5- 红色，bit6- 深蓝色，bit7- 蓝色，bit8- 黄色，bit9- 绿色，bit10- 棕色，bit11- 粉色，bit12- 紫色，bit13- 深灰色

### *dwVehicleType*

车辆类型，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

Bit0- 未知（其他），Bit1- 客车，Bit2- 大货车，Bit3- 轿车，Bit4- 面包车，Bit5- 小货车，Bit6- 行人，Bit7- 二轮车，Bit8- 三轮车，Bit9- SUV/MPV，Bit10- 中型客车

### *dwIllegalType*

违法类型，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 其他（保留），bit1- 低速，bit2- 超速，bit3- 逆行，bit4- 闯红灯，bit5- 压车道线，bit6- 不按导向，bit7- 路口滞留，bit8- 机占非，bit9- 违法变道，bit10- 不按车道，bit11- 违反禁令，bit12- 路口停车，bit13- 绿灯停车，bit14- 未礼让行人，bit15- 违章停车，bit16- 违章掉头，bit17- 占用应急车道，bit18- 未系安全带

### *dwEventType*

事件类型，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 其他（保留），bit1- 拥堵，bit2- 停车，bit3- 逆行，bit4- 行人，bit5- 抛洒物，bit6- 烟雾，bit7- 压线，bit8- 禁止名单，bit9- 超速，bit10- 变道，bit11- 掉头，bit12- 机占非，bit13- 加塞

#### *dwForensicType*

事件类型，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 其他（保留），bit1- 城市公路违法停车，bit2- 高速公路违法停车，bit3- 压线，bit4- 逆行，bit5- 违法变道，bit6- 机占非

#### *wVehicleLogoRecog*

车辆主品牌

#### *byLaneNo*

车道号（0~255，0 表示车道号未知）

#### *byDirection*

监测方向：1- 上行，2- 下行，3- 双向，4- 由东向西，5- 由南向北，6- 由西向东，7- 由北向南

#### *wSpeed*

速度，取值范围：0~999，单位：km/h

#### *byDataType*

数据类型：0- 卡口数据，1- 违法数据，2- 交通事件，3- 取证数据

#### *byRes*

保留，置为 0

#### *struTrafficPic*

交通图片参数

## 7.110 NET\_DVR\_TRAFFIC\_FLOW\_QUERY\_COND: 交通流量查询条件

```
struct{
```

```
    DWORD           dwSize;
    DWORD           dwQueryCond;
    DWORD           dwChannel;
    DWORD           dwResChan[10];
    NET_DVR_TIME_V30 struStartTime;
    NET_DVR_TIME_V30 struEndTime;
    BYTE            byLaneNo;
    BYTE            byRes[255];
```

```
}NET_DVR_TRAFFIC_FLOW_QUERY_COND,*LPNET_DVR_TRAFFIC_FLOW_QUERY_COND;
```

### Members

#### *dwSize*

结构体大小

#### *dwQueryCond*

查询条件，按位表示，支持复选，取值：0- 无效，1- 有效，定义如下所示：

bit0- 通道，bit1- 时间，bit2- 车道号

#### *dwChannel*

查询通道，按位表示，bit0 表示数字通道 01（通道号 33），bit1 表示数字通道 02（通道号 34），依次类推，取值：0- 无效，1- 有效。例如：

`dwChannel & 0x1==1` 表示查询数字通道 01

`dwChannel & 0x8==1` 表示查询数字通道 04

*dwResChan*

预留可扩展的通道号

*struStartTime*

开始时间

*struEndTime*

结束时间

*byLaneNo*

车道号（0~255，0 表示车道号未知）

*byRes*

保留，置为 0

## 7.111 NET\_DVR\_TRAFFIC\_FLOW\_QUERY\_RESULT: 交通流量查询结果

```
struct{
```

```
    DWORD                dwSize;
```

```
    NET\_DVR\_TIME\_V30     struStartTime;
```

```
    NET\_DVR\_TIME\_V30     struEndTime;
```

```
    DWORD                dwChannel;
```

```
    DWORD                dwFlow;
```

```
    BYTE                 byLaneNo;
```

```
    BYTE                 byRes[511];
```

```
}NET_DVR_TRAFFIC_FLOW_QUERY_RESULT,*LPNET_DVR_TRAFFIC_FLOW_QUERY_RESULT;
```

### Members

*dwSize*

结构体大小

*struStartTime*

开始时间

*struEndTime*

结束时间

*dwChannel*

查询通道，按位表示，bit0 表示数字通道 01（通道号 33），bit1 表示数字通道 02（通道号 34），依次类推，取值：0- 无效，1- 有效。例如：

$dwChannel \& 0x1 == 1$  表示数字通道 01 的查询结果

$dwChannel \& 0x8 == 1$  表示数字通道 04 的查询结果

*dwFlow*

车流量

*byLaneNo*

车道号（0~255，0 表示车道号未知）

*byRes*

保留，置为 0

## 7.112 NET\_DVR\_TRAFFIC\_PICTURE\_PARAM:交通图片参数

```

struct{
    NET\_DVR\_TIME\_V30    struRelativeTime;
    NET\_DVR\_TIME\_V30    struAbsTime;
    char                szPicName[PICTURE_NAME_LEN];
    BYTE                byPicType;
    BYTE                byRes[63];
}NET_DVR_TRAFFIC_PICTURE_PARAM,*LPNET_DVR_TRAFFIC_PICTURE_PARAM;

```

### Members

*struRelativeTime*

抓拍相对时标

*struAbsTime*

抓拍绝对时标

*szPicName*

图片名称

*byPicType*

图片类型：0- 车牌图，1- 抓拍原图，2- 合成图，3- 特写图

*byRes*

保留，置为 0

### Remarks

查找到图片名称之后通过接口 [NET\\_DVR\\_GetPicture\\_V30](#) 进行下载。

## 7.113 NET\_DVR\_USER\_INFO\_V30:单用户参数

```

struct{
    BYTE                sUserName[NAME_LEN];
    BYTE                sPassword[PASSWD_LEN];
    BYTE                byLocalRight[MAX_RIGHT];
    BYTE                byRemoteRight[MAX_RIGHT];
    BYTE                byNetPreviewRight[MAX_CHANNUM_V30];
    BYTE                byLocalPlaybackRight[MAX_CHANNUM_V30];
    BYTE                byNetPlaybackRight[MAX_CHANNUM_V30];
    BYTE                byLocalRecordRight[MAX_CHANNUM_V30];
    BYTE                byNetRecordRight[MAX_CHANNUM_V30];
    BYTE                byLocalPTZRight[MAX_CHANNUM_V30];
    BYTE                byNetPTZRight[MAX_CHANNUM_V30];
    BYTE                byLocalBackupRight[MAX_CHANNUM_V30];
    NET\_DVR\_IPADDR     struUserIP;
    BYTE                byMACAddr[MACADDR_LEN];
    BYTE                byPriority;
    BYTE                byAlarmOnRight;
    BYTE                byAlarmOffRight;
}

```

```

    BYTE                byBypassRight;
    BYTE                byRes[14];
}NET_DVR_USER_INFO_V30,*LPNET_DVR_USER_INFO_V30;

```

### Members

#### *sUserName*

用户名

#### *sPassword*

密码

#### *byLocalRight*

本地操作权限，参数取值为 1 表示使能：

数组 0—本地控制云台

数组 1—本地手动录象

数组 2—本地回放

数组 3—本地设置参数

数组 4—本地查看状态、日志

数组 5—本地高级操作（升级，硬盘管理（格式化、设置硬盘属性、设置盘组、阵列扩容、RAID 固件升级））

数组 6—本地查看参数

数组 7—本地管理模拟和 IP camera

数组 8—本地备份

数组 9—本地关机/重启

#### *byRemoteRight*

远程操作权限，参数取值为 1 表示使能：

数组 0—远程控制云台

数组 1—远程手动录象

数组 2—远程回放

数组 3—远程设置参数（恢复默认参数，写日志）

数组 4—远程查看状态、日志

数组 5—远程高级操作（升级，硬盘管理（格式化、设置硬盘属性、设置盘组、阵列扩容、RAID 固件升级），JPEG 抓图，前面板锁定与解锁）

数组 6—远程发起语音对讲

数组 7—远程预览

数组 8—远程请求报警上传、报警输出

数组 9—远程控制，本地输出

数组 10—远程控制串口

数组 11—远程查看参数

数组 12—远程管理模拟和 IP camera

数组 13—远程关机/重启

#### *byNetPreviewRight*

远程可以预览的通道：1-有权限，0-无权限

#### *byLocalPlaybackRight*

本地可以回放的通道：1-有权限，0-无权限

#### *byNetPlaybackRight*

远程可以回放的通道：1-有权限，0-无权限

**byLocalRecordRight**

本地可以录像的通道：1-有权限，0-无权限

**byNetRecordRight**

远程可以录像的通道：1-有权限，0-无权限

**byLocalPTZRight**

本地可以控制 PTZ 的通道：1-有权限，0-无权限

**byNetPTZRight**

远程可以控制 PTZ 的通道：1-有权限，0-无权限

**byLocalBackupRight**

本地备份权限通道：1-有权限，0-无权限

**struUserIP**

用户 IP 地址(为 0 时表示允许任何地址)

**byMACAddr**

物理地址

**byPriority**

优先级：0xff-无，0-低，1-中，2-高

无（表示不支持优先级的设置）

低（默认权限：包括本地和远程回放，本地和远程查看日志和状态，本地和远程关机/重启）

中（包括本地和远程控制云台，本地和远程手动录像，本地和远程回放，语音对讲和远程预览，本地备份，本地/远程关机/重启）

高（管理员）

**byAlarmOnRight**

报警输入口布防权限

**byAlarmOffRight**

报警输入口撤防权限

**byBypassRight**

报警输入口旁路权限

**byRes**

保留，置为 0

## 7.114 NET\_DVR\_USER\_LOGIN\_INFO:用户登录参数

```
struct{
    char                sDeviceAddress[NET_DVR_DEV_ADDRESS_MAX_LEN];
    BYTE                byRes1;
    WORD                wPort;
    char                sUserName[NET_DVR_LOGIN_USERNAME_MAX_LEN];
    char                sPassword[NET_DVR_LOGIN_PASSWD_MAX_LEN];
    fLoginResultCallBack cbLoginResult;
    void                *pUser;
    BOOL                bUseAsynLogin;
    BYTE                byRes2[128];
}NET_DVR_USER_LOGIN_INFO,*LPNET_DVR_USER_LOGIN_INFO;
```

### Members

**sDeviceAddress**

设备地址，IP 或者普通域名

**byRes1**

保留，设为 0

**wPort**

设备端口号，例如：8000

**sUserName**

登录用户名，例如：admin

**sPassword**

登录密码，例如：12345

**cbLoginResult**

登录状态回调函数，bUseAsynLogin 为 1 时有效

**pUser**

用户数据

**bUseAsynLogin**

是否异步登录：0- 否，1- 是

**byRes2**

保留，置为 0

**Callback Function**

```
typedef void(CALLBACK *fLoginResultCallBack)(LONG IUserID, DWORD dwResult,
LPNET\_DVR\_DEVICEINFO\_V30 lpDeviceInfo, void *pUser);
```

**Callback Function Parameters****IUserID**

[out] 用户 ID，NET\_DVR\_Login\_V40 的返回值

**dwResult**

[out] 登录状态：0- 异步登录失败，1- 异步登录成功

**lpDeviceInfo**

[out] 设备信息，设备序列号、通道、能力等参数

**pUser**

[out] 用户数据

**7.115 NET\_DVR\_USER\_V30:用户参数**

```
struct{
    DWORD dwSize;
    NET\_DVR\_USER\_INFO\_V30 struUser[MAX_USERNUM_V30];
}NET_DVR_USER_V30,*LPNET_DVR_USER_V30;
```

**Members****dwSize**

结构体大小

**struUser**

用户信息参数

## 7.116 NET\_DVR\_VEHICLE\_INFO:车辆信息参数

```

struct{
    DWORD    dwIndex;
    BYTE     byVehicleType;
    BYTE     byColorDepth;
    BYTE     byColor;
    BYTE     byRes1;
    WORD     wSpeed;
    WORD     wLength;
    BYTE     byIllegalType;
    BYTE     byVehicleLogoRecog;
    BYTE     byVehicleSubLogoRecog;
    BYTE     byVehicleModel;
    BYTE     byCustomInfo[16];
    WORD     wVehicleLogoRecog;
    BYTE     byRes3[14];
}NET_DVR_VEHICLE_INFO, *LPNET_DVR_VEHICLE_INFO;

```

### Members

#### *dwIndex*

车辆序号

#### *byVehicleType*

车辆类型，0-其他车辆，1-小型车，2-大型车，3- 行人触发，4- 二轮车触发，5- 三轮车触发，6- 机动车触发

#### *byColorDepth*

车身颜色深浅，0-深色，1-浅色

#### *byColor*

车身颜色，0-其他色，1-白色，2-银色，3-灰色，4-黑色，5-红色，6-深蓝，7-蓝色，8-黄色，9-绿色，10-棕色，11-粉色，12-紫色

#### *byRes1*

保留

#### *wSpeed*

车辆速度，单位 km/h

#### *wLength*

车身长度

#### *byIllegalType*

0-正常：1-低速，2-超速，3-逆行，4-违反交通灯指示，5-压车道线，6-不按导向，7-路口滞留，8-机占非，9-违法变道，10-机动车违反规定使用专用车道，11-黄牌车禁限，12-路口停车，13-绿灯停车，14-未礼让行人。对于 ITS 终端服务器，该参数无效，违法类型通过 [NET ITS PLATE RESULT](#) 中的 *wIllegalType* 进行判断。

#### *byVehicleLogoRecog*

车辆主品牌

#### *byVehicleSubLogoRecog*

车辆子品牌  
*byVehicleModel*  
 车辆子品牌年款  
*byCustomInfo*  
 自定义信息  
*wVehicleLogoRecog*  
 车辆主品牌(该字段兼容 *byVehicleLogoRecog*)  
*byRes3*  
 保留

## 7.117 NET\_DVR\_VEHICLE\_RECOG\_CFG: 车辆二次识别任务配置

```

struct{
    DWORD          dwSize;
    char           sDataIndex[64];
    WORD          wTaskNo;
    BYTE          byRes1[2];
    NET\_VCA\_RECT struPlateRect;
    char          sLicense[MAX_LICENSE_LEN];
    DWORD        dwRecogOperate;
    DWORD        dwDataUploadType;
    BYTE         byRes[131];
    BYTE         byPicDataType;
    char         sPicDataPath[256];
}NET_DVR_VEHICLE_RECOG_CFG, *LPNET_DVR_VEHICLE_RECOG_CFG;
  
```

### Members

*dwSize*

结构体大小

*sDataIndex*

数据流水号

*wTaskNo*

任务处理号

*byRes1*

保留，置为 0

*struPlateRect*

车牌位置，归一化值，相对于场景图

*sLicense*

车牌号码

*dwRecogOperate*

车辆图片二次识别操作，按位表示：

bit0- 车牌识别：0-不启用识别，1-启用识别

bit1- 车辆类型识别：0-不启用识别，1-启用识别

bit2- 车身颜色识别：0-不启用识别，1-启用识别

bit3- 车辆品牌识别：0-不启用识别，1-启用识别

- bit4- 遮挡板识别(主驾驶): 0-不启用识别, 1-启用识别
- bit5- 安全带识别(主驾驶): 0-不启用识别, 1-启用识别
- bit6- 遮挡板识别(副驾驶): 0-不启用识别, 1-启用识别
- bit7- 安全带识别(副驾驶): 0-不启用识别, 1-启用识别
- bit8- 车辆图片叠加使能: 0-不启用识别, 1-启用识别
- bit9- 上传车辆图片使能(通过 SDK): 0-不启用识别, 1-启用识别
- bit10-上传车辆图片使能(通过 URL): 0-不启用识别, 1-启用识别

***dwDataUploadType***

车辆图片数据上传, 按位表示:

- bit0- 车牌图片: 0-不上传, 1-上传
- bit1- 车辆图片: 0-不上传, 1-上传
- bit2- 人脸子图(主驾驶): 0-不上传, 1-上传
- bit3- 人脸子图(副驾驶): 0-不上传, 1-上传
- bit4- 安全带子图(主驾驶): 0-不上传, 1-上传
- bit5- 安全带子图(副驾驶): 0-不上传, 1-上传

***byRes***

保留, 置为 0

***byPicDataType***

图片数据类型: 0- 数据直接上传, 1- 云存储服务 URL

***sPicDataPath***

图片数据:

byPicDataType 为 0 时, 表示本地 PC 上的图片路径, 包括文件名的绝对路径, 例如: "C:/test.jpg"

byPicDataType 为 1 时, 表示云存储服务 URL

**Remarks**

- 车辆二次识别功能, 即可以将图片从本地或者云存储服务上传到设备, 进行二次检测, 识别图片里面的车辆、车牌等信息, 上传的图片格式需要设备支持, 目前设备只支持 jpg 格式。
- 该结构体中的 wTaskNo, 与布防连接([NET\\_DVR\\_SETUPALARM\\_PARAM](#))和上传检测结果信息([NET\\_DVR\\_VEHICLE\\_RECOG\\_RESULT](#))里面的任务号都是对应的, 例如: 布防链接中 wTaskNo==1, 任务 A 中 wTaskNo==1, 结果信息回调 wTaskNo==1(该信息回调只在布防中 wTaskNo 为 1 或者 0 的链接中回调)。

**7.118 NET\_DVR\_VEHICLE\_RECOG\_COND: 车辆二次识别任务提交条件**

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    BYTE     byRes[64];
}NET_DVR_VEHICLE_RECOG_COND, *LPNET_DVR_VEHICLE_RECOG_COND;
```

**Members*****dwSize***

结构体大小

***dwChannel***

通道号

***byRes***

保留, 置为 0

## 7.119 NET\_DVR\_VEHICLE\_RECOG\_RESULT: 车辆二次识别结果信息

```

struct{
    DWORD          dwSize;
    char           sDataIndex[64];
    WORD          wTaskNo;
    BYTE          byRes[2];
    NET\_VCA\_RECT  struPlateRect;
    char          sLicense[MAX_LICENSE_LEN];
    BYTE          byVehicleType;
    BYTE          byColorDepth;
    BYTE          byColor;
    BYTE          byVehicleLogoRecog;
    BYTE          byVehicleSubLogoRecog;
    BYTE          byPilotSafebelt;
    BYTE          byCopilotSafebelt;
    BYTE          byPilotSunVisor;
    BYTE          byCopilotSunVisor;
    BYTE          byRes1[254];
    BYTE          byDataType;
    DWORD         dwPicType;
    BYTE          *pVehicleBuffer;
    DWORD         dwVehicleBufferLen;
    BYTE          *pPlateBuffer;
    DWORD         dwPlateBufferLen;
    BYTE          *pPilotFaceBuffer;
    DWORD         dwPilotFaceBufferLen;
    BYTE          *pCopilotFaceBuffer;
    DWORD         dwCopilotFaceBufferLen;
    BYTE          *pPilotSafebeltBuffer;
    DWORD         dwPilotSafebeltBufferLen;
    BYTE          *pCopilotSafebeltBuffer;
    DWORD         dwCopilotSafebeltBufferLen;
    BYTE          byRes2[128];
}NET_DVR_VEHICLE_RECOG_RESULT, *LPNET_DVR_VEHICLE_RECOG_RESULT;

```

### Members

#### *dwSize*

结构体大小

#### *sDataIndex*

数据流水号

#### *wTaskNo*

任务处理号，与布防连接([NET\\_DVR\\_SETUPALARM\\_PARAM](#))和车辆二次识别任务提交([NET\\_DVR\\_VEHICLE\\_RECOG\\_CFG](#))里面的 wTaskNo 都是对应的

**byRes**

保留，置为 0

**struPlateRect**

车牌位置，归一化值，相对于场景图

**sLicense**

车牌号码

**byVehicleType**

车型识别：0- 未知，1- 客车(大型)，2- 货车(大型)，3- 轿车(小型)，4- 面包车，5- 小货车，6- 行人，7- 二轮车，8- 三轮车，9- SUV/MPV，10- 中型客车

**byColorDepth**

车身颜色深浅：0- 深色，1- 浅色

**byColor**

车身颜色：0-其他色，1-白色，2-银色，3-灰色，4-黑色，5-红色，6-深蓝，7-蓝色，8-黄色，9-绿色，10-棕色，11-粉色，12-紫色，13-深灰，14-青色，0xff-未进行车身颜色识别

**byVehicleLogoRecog**

汽车品牌主类型，定义详见 **VLR\_VEHICLE\_CLASS**

**byVehicleSubLogoRecog**

汽车品牌次类型，根据不同的主类型，次类型取值定义不同，详见“Remarks”说明

**byPilotSafebelt**

主驾驶员是否系安全带：0- 未知，1- 系安全带，2- 未系安全带

**byCopilotSafebelt**

副驾驶员是否系安全带(保留，暂不支持)：0- 未知，1- 系安全带，2- 未系安全带

**byPilotSunVisor**

主驾驶是否打开遮阳板：0- 未知，1- 未打开遮阳板，2- 打开遮阳板

**byCopilotSunVisor**

副驾驶是否打开遮阳板：0- 未知，1- 未打开遮阳板，2- 打开遮阳板

**byRes1**

保留，置为 0

**byDataType**

数据上传方式：0-数据直接上传；1-云存储服务器 URL（原先的图片数据变成 URL 数据，图片长度变成 URL 长度）

**dwPicType**

上传图片类型信息，0 表示无图片信息，其他取值按位表示：

bit0- 车辆图：0- 不上传，1- 上传

bit1- 车牌图：0- 不上传，1- 上传

bit2- 人脸子图(主驾驶)：0- 不上传，1- 上传

bit3- 人脸子图(副驾驶)：0- 不上传，1- 上传

bit4- 安全带识别(主驾驶)：0- 不上传，1- 上传

bit5- 安全带识别(副驾驶)：0- 不上传，1- 上传

**pVehicleBuffer**

车辆图片数据指针

**dwVehicleBufferLen**

车辆图片数据长度

**pPlateBuffer**

车牌图片数据指针

*dwPlateBufferLen*

车牌图片数据长度

*pPilotFaceBuffer*

人脸子图(主驾驶)图片数据指针

*dwPilotFaceBufferLen*

人脸子图(主驾驶)图片数据长度

*pCopilotFaceBuffer*

人脸子图(副驾驶)图片数据指针

*dwCopilotFaceBufferLen*

人脸子图(副驾驶)图片数据长度

*pPilotSafebeltBuffer*

安全带识别(主驾驶)图片数据指针

*dwPilotSafebeltBufferLen*

安全带识别(主驾驶)图片数据长度

*pCopilotSafebeltBuffer*

安全带识别(副驾驶)图片数据指针

*dwCopilotSafebeltBufferLen*

安全带识别(副驾驶)图片数据长度

*byRes2*

保留，置为 0

### Remarks

根据不同的主类型，汽车品牌的次类型取值定义如下表所示，列表未列出的主类型暂无次类型。

byVehicleLogoRecog	品牌	byVehicleSubLogoRecog
1	大众	VSB_VOLKSWAGEN_CLASS
2	别克	VBR_BUICK_CLASS
3	宝马	VSB_BMW_CLASS
4	本田	VSB_HONDA_CLASS
5	标致	VSB_PEUGEOT_CLASS
6	丰田	VSB_TOYOTA_CLASS
7	福特	VSB_FORD_CLASS
8	日产	VSB_NISSAN_CLASS
9	奥迪	VSB_AUDI_CLASS
10	马自达	VSB_MAZDA_CLASS
11	雪佛兰	VSB_CHEVROLET_CLASS
12	雪铁龙	VSB_CITROEN_CLASS
13	现代	VSB_HYUNDAI_CLASS
14	奇瑞	VSB_CHERY_CLASS
15	起亚	VSB_KIA_CLASS

16	荣威	VSBS_ROEWE_CLASS
17	三菱	VSBS_MITSUBISHI_CLASS
18	斯柯达	VSBS_SKODA_CLASS
19	吉利	VSBS_GEELY_CLASS
20	中华	VSBS_ZHONGHUA_CLASS
21	沃尔沃	VSBS_VOLVO_CLASS
22	雷克萨斯	VSBS_LEXUS_CLASS
23	菲亚特	VSBS_FIAT_CLASS
24	吉利帝豪	VSBS_EMGRAND_CLASS
25	东风	VSBS_DONGFENG_CLASS
26	比亚迪	VSBS_BYD_CLASS
27	铃木	VSBS_SUZUKI_CLASS
28	金杯	VSBS_JINBEI_CLASS
29	海马	VSBS_HAIMA_CLASS
30	五菱	VSBS_SGMW_CLASS
31	江淮	VSBS_JAC_CLASS
32	斯巴鲁	VSBS_SUBARU_CLASS
33	吉利英伦	VSBS_ENGLON_CLASS
34	长城	VSBS_GREATWALL_CLASS
35	哈飞	VSBS_HAFEI_CLASS
36	庆铃(五十铃)	VSBS_ISUZU_CLASS
37	东南	VSBS_SOUFAST_CLASS
38	长安	VSBS_CHANA_CLASS
39	福田	VSBS_FOTON_CLASS
40	夏利	VSBS_XIALI_CLASS
41	奔驰	VSBS_BENZ_CLASS
42	一汽	VSBS_FAW_CLASS
43	依维柯	VSBS_NAVECO_CLASS
44	力帆	VSBS_LIFAN_CLASS
45	奔腾	VSBS_BESTURN_CLASS
47	雷诺	VSBS_RENAULT_CLASS
48	江铃	VSBS_JMC_CLASS
49	名爵	VSBS_MG_CLASS
50	凯马 KAMA	VSBS_KAMA_CLASS

51	众泰	VSZ_ZOTYE_CLASS
52	昌河	VSZ_CHANGHE_CLASS
55	苏州金龙	VSZ_SZKINGLONG_CLASS
56	海格	VSZ_HIGER_CLASS
57	宇通	VSZ_YUTONG_CLASS
58	中国重汽	VSZ_CNHTC_CLASS
59	北奔重汽	VSZ_BEIBEN_CLASS
60	华菱星马	VSZ_XINGMA_CLASS
61	跃进	VSZ_YUEJIN_CLASS
62	黄海	VSZ_HUANGHAI_CLASS
65	保时捷	VSZ_PORSCHE_CLASS
66	凯迪拉克	VSZ_CADILLAC_CLASS
67	英菲尼迪	VSZ_INFINITI_CLASS
68	吉利全球鹰	VSZ_GLEAGLE_CLASS
69	吉普	VSZ_JEEP_CLASS
70	路虎	VSZ_LANDROVER_CLASS
71	长丰猎豹	VSZ_CHANGFENG_CLASS
73	时代汽车	VSZ_ERA_CLASS
75	长安轿车	VSZ_EADO_CLASS
76	陕汽重卡	VSZ_SHANQI_CLASS
81	安凯	VSZ_ANKAI_CLASS
82	申龙	VSZ_SHENLONG_CLASS
83	大宇	VSZ_DAEWOO_CLASS
86	中通	VSZ_ZHONGTONG_CLASS
87	宝骏	VSZ_BAOJUN_CLASS
88	北汽威旺	VSZ_BQWEIWANG_CLASS
89	广汽传祺	VSZ_TRUMPCHE_CLASS
90	陆风	VSZ_LUFENG_CLASS
92	北京	VSZ_BEIJING_CLASS
94	威麟	VSZ_WEILIN_CLASS
95	欧宝	VSZ_OPEL_CLASS
96	开瑞	VSZ_KARRY_CLASS
97	华普	VSZ_SMA_CLASS
103	讴歌	VSZ_OGA_CLASS

104	启辰	VSBS_VENUCIA_CLASS
107	北汽制造	VSBS_BAW_CLASS
108	纳智捷	VSBS_LUXGEN_CLASS
109	野马	VSBS_YEMA_CLASS
110	中兴	VSBS_ZTE_CLASS
112	克莱斯勒	VSBS_CHRYSLER_CLASS
113	广汽吉奥	VSBS_GONOW_CLASS
115	瑞麟	VSBS_RUILIN_CLASS
117	捷豹	VSBS_GAGUAR_CLASS
119	唐骏欧铃	VSBS_TKING_CLASS
121	福迪	VSBS_FODAY_CLASS
122	莲花	VSBS_LOTUS_CLASS
124	双环	VSBS_SHUANGHUAN_CLASS
128	永源	VSBS_JONWAY_CLASS
136	江南	VSBS_JIANGNAN_CLASS
144	道奇	VSBS_DS_CLASS
155	大运汽车	VSBS_DAYUN_CLASS
167	北方客车	VSBS_BEIFANG_CLASS
176	九龙	VSBS_JIULONG_CLASS
191	宾利	VSBS_BINLI_CLASS
201	舒驰客车	VSBS_SHUCHI_CLASS
230	红旗	VSBS_HONGQI_CLASS

## 7.120 NET\_DVR\_VEHICLE\_RECOG\_TASK\_COND: 车辆二次识别任务信息

### 查询条件

```

struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    char     sDataIndex[64];
    WORD     wTaskNo;
    BYTE     byTask;
    BYTE     byRes[125];
}NET_DVR_VEHICLE_RECOG_TASK_COND, *LPNET_DVR_VEHICLE_RECOG_TASK_COND;

```

#### Members

*dwSize*

结构体大小

*dwChannel*

通道号

*sDataIndex*

数据流水号，0 表示查询全部任务

*wTaskNo*

任务处理号，0 表示查询全部任务

*byTask*

需要查询的任务状态，按位表示：

bit0- 完成状态的任务：0-不查询，1-查询

bit1- 执行中的任务：0-不查询，1-查询

bit2- 等待中的任务：0-不查询，1-查询

*byRes*

保留，置为 0

## 7.121 NET\_DVR\_VEHICLE\_RECOG\_TASK\_INFO:车辆二次识别任务信息

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    BYTE     sDataIndex[64];
    WORD     wTaskNo;
    WORD     wTaskProgress;
    BYTE     byTaskState;
    BYTE     byRes1[3];
    DWORD    dwRecogOperate;
    BYTE     byRes[128];
}NET_DVR_VEHICLE_RECOG_TASK_INFO, *LPNET_DVR_VEHICLE_RECOG_TASK_INFO;
```

### Members

*dwSize*

结构体大小

*dwChannel*

通道号

*sDataIndex*

数据流水号

*wTaskNo*

任务处理号

*wTaskProgress*

任务执行进度（实际进度\*1000），例如：wTaskProgress 为 1000 时，实际进度为 1，也就是进度为 100%

*byTaskState*

任务状态：0- 等待中，1- 执行中，2- 完成

*byRes1*

保留，置为 0

**dwRecogOperate**

车辆图片二次识别操作，按位表示：

bit0- 车牌识别：0-不启用识别，1-启用识别

bit1- 车辆类型识别：0-不启用识别，1-启用识别

bit2- 车身颜色识别：0-不启用识别，1-启用识别

bit3- 车辆品牌识别：0-不启用识别，1-启用识别

bit4- 遮挡板识别(主驾驶)：0-不启用识别，1-启用识别

bit5- 安全带识别(主驾驶)：0-不启用识别，1-启用识别

bit6- 遮挡板识别(副驾驶)：0-不启用识别，1-启用识别

bit7- 安全带识别(副驾驶)：0-不启用识别，1-启用识别

bit8- 车辆图片叠加使能：0-不启用识别，1-启用识别

bit9- 上传车辆图片使能(通过 SDK)：0-不启用识别，1-启用识别

bit10-上传车辆图片使能(通过 URL)：0-不启用识别，1-启用识别

**byRes**

保留，置为 0

**7.122 NET\_DVR\_VIDEOEFFECT:视频参数**

```
struct{
    BYTE        byBrightnessLevel;
    BYTE        byContrastLevel;
    BYTE        bySharpnessLevel;
    BYTE        bySaturationLevel;
    BYTE        byHueLevel;
    BYTE        byEnableFunc;
    BYTE        byLightInhibitLevel;
    BYTE        byGrayLevel;
}NET_DVR_VIDEOEFFECT, *LPNET_DVR_VIDEOEFFECT;
```

**Members****byBrightnessLevel**

亮度，取值范围[0,100]

**byContrastLevel**

对比度，取值范围[0,100]

**bySharpnessLevel**

锐度，取值范围[0,100]

**bySaturationLevel**

饱和度，取值范围[0,100]

**byHueLevel**

色度，取值范围[0,100]，保留

**byEnableFunc**

使能，按位表示，bit0-SMART IR(防过曝)，bit1-低照度，bit2-强光抑制使能，值：0-否，1-是。例如：  
byEnableFunc&0x2==1 表示使能低照度功能。

**byLightInhibitLevel**

强光抑制等级，取值范围：[1,3]

*byGrayLevel*

灰度值域:0-[0,255], 1-[16,235]

## 7.123 NET\_DVR\_VOD\_PARA:录像回放参数

```
struct{
    DWORD                dwSize;
    NET_DVR_STREAM_INFO struIDInfo;
    NET_DVR_TIME        struBeginTime;
    NET_DVR_TIME        struEndTime;
    HWND                hWnd;
    BYTE                byDrawFrame;
    BYTE                byVolumeType;
    BYTE                byVolumeNum;
    BYTE                byRes1;
    DWORD                dwFileIndex;
    BYTE                byRes2[24];
}NET_DVR_VOD_PARA,*LPNET_DVR_VOD_PARA;
```

### Members

*dwSize*

结构体大小

*struIDInfo*

流 ID 信息

*struBeginTime*

开始时间

*struEndTime*

结束时间

*hWnd*

回放的窗口句柄，若置为空，SDK 仍能收到码流数据，但不解码显示

*byDrawFrame*

是否抽帧：0- 不抽帧，1- 抽帧

*byVolumeType*

0-普通录像卷，1-存档卷，适用于 CVR 设备，普通卷用于通道录像，存档卷用于备份录像

*byVolumeNum*

存档卷号

*byRes1*

保留，置为 0

*dwFileIndex*

存档卷上的录像文件索引，搜索存档卷录像时返回的值

*byRes2*

保留，置为 0

## 7.124 NET\_DVR\_WDR:宽动态参数

```
struct{
    BYTE        byWDREnabled;
    BYTE        byWDRLevel1;
    BYTE        byWDRLevel2;
    BYTE        byWDRContrastLevel;
    BYTE        byRes[16];
}NET_DVR_WDR, *LPNET_DVR_WDR;
```

### Members

#### *byWDREnabled*

宽动态是否启用，0-不启用，1-启用，2-自动

#### *byWDRLevel1*

宽动态等级 1: 0~F

#### *byWDRLevel2*

宽动态等级 2: 0~F

#### *byWDRContrastLevel*

宽动态对比度: 0~100

#### *byRes*

保留

## 7.125 NET\_DVR\_WHITEBALANCE:白平衡参数

```
struct{
    BYTE        byWhiteBalanceMode;
    BYTE        byWhiteBalanceModeRGain;
    BYTE        byWhiteBalanceModeBGain;
    BYTE        byRes[5];
}NET_DVR_WHITEBALANCE, *LPNET_DVR_WHITEBALANCE;
```

### Members

#### *byWhiteBalanceMode*

0-手动白平衡，1-自动白平衡 1（范围小），2-自动白平衡 2（范围宽，2200K-15000K），3-锁定白平衡，4-白炽灯，5-暖光灯，6-自然光，7-日光灯

#### *byWhiteBalanceModeRGain*

手动白平衡时有效，手动白平衡 R 增益

#### *byWhiteBalanceModeBGain*

手动白平衡时有效，手动白平衡 B 增益

#### *byRes*

保留

## 7.126 NET\_ITS\_ECT\_BLOCKLIST: 车辆名单报警信息

```

struct{
    DWORD            dwSize;
    DWORD            dwChannel;
    BYTE             byLogicalLaneNo;
    BYTE             byRes1[3];
    BYTE             byLaneName[NAME_LEN];
    NET\_DVR\_PLATE\_INFO struPlateInfo;
    BYTE             byRes2[256];
}NET_ITS_ECT_BLOCKLIST, *LPNET_ITS_ECT_BLOCKLIST;

```

### Members

*dwSize*  
结构体大小

*dwChannel*  
通道号

*byLogicalLaneNo*  
逻辑车道号

*byRes1*  
保留，置为 0

*byLaneName*  
车道名称

*struPlateInfo*  
车牌信息结构

*byRes2*  
保留，置为 0

## 7.127 NET\_ITS\_ECT\_CHANNELSTATE: 终端通道状态信息

```

struct{
    DWORD            dwSize;
    BYTE             byRecordStatic;
    BYTE             bySignalStatic;
    BYTE             byHardwareStatic;
    BYTE             byChannelArmState;
    DWORD            dwChannel;
    DWORD            dwBitRate;
    DWORD            dwLinkNum;
    NET\_DVR\_IPADDR struClientIP[MAX_LINK];
    DWORD            dwIPLinkNum;
    BYTE             byExceedMaxLink;
    BYTE             byRes[139];
}NET_ITS_ECT_CHANNELSTATE, *LPNET_ITS_ECT_CHANNELSTATE;

```

**Members***dwSize*

结构体大小

*byRecordStatic*

通道是否在录像：0—不录像；1—录像

*bySignalStatic*

连接的信号状态：0—正常，1—信号丢失

*byHardwareStatic*

通道硬件状态：0—正常，1—异常（例如 DSP 异常）

*byChannelArmState*

接入 IPC 的布防状态

*dwChannel*

通道号

*dwBitRate*

实际码率

*dwLinkNum*

连接的客户端个数

*struClientIP*

连接的客户端 IP 地址

*dwIPLinkNum*

如果该通道为 IP 接入，表示 IP 接入当前的连接数

*byExceedMaxLink*

是否超出了单路 128 路连接数 0 - 未超出, 1-超出

*byRes*

保留，置为 0

## 7.128 NET\_ITS\_ECT\_WORKSTATE\_COMMAND: 出入口终端工作状态获取的条件结构

```

struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    BYTE     byRes[256];
}NET_ITS_ECT_WORKSTATE_COMMAND, *LPNET_ITS_ECT_WORKSTATE_COMMAND;

```

**Members***dwSize*

结构体大小

*dwChannel*

获取出入口终端服务器自身的工作状态时，该通道号设为 0xffffffff；如果出入口终端服务器级联，获取下级服务器状态时，为 IP 通道，通道号从 33 开始。

*byRes*

保留，置为 0

## 7.129 NET\_ITS\_ECTWORKSTATE: 出入口终端工作状态信息

```

struct{
    DWORD                dwSize;
    BYTE                 byDevName[NAME_LEN];
    DWORD                dwRunTime;
    NET_ITS_TRAFFIC_DATA_INFO struTrafficDataInfo[2];
    DWORD                dwMemoryUsage;
    DWORD                dwCpuUsage;
    DWORD                dwDevTemperature;
    DWORD                dwDeviceStatic;
    NET_DVR_DISKSTATE   struHardDiskStatic[MAX_DISKNUM_V30];
    BYTE                 byAlarmInStatic[MAX_ALARMIN_V30];
    BYTE                 byAlarmOutStatic[MAX_ALARMOUT_V30];
    DWORD                dwLocalDisplay;
    BYTE                 byRes[256];
}NET_ITS_ECTWORKSTATE,*LPNET_ITS_ECTWORKSTATE;

```

### Members

*dwSize*

结构体大小

*byDevName*

设备名称

*dwRunTime*

系统运行时间，单位：s

*struTrafficDataInfo*

支持两台主机（数据上传配置的主机）

*dwMemoryUsage*

内存占用率（如果内存占用率为 30%，此处填 30）

*dwCpuUsage*

CPU 占用率（如果 CPU 占用率为 30%，此处填 30）

*dwDevTemperature*

本机温度

*dwDeviceStatic*

设备的状态：0-正常；1-CPU 占用率太高，超过 85%；2-硬件错误，例如串口死掉

*struHardDiskStatic*

硬盘状态

*byAlarmInStatic*

报警输入口的状态：0-没有报警，1-有报警

*byAlarmOutStatic*

报警输出端口的状态：0-没有输出，1-有报警输出

*dwLocalDisplay*

本地显示状态：0-正常，1-不正常

*byRes*

保留，置为 0

## 7.130 NET\_ITS\_GATE\_FACE: 出入口人脸抓拍结果

```

struct{
    DWORD                dwSize;
    BYTE                 byGroupNum;
    BYTE                 byPicNo;
    BYTE                 byFeaturePicNo;
    BYTE                 byRes;
    WORD                 wLaneid;
    BYTE                 byCamLaneId;
    BYTE                 byDir;
    DWORD                dwChanIndex;
    BYTE                 byMonitoringSiteID[MAX_ID_LEN];
    BYTE                 byDeviceID[MAX_ID_LEN];
    NET_VCA_FACESNAP_RESULT struFaceInfo;
    BYTE                 byRes2[256];
}NET_ITS_GATE_FACE, *LPNET_ITS_GATE_FACE;

```

### Members

#### *dwSize*

结构体大小

#### *byGroupNum*

图片组数量（一辆过车多台相机抓拍的图片组的总数，用于多相机数据匹配，目前该参数值为 1）

#### *byPicNo*

连拍的图片序号（接收到图片组数量后，表示接收完成；接收超时不足图片组数量时，根据需要保留或删除）

#### *byFeaturePicNo*

连拍时取第几张图作为特写图，0xff-表示不取

#### *byRes*

保留，置为 0

#### *wLaneid*

车道号，1~32（索引车道号，可以跳跃）

#### *byCamLaneId*

对应相机车道号 1~16（相机配置的车道号，可以跳跃，可以相同）

#### *byDir*

监测方向：0- 其他，1- 入场，2- 出场

#### *dwChanIndex*

通道号

#### *byMonitoringSiteID*

监测点编号

#### *byDeviceID*

设备编号

#### *struFaceInfo*

人脸抓拍数据

*byRes2*

保留，置为 0

## 7.131 NET\_ITS\_GATE\_LANE\_CFG:车道配置

```
struct{
    BYTE    byGateSiteID[MAX_ID_LEN];
    BYTE    byGateInfo[MAX_ID_LEN];
    BYTE    byLaneName[NAME_LEN];
    BYTE    byValid;
    BYTE    byCamLaneId;
    WORD    wLaneid;
    BYTE    byRelativeIoNum;
    BYTE    byDirection;
    BYTE    byLprMode;
    BYTE    byCardMode;
    BYTE    byGateLaneMode;
    BYTE    byCharge;
    BYTE    byChargeMode;
    BYTE    byRes1;
    BYTE    byLedRelativeIndex[MAX_INDEX_LED];
    BYTE    byGateRelativeIndex;
    BYTE    byFarRrRelativeIndex
    BYTE    byRes[82];
}NET_ITS_GATE_LANE_CFG, *LPNET_ITS_GATE_LANE_CFG;
```

### Members

*byGateSiteID*

出入口编号

*byGateInfo*

出入口信息描述

*byLaneName*

车道名称

*byValid*

是否启用：0- 未启用，1- 启用

*byCamLaneId*

对应相机车道号，1~16（相机配置的车道号，可以跳跃，可以相同）

*wLaneid*

车道号，1~32（索引车道号，可以跳跃，逻辑车道）

*byRelativeIoNum*

逻辑车道号关联的输出口编号

*byDirection*

方向编号：0- 其他，1- 入场，2- 出场，3- 双向

*byLprMode*

牌识放行配置：0- 全部进，1- 除禁止名单进，2- 允许名单进

*byCardMode*

卡片放行配置：0- 全部进，1- 除禁止名单进，2- 允许名单进

*byGateLaneMode*

出入口放行配置模式：0- 全部进，1- 只有牌识匹配放行，2- 只有刷卡匹配放行，3- 牌识或卡片有一个匹配则放行，4- 牌识且卡片同时匹配则放行，5- 全不进

*byCharge*

是否收费：0-不收费，1-收费

*byChargeMode*

收费类型：0-车牌收费，1-卡号收费

*byRes1*

保留，置为 0

*byLedRelativeIndex*

LED 索引（每位数组表示一个 LED 设备，最大支持 8 个 LED 设备），取值范围：1~255，紧凑排列，0 表示无效(不解析后面的数据)

*byGateRelativeIndex*

出入口控制机相对索引

*byFarRrRelativeIndex*

远距离读卡器相对索引

*byRes*

保留，置为 0

## 7.132 NET\_ITS\_GATE\_VEHICLE:停车场数据信息

```
struct{
    DWORD                dwSize;
    DWORD                dwMatchNo;
    BYTE                 byGroupNum;
    BYTE                 byPicNo;
    BYTE                 bySecondCam;
    BYTE                 byRes;
    WORD                 wLaneid;
    BYTE                 byCamLaneId;
    BYTE                 byRes1;
    BYTE                 byAlarmReason[MAX_ALARMREASON_LEN];
    WORD                 wBackList;
    WORD                 wSpeedLimit;
    DWORD                dwChanIndex;
    NET_DVR_PLATE_INFO  struPlateInfo;
    NET_DVR_VEHICLE_INFO struVehicleInfo;
    BYTE                 byMonitoringSiteID[MAX_ID_LEN];
    BYTE                 byDeviceID[MAX_ID_LEN];
    BYTE                 byDir;
    BYTE                 byDetectType;
```

```

BYTE                byRes2[2];
BYTE                byCardNo[MAX_ID_LEN];
DWORD              dwPicNum;
NET\_ITS\_PICTURE\_INFO struPicInfo[4];
BYTE                bySwipeTime[MAX_TIME_LEN];
BYTE                byRes3[224];
}NET_ITS_GATE_VEHICLE, *LPNET_ITS_GATE_VEHICLE;

```

### Members

#### *dwSize*

结构体大小

#### *dwMatchNo*

匹配序号，由(车辆序号、数据类型、车道号)组成匹配码，默认：0

#### *byGroupNum*

图片组数量（一辆过车多台相机抓拍的图片组的总数，用于多相机数据匹配，目前该参数值为1）

#### *bySecondCam*

是否第二相机抓拍（如远景抓拍远景相机，或前后抓拍的后相机，特殊项目中会用到，标示前后相机抓拍图片）：0- 否，1- 是

#### *byRes*

保留，置为0

#### *wLaneid*

车道号，1~32（索引车道号，可以跳跃）

#### *byCamLaneid*

对应相机车道号，1~16（相机配置的车道号，可以跳跃，可以相同）

#### *byRes1*

保留，置为0

#### *byAlarmReason*

自定义报警类型，默认为中文

#### *wBackList*

标记为是否报警数据：0- 正常过车数据，1- 禁止名单

#### *wSpeedLimit*

限速上限（超速时有效），单位：km/h

#### *dwChanIndex*

通道号

#### *struPlateInfo*

车牌信息结构

#### *struVehicleInfo*

车辆信息

#### *byMonitoringSiteID*

监测点编号

#### *byDeviceID*

设备编号

#### *byDir*

监测方向：0- 其它，1- 入场，2- 出场

#### *byDetectType*

检测方式：0- 其他，1- 地感触发，2- 视频触发，3- 多帧识别，4- 雷达触发

*byRes2*

保留，置为 0

*byCardNo*

卡号

*dwPicNum*

图片数量（与 *byGroupNum* 不同，代表本条信息附带的图片数量）

*struPicInfo*

图片信息，单张回调，最多 4 张图，由序号区分

*bySwipeTime*

增加刷卡时间，时间格式为 *yyyymmddhh24missfff*

*byRes3*

保留，置为 0

### 7.133 NET\_ITS\_HANDOVER\_INFO: 出入口交接班数据信息

```
struct{
    DWORD    dwSize;
    BYTE     byOperatorName[MAX_OPERATORNAME_LEN];
    BYTE     byOperatorCard[MAX_CARD_LEN];
    BYTE     byStartTime[MAX_TIME_LEN];
    BYTE     byEndTime[MAX_TIME_LEN];
    float    fTotal_Pay;
    DWORD    dwTotal_Records;
    BYTE     byRes[64];
}NET_ITS_HANDOVER_INFO, *LPNET_ITS_HANDOVER_INFO;
```

#### Members

*dwSize*

结构体大小

*byOperatorName*

操作人员名称

*byOperatorCard*

操作人员卡号

*byStartTime*

上班时间，格式为 *yyyymmddhh24miss*

*byEndTime*

下班时间，格式为 *yyyymmddhh24miss*

*fTotal\_Pay*

本次上班期间总的收费金额，单位：元

*dwTotal\_Records*

本次上班期间总的过车收费的记录条数

*byRes*

保留，置为 0

## 7.134 NET\_ITS\_IPC\_CHAN\_LANE\_CFG:出入口参数配置

```
struct{
    DWORD                dwSize;
    BYTE                 byIpcType;
    BYTE                 byRes[135];
    NET\_ITS\_GATE\_LANE\_CFG struGateLane[4];
}NET_ITS_IPC_CHAN_LANE_CFG, *LPNET_ITS_IPC_CHAN_LANE_CFG;
```

### Members

#### *dwSize*

结构体大小

#### *byIpcType*

摄像机类型：0- 无，1- 车牌抓拍，2- 人脸抓拍，3- 监控相机

#### *byRes*

保留，置为 0

#### *struGateLane*

车道配置结构体

## 7.135 NET\_ITS\_PARK\_VEHICLE:停车场数据信息

```
struct{
    DWORD                dwSize;
    BYTE                 byGroupNum;
    BYTE                 byPicNo;
    BYTE                 byLocationNum;
    BYTE                 byParkError;
    BYTE                 byParkingNo[MAX_PARKNO_LEN];
    BYTE                 byLocationStatus;
    BYTE                 bylogicalLaneNum;
    WORD                 wUpLoadType;
    BYTE                 byRes1[4];
    DWORD                dwChanIndex;
    NET\_DVR\_PLATE\_INFO   struPlateInfo;
    NET\_DVR\_VEHICLE\_INFO struVehicleInfo;
    BYTE                 byMonitoringSiteID[MAX_ID_LEN];
    BYTE                 byDeviceID[MAX_ID_LEN];
    DWORD                dwPicNum;
    NET\_ITS\_PICTURE\_INFO struPicInfo[2];
    BYTE                 byRes2[256];
}NET_ITS_PARK_VEHICLE, *LPNET_ITS_PARK_VEHICLE;
```

### Members

#### *dwSize*

结构体大小

**byGroupNum**

图片组数量（一辆过车多台相机抓拍的图片组的总数，用于多相机数据匹配，目前该参数值为 1）

**byPicNo**

连拍的图片序号（接收到图片组数量后，表示接收完成；接收超时不足图片组数量时，根据需要保留或删除）

**byLocationNum**

单张图片所管理的车位数

**byParkError**

停车异常：0- 正常，1- 异常

**byParkingNum**

车位编号

**byLocationStatus**

车位车辆状态：0- 无车，1- 有车

**byLogicalLaneNum**

逻辑车位号，取值范围：0~3（一个相机最大能管 4 个车位，0 代表最左边，3 代表最右边）

**wUploadType**

第零位表示：0- 轮训上传，1- 变化上传

**byRes1**

保留，置为 0

**dwChanIndex**

通道号（IP 通道，起始通道号为 33）

**struPlateInfo**

车牌信息结构

**struVehicleInfo**

车辆信息

**byMonitoringSiteID**

监测点编号

**byDeviceID**

设备编号

**dwPicNum**

图片数量（与 byGroupNum 不同，代表本条信息附带的图片数量）

**struPicInfo**

图片信息，单张回调，最多 2 张图，由序号区分

**byRes2**

保留，置为 0

## 7.136 NET\_ITS\_PASSVEHICLE\_COST\_ITEM: 出入口过车收费明细信息

```
struct{
    DWORD    dwSize;
    DWORD    dwPassVehicleID;
    BYTE     byIntime[MAX_TIME_LEN];
    BYTE     byOuttime[MAX_TIME_LEN];
    BYTE     byCardNo[MAX_CARD_LEN];
}
```

```

BYTE        byPlateInfo[MAX_LICENSE_LEN];
float       fPayCost;
BYTE        byOperatorName[MAX_OPERATORNAME_LEN];
BYTE        byVehicleType;
BYTE        byRes1[3];
DWORD       dwPayRuleID;
DWORD       dwFreeRuleID;
BYTE        byRes2[256];
}NET_ITS_PASSVEHICLE_COST_ITEM, *LPNET_ITS_PASSVEHICLE_COST_ITEM;

```

### Members

#### *dwSize*

结构体大小

#### *dwPassVehicleID*

过车序号与出入口数据的过车序号相关联

#### *byIntime*

入场时间，格式为 `yyyymmddhh24miss`

#### *byOuttime*

出场时间，格式为 `yyyymmddhh24miss`

#### *byCardNo*

卡号

#### *byPlateInfo*

车牌号码

#### *fPayCost*

收费金额(单位：元)

#### *byOperatorName*

操作人员名称

#### *byVehicleType*

车辆类型：0-小型车，2 大型车

#### *byRes1*

保留，置为 0

#### *dwPayRuleID*

收费规则流水号

#### *dwFreeRuleID*

减免规则流水号

#### *byRes2*

保留，置为 0

## 7.137 NET\_ITS\_PICTURE\_INFO:图片信息结构体

```

struct{
    DWORD        dwDataLen;
    BYTE         byType;
    BYTE         byDataType;
    BYTE         byRes1[2];
}

```

```

DWORD        dwRedLightTime;
BYTE         byAbsTime[32];
NET_VCA_RECT struPlateRect;
NET_VCA_RECT struPlateRecgRect;
BYTE        *pBuffer;
BYTE        byRes2[12];
}NET_ITS_PICTURE_INFO, *LPNET_ITS_PICTURE_INFO;

```

### Members

#### *dwDataLen*

媒体数据长度

#### *byType*

数据类型：0-车牌图，1- 场景图，2- 合成图，3- 特写图，4- 二值图，5- 码流

#### *byDataType*

数据上传方式：0-数据直接上传；1-云存储服务器 URL（原先的图片数据变成 URL 数据，图片长度变成 URL 长度）

#### *byRes1*

保留

#### *dwRedLightTime*

经过的红灯时间，单位：s

#### *byAbsTime*

绝对时间点：yyyymmddhhmmssxxx，e.g.20090810235959999，最后三位为毫秒数

#### *struPlateRect*

车牌位置

#### *struPlateRecgRect*

牌识区域坐标。参数中的边界宽 fWidth 和高 fHeight 若为 0，fx 和 fy 不为 0，则(fx,fy)表示牌识的中心点坐标

#### *pBuffer*

保存数据的缓冲区

#### *byRes2*

保留

### Remarks

- 如果设备只上传了场景图，用户可以根据牌识区域坐标（struPlateRecgRect）自己从场景图中截取特写图，宽和高可以根据实际情况自己调节。
- 通过 [NET\\_DVR\\_CLOUDSTORAGE\\_CFG](#) 配置可以启用云存储功能，则上传的图片信息将变成获取图片信息的 URL 地址，平台通过该 URL 地址去云存储服务器上获取数据
- 图片云存储 URL 格式：

```
http://CVMIP:Port/pic?did=DevID&bid=BlkID&pid=PictureID&ptime=PicTime
```

CVMIP: CVM(云存储服务器)的 IP 地址

Port: CVM(云存储服务器)对外提供 http 服务的端口（固定 8009）

DevID: CVS(云存储服务器)中设备 ID 号

BlkID: CVS(云存储服务器)中设备的块号

PictureID: CVS(云存储服务器)为图片生成的编号

PicTime: 图片的时间戳

示例：

http://10.192.65.140:8009/pic?did=35b9cbd0-8ffa-1031-87e6-0025903c6a50&bid=387&pid=2952790009&pti  
me=1378106185

## 7.138 NET\_ITS\_PLATE\_RESULT:识别结果结构体

```

struct{
    DWORD                dwSize;
    DWORD                dwMatchNo;
    BYTE                 byGroupNum;
    BYTE                 byPicNo;
    BYTE                 bySecondCam;
    BYTE                 byFeaturePicNo;
    BYTE                 byDriveChan;
    BYTE                 byVehicleType;
    BYTE                 byRes1[2];
    WORD                 wIllegalType;
    BYTE                 byIllegalSubType[8];
    BYTE                 byPostPicNo;
    BYTE                 byChanIndex;
    WORD                 wSpeedLimit;
    BYTE                 byRes2[2];
    NET\_DVR\_PLATE\_INFO    struPlateInfo;
    NET\_DVR\_VEHICLE\_INFO struVehicleInfo;
    BYTE                 byMonitoringSiteID[48];
    BYTE                 byDeviceID[48];
    BYTE                 byDir;
    BYTE                 byDetectType;
    BYTE                 byRes3[22];
    NET\_DVR\_TIME\_V30    struSnapFirstPicTime;
    DWORD                dwIllegalTime;
    DWORD                dwPicNum;
    NET\_ITS\_PICTURE\_INFO struPicInfo[6];
}NET_ITS_PLATE_RESULT, *LPNET_ITS_PLATE_RESULT;

```

### Members

#### *dwSize*

结构体大小

#### *dwMatchNo*

匹配序号, 由(车辆序号、数据类型、车道号)组成匹配码

#### *byGroupNum*

图片组数量 (一辆过车相机多次抓拍的数量, 代表一组图片的总数, 用于延时匹配数据)

#### *byPicNo*

连拍的图片序号 (接收到图片组数量后, 表示接收完成; 接收超时不足图片组数量时, 根据需要保留或删除)

#### *bySecondCam*

是否第二相机抓拍（如远近景抓拍的远景相机，或前后抓拍的后相机，特殊项目中会用到）

*byFeaturePicNo*

闯红灯电警，取第几张图作为特写图，0xff-表示不取

*byDriveChan*

触发车道号

*byVehicleType*

车辆类型：0- 未知，1- 客车，2- 货车，3- 轿车，4- 面包车，5- 小货车，6- 行人，7- 二轮车，8- 三轮车

*byRes1*

保留

*wlllegalType*

违章类型，采用国标定义，详细定义说明见 [7.161](#) 违章类型国标定义

*bylllegalSubType*

违章子类型

*byPostPicNo*

违章时取第几张图片作为卡口图，0xff-表示不取

*byChanIndex*

通道号（保留）

*wSpeedLimit*

限速上限（超速时有效）

*byRes2*

保留

*struPlateInfo*

车牌信息结构

*struVehicleInfo*

车辆信息

*byMonitoringSiteID*

监测点编号

*byDeviceID*

设备编号

*byDir*

监测方向：1-上行，2-下行，3-双向，4-由东向西，5-由南向北，6-由西向东，7-由北向南，8-其它

*byDetectType*

检测方式：1-地感触发，2-视频触发，3-多帧识别，4-雷达触发

*byRes3*

保留

*dwPicNum*

图片数量（与 *picGroupNum* 不同，代表本条信息附带的图片数量）

*struPicInfo*

图片信息，单张回调，最多 6 张图，由序号区分

## 7.139 NET\_ITS\_REMOTE\_COMMAND:远程控制命令

```
struct{
```

```

WORD    wLaneid;
BYTE    byCamLaneId;
BYTE    byRes;
DWORD   dwCode;
BYTE    byRes1[128];

```

```
}NET_ITS_REMOTE_COMMAND, *LPNET_ITS_REMOTE_COMMAND;
```

### Members

#### *wLaneid*

车道号，1~32（索引车道号，可以跳跃）

#### *byCamLaneId*

对应相机车道号，1~16（相机配置的车道号，可以跳跃，可以相同）

#### *byRes*

保留，置为 0

#### *dwCode*

命令码：0x0- 开闸，0x1- 关闸

#### *byRes1*

保留，置为 0

## 7.140 NET\_ITS\_REMOTE\_CONTROL:远程控制结构体

```

struct{
    DWORD                dwSize;
    NET\_ITS\_REMOTE\_COMMAND struRemoteCommand;
}NET_ITS_REMOTE_CONTROL, *LPNET_ITS_REMOTE_CONTROL;

```

### Members

#### *dwSize*

结构体大小

#### *struRemoteCommand*

远程命令结构

## 7.141 NET\_ITS\_ROADINFO:路口设备异常报警信息

```

struct{
    DWORD                dwSize;
    DWORD                dwChannel;
    BYTE                 bytriggerMode;
    BYTE                 byRes1[3];
    DWORD                dwDeviceNum;
    BYTE                 byMonitoringSiteID[48];
    BYTE                 byRoadInfo[48];
    NET\_ITS\_SINGLE\_DEVICE\_INFO struSingleDevice[ITS_MAX_DEVICE_NUM];
    BYTE                 byRes[16];
}NET_ITS_ROADINFO, *LPNET_ITS_ROADINFO;

```

**Members***dwSize*

结构体大小

*dwChannel*

通道号, 0xffffffff 表示终端, 其它表示对应的相机

*bytriggerMode*

触发模式: 0-线圈触发, 1-视频触发

*byRes1*

保留, 置为 0

*dwDeviceNum*

实际设备个数

*byMonitoringSiteID*

监测点编号

*byRoadInfo*

监测点信息

*struSingleDevice*

路口设备信息

*byRes*

保留, 置为 0

**7.142 NET\_ITS\_SINGLE\_DEVICE\_INFO:路口设备信息**

```

struct{
    DWORD    dwDeviceType;
    DWORD    dwDirID;
    DWORD    dwLaneID;
    DWORD    dwDeviceState;
    BYTE     byDeviceName[32];
    BYTE     byDeviceID[48];
    BYTE     byRes[16];
}NET_ITS_SINGLE_DEVICE_INFO,*LPNET_ITS_SINGLE_DEVICE_INFO;

```

**Members***dwDeviceType*

设备类型: 0-终端, 1-相机, 2-补光灯, 3-车检器, 4-线圈 1, 5-线圈 2, 6-线圈 3, 7-红绿灯检测器, 8-机柜, 9-雷达, 10-道闸, 11-出入口控制机(票箱), 12-LED 显示屏, 13-远距离读卡器, 14-近距离读卡器, 15-红外扫描枪, 16-票据打印机, 17-节点管理器(诱导服务器相关)

*dwDirID*

方向编号: 1-上行, 2-下行, 3-双向, 4-由东向西, 5-由南向北, 6-由西向东, 7-由北向南, 8-其他

*dwLaneID*

车道编号

*dwDeviceState*

设备状态: 0-正常, 非 0-异常 (参考设备异常代码表)

*byDeviceName*

设备名称

**byDeviceID**

设备编号，一般用序列号，车检器用地址

**byRes**

保留

**Remarks**

设备异常代码表如下所示：

设备名称	异常代码	异常说明
机柜	0101	机柜温度异常
	0102	机柜湿度异常
	0103	机柜门状态异常
	0104	机柜异常震动
	0105	其他异常
抓拍单元	0201	抓拍单元温度异常
	0202	抓拍单元湿度异常
	0203	摄像机 DSP 负载率异常
	0204	摄像机内存使用率异常
	0205	摄像机参数配置异常
	0206	摄像机布防异常
	0207	抓拍单元故障
	0208	其他异常
补光灯	0301	补光灯故障
雷达	0401	雷达故障
车辆检测处理器	0501	车检卡异常
	0502	地感线圈异常
	0503	车辆检测处理器异常
	0504	其他异常
交通信号灯检测器	0601	信号灯电信号异常
	0602	交通信号灯检测器异常
终端服务器	0701	终端服务器温度异常
	0702	终端服务器 CPU 使用率异常
	0703	终端服务器内存使用率异常
	0704	终端服务器存储空间异常
	0705	终端服务器软件异常
	0706	其他异常
	0707	终端服务器异常

视频分析记录仪	0801	视频分析记录仪温度异常
	0802	视频分析记录仪 CPU 使用率异常
	0803	视频分析记录仪内存使用率异常
	0804	视频分析记录仪存储空间异常
	0805	视频分析记录仪软件异常
	0806	其他异常
	0807	视频分析记录仪异常
牌识处理单元	0901	视频输入异常
	0902	牌识处理单元异常
出入口控制终端	1001	出入口控制终端温度异常
	1002	出入口控制终端 CPU 使用率异常
	1003	出入口控制终端内存使用率异常
	1004	出入口控制终端存储空间异常
	1005	出入口控制终端软件异常
	1006	其他异常
	1007	出入口控制终端异常
LED 诱导屏	1101	LED 诱导屏异常
串口控制器	1201	串口控制器异常

### 7.143 NET\_ITS\_TRAFFIC\_COLLECT: 交通统计数据

```

struct{
    DWORD                dwSize;
    BYTE                 byMonitoringSiteID[48];
    BYTE                 byDeviceID[48];
    BYTE                 byLaneNum;
    BYTE                 byDir;
    BYTE                 byDetectType;
    BYTE                 byRes1;
    DWORD                dwChannel;
    NET_DVR_SYSTEM_TIME struStartTime;
    DWORD                dwSamplePeriod;
    NET_ITS_TRAFFIC_DRIVE_CHAN struDriveChan[6];
    BYTE                 byRes2[24];
}NET_ITS_TRAFFIC_COLLECT,*LPNET_ITS_TRAFFIC_COLLECT;

```

#### Members

*dwSize*

结构体大小

*byMonitoringSiteID*

监测点编号

*byDeviceID*

设备编号

*byLaneNum*

车道总数

*byDir*

监测方向：1-上行，2-下行，3-双向，4-由东向西，5-由南向北，6-由西向东，7-由北向南，8-其它

*byDetectType*

检测方式：1-地感线圈，2-视频触发，3-多帧识别，4-雷达触发

*byRes1*

保留

*dwChannel*

通道号

*struStartTime*

统计开始时间

*dwSamplePeriod*

统计时间，单位：秒

*struDriveChan*

交通流量数据，每个相机支持 6 个车道

*byRes*

保留

## 7.144 NET\_ITS\_TRAFFIC\_DATA\_HOST:接收交通数据主机信息

```
struct{
    NET_DVR_IPADDR    struHostAddr;
    WORD              wHostPort;
    BYTE              byRes1[2];
    DWORD             dwDataType;
    BYTE              bySuspendUpload;
    BYTE              byUploadStrategy;
    WORD              wUploadInterval;
    DWORD             dwUploadTimeOut;
    BYTE              byRes[24];
}NET_ITS_TRAFFIC_DATA_HOST,*LPNET_ITS_TRAFFIC_DATA_HOST;
```

### Members

*struHostAddr*

远程主机 IP 地址

*wHostPort*

远程主机端口号

*byRes1*

保留，置为 0

*dwDataType*

上传远程主机的数据类型，按位表示，0 代表全部不上传，0xffffffff 代表全部上传，不同的类型需上传则对应的数值取或，定义如下：

违章代码	违章描述	数值
0	普通卡口数据	0x01
1018	机动车不在机动车道内行驶的(车道禁行)	0x01 << 1
1019	机动车违反规定使用专用车道的(预留)	0x01 << 2
1208	不按导向标志行驶（违章左转、违章直行、违章右转等）	0x01 << 3
1211	通过路口遇停止信号时，停在停止线以内或路口内的(压停止线)	0x01 << 4
1229	违反禁令标志(预留，目前没有提供此类违章代码)	0x01 << 5
1230	机动车禁止标线指示（压车道线等）	0x01 << 6
1301	逆行	0x01 << 7
1302	违反信号灯（闯红灯）	0x01 << 8
1042	机动车不按规定车道行驶	0x01 << 9
1303	机动车行驶超过规定时速 50%以下	0x01 << 10
1603	机动车行驶超过规定时速 50%	0x01 << 11

*bySuspendUpload*

是否暂停数据上传：0-正常上传，1-暂停上传

*byUploadStrategy*

上传策略：0-最新数据优先上传，1-按照通行时间上传

*wUploadInterval*

上传间隔时间，单位：ms，取值范围：1~2000，该参数只对历史数据有效

*dwUploadTimeOut*

上传超时时间

*byRes*

保留，置为 0

## 7.145 NET\_ITS\_TRAFFIC\_DATA\_INFO:交通数据信息

```
struct{
    DWORD                dwDataQuantity;
    DWORD                dwDataRsendQuantity;
    NET_DVR_SYSTEM_TIME struStartTime;
    NET_DVR_SYSTEM_TIME struEndTime;
    NET_DVR_IPADDR       struDataHost;
}NET_ITS_TRAFFIC_DATA_INFO,*LPNET_ITS_TRAFFIC_DATA_INFO;
```

### Members

*dwDataQuantity*

数据总量

*dwDataRsendQuantity*

未发送数量（需要上传交通数据主机的）

*struStartTime*

未发送数据的最早时间

*struEndTime*

未发送数据的最晚时间

*struDataHost*

接收交通数据主机地址

## 7.146 NET\_ITS\_TRAFFIC\_DRIVE\_CHAN:单个车道路况信息

```
struct{
    BYTE    byDriveChan;
    BYTE    byRes1[3];
    WORD    wCarFlux;
    WORD    wPasserbyFlux;
    WORD    wShayFlux;
    float   fAverOccpancy;
    WORD    wAverSpeed;
    WORD    wAverCarDis;
    BYTE    byRes2[16];
}NET_ITS_TRAFFIC_DRIVE_CHAN,*LPNET_ITS_TRAFFIC_DRIVE_CHAN;
```

### Members

*byDriveChan*

车道号

*byRes1*

保留

*wCarFlux*

汽车流量数

*wPasserbyFlux*

行人流量数

*wShayFlux*

二轮车流量数

*fAverOccpancy*

平均车道占有率百分比，单位：%，例如 12.6 表示占有率为 12.6%

*wAverSpeed*

平均车速(km/h)

*wAverCarDis*

平均车头时距，单位：毫秒

*byRes2*

保留

## 7.147 NET\_ITS\_UPLOAD\_CFG:数据上传配置参数

struct{

```

DWORD                                dwSize;
NET\_ITS\_TRAFFIC\_DATA\_HOST          struRemoteDataHost1;
NET\_ITS\_TRAFFIC\_DATA\_HOST          struRemoteDataHost2;
}NET_ITS_UPLOAD_CFG,*LPNET_ITS_UPLOAD_CFG;

```

### Members

*dwSize*

结构体大小

*struRemoteDataHost1*

接收交通数据主机 1

*struRemoteDataHost2*

接收交通数据主机 2

## 7.148 **NET\_ITS\_WORKSTATE**:终端工作状态信息

```

struct{
  DWORD                                dwSize;
  BYTE                                  byDevName[32];
  DWORD                                dwRunTime;
  NET\_ITS\_TRAFFIC\_DATA\_INFO          struTrafficDataInfo[2];
  DWORD                                dwMemoryUsage;
  DWORD                                dwCpuUsage;
  DWORD                                dwDevTemperature;
  DWORD                                dwDeviceStatic;
  NET\_DVR\_DISKSTATE                struHardDiskStatic[MAX_DISKNUM_V30];
  NET\_DVR\_CHANNELSTATE\_V30         struChanStatic[MAX_CHANNUM_V30];
  BYTE                                  byAlarmInStatic[MAX_ALARMIN_V30];
  BYTE                                  byAlarmOutStatic[MAX_ALARMOUT_V30];
  DWORD                                dwLocalDisplay;
  BYTE                                  byAudioInChanStatus[8];
  BYTE                                  byRes[4];
}NET_ITS_WORKSTATE,*LPNET_ITS_WORKSTATE;

```

### Members

*dwSize*

结构体大小

*byDevName*

设备名称

*dwRunTime*

系统运行时间，单位：s

*struTrafficDataInfo*

支持两台主机（数据上传配置的主机）

*dwMemoryUsage*

内存占用率（如果内存占用率为 30%，此处填 30）

*dwCpuUsage*

CPU 占用率（如果 CPU 占用率为 30%，此处填 30）

*dwDevTemperature*

本机温度

*dwDeviceStatic*

设备的状态：0-正常；1-CPU 占用率太高，超过 85%；2-硬件错误，例如串口死掉

*struHardDiskStatic*

硬盘状态

*struChanStatic*

通道的状态

*byAlarmInStatic*

报警输入口的状态：0-没有报警，1-有报警

*byAlarmOutStatic*

报警输出端口的状态：0-没有输出，1-有报警输出

*dwLocalDisplay*

本地显示状态：0-正常，1-不正常

*byAudioInChanStatus*

表示语音通道的状态：0-未使用；1-使用中，数组 0 表示第 1 个语音通道

*byRes*

保留，置为 0

## 7.149 NET\_VCA\_DEV\_INFO:前端设备信息

```
struct{
    NET_DVR_IPADDR    struDevIP;
    WORD              wPort;
    BYTE              byChannel;
    BYTE              byIvmsChannel;
}NET_VCA_DEV_INFO,*LPNET_VCA_DEV_INFO;
```

### Members

*struDevIP*

前端设备地址

*wPort*

前端设备端口号

*byChannel*

前端设备通道号，参数值表示通道号。比如，byChannel=1，表示通道 1。

*byIvmsChannel*

IVMS 通道号

## 7.150 NET\_VCA\_FACESNAP\_RESULT:人脸抓拍结果

```
struct{
    DWORD              dwSize;
    DWORD              dwRelativeTime;
    DWORD              dwAbsTime;
```

```

DWORD          dwFacePicID;
DWORD          dwFaceScore;
NET\_VCA\_TARGET\_INFO  struTargetInfo;
NET\_VCA\_RECT      struRect;
NET\_VCA\_DEV\_INFO  struDevInfo;
DWORD          dwFacePicLen;
DWORD          dwBackgroundPicLen;
BYTE           byRes[60];
BYTE           *pBuffer1;
BYTE           *pBuffer2;
}NET_VCA_FACESNAP_RESULT, *LPNET_VCA_FACESNAP_RESULT;

```

### Members

*dwSize*

结构体大小

*dwRelativeTime*

相对时标

*dwAbsTime*

绝对时标

*dwFacePicID*

人脸图 ID

*dwFaceScore*

人脸评分，范围：0~100

*struTargetInfo*

报警目标信息

*struRect*

人脸子图区域

*struDevInfo*

前端设备信息

*dwFacePicLen*

人脸子图的长度，为 0 表示没有图片，大于 0 表示有图片

*dwBackgroundPicLen*

背景图的长度，为 0 表示没有图片，大于 0 表示有图片(保留)

*byRes*

保留

*\*pBuffer1*

人脸子图的图片数据

*\*pBuffer2*

背景图的图片数据（保留，通过查找背景图接口可以获取背景图）

## 7.151 NET\_VCA\_POINT:点坐标参数

```

struct{
    float  fX;
    float  fY;
}

```

```
}NET_VCA_POINT,*LPNET_VCA_POINT;
```

### Members

*fX*

X 轴坐标, 取值范围[0.001,1]

*fY*

Y 轴坐标, 取值范围[0.001,1]

## 7.152 NET\_VCA\_RECT:区域框参数

```
struct{
    float    fX;
    float    fY;
    float    fWidth;
    float    fHeight;
}NET_VCA_RECT,*LPNET_VCA_RECT;
```

### Members

*fX*

边界框左上角点的 X 轴坐标, 取值范围[0.001,1]

*fY*

边界框左上角点的 Y 轴坐标, 取值范围[0.001,1]

*fWidth*

边界框的宽度, 取值范围[0.001,1]

*fHeight*

边界框的高度, 取值范围[0.001,1]

## 7.153 NET\_VCA\_SIZE\_FILTER:尺寸过滤器参数

```
struct{
    BYTE        byActive;
    BYTE        byMode;
    BYTE        byRes[2];
    NET\_VCA\_RECT    struMiniRect;
    NET\_VCA\_RECT    struMaxRect;
}NET_VCA_SIZE_FILTER,*LPNET_VCA_SIZE_FILTER;
```

### Members

*byActive*

是否激活尺寸过滤器, 0-否, 非 0-是

*byMode*

过滤器模式, 具体定义如下:

```
enum_VCA_SIZE_FILTER_MODE_{
    IMAGE_PIX_MODE,
    REAL_WORLD_MODE,
    DEFAULT_MODE
```

```
}SIZE_FILTER_MODE
```

```
IMAGE_PIX_MODE
```

根据像素大小设置

```
REAL_WORLD_MODE
```

根据实际大小设置

```
DEFAULT_MODE
```

默认模式(目前 ATM 支持)

*byRes*

保留，置为 0

*struMiniRect*

最小目标框，全 0 表示不设置。实际模式时取值范围为 0-50，单位为 m。

*struMaxRect*

最大目标框，全 0 表示不设置

## 7.154 NET\_VCA\_TARGET\_INFO:报警目标信息

```
struct{
    DWORD          dwID;
    NET_VCA_RECT   struRect;
    BYTE           byRes[4];
}NET_VCA_TARGET_INFO,*LPNET_VCA_TARGET_INFO;
```

### Members

*dwID*

目标 ID，人员密度过高报警时为 0

*struRect*

目标边界框

*byRes*

保留，置为 0

## 7.155 NET\_VCA\_POLYGON:多边形

```
struct{
    DWORD          dwPointNum;
    NET_VCA_POINT  struPos[VCA_MAX_POLYGON_POINT_NUM];
}NET_VCA_POLYGON,*LPNET_VCA_POLYGON;
```

### Members

*dwPointNum*

有效点（大于等于 3），若是 3 点在一条线上认为是无效区域，线交叉认为是无效区域

*struPos*

多边形边界点，最大值为 10

## 7.156 BasicCapability:设备软硬件能力集

```

<?xml version="1.0" encoding="utf-8" ?>
<!-- req, 设备软硬件能力集描述 -->
<BasicCapability version="2.0">
  <!-- req, 硬件支持能力 -->
  <HardwareCapability>
    <!-- req, 前面板版本,高 16 位是主版本,低 16 位是次版本 -->
    <HardwareVersion>0x0</HardwareVersion>
    <!-- req, 报警输入个数 -->
    <AlarmInPortNum>4</AlarmInPortNum>
    <!-- req, 报警输出个数 -->
    <AlarmOutPortNum>4</AlarmOutPortNum>
    <!-- req, 232 串口个数 -->
    <RS232Num>1</RS232Num>
    <!-- req, 485 串口个数 -->
    <RS485Num>1</RS485Num>
    <!-- req, 网络口个数 -->
    <NetworkPortNum>1</NetworkPortNum>
    <!-- req, USB 口的个数 -->
    <USBNum>1</USBNum>
    <!-- req, Flash 大小 (单位 M) -->
    <FlashSize />
    <!-- req, Ram 大小 (单位 M) -->
    <RamSize />
    <!-- req, USB 版本 -->
    <USBVersion />
    <!-- req, SD 卡个数 -->
    <SDNum>0</SDNum>
    <!-- req, 硬盘个数(包括 SATA、eSATA 和 NAS), 实际已接可用的硬盘个数 -->
    <HardDiskNum>1</HardDiskNum>
    <!-- req, 2012-5-14SATA 盘个数 -->
    <SATANum>1</SATANum>
    <!-- req, 2012-5-14eSATA 盘个数 -->
    <eSATANum>1</eSATANum>
    <!-- req, 2012-5-14miniSAS 盘个数 -->
    <miniSASNum>1</miniSASNum>
    <!-- req, VIN 个数 -->
    <VideoInNum>4</VideoInNum>
    <!-- req, AIN 个数 -->
    <AudioInNum>4</AudioInNum>
    <!-- req, VOUT 个数 -->
    <VideoOutNum>1</VideoOutNum>
  </HardwareCapability>
</BasicCapability>

```

```

<!-- req, AOUT 个数 -->
<AudioOutNum>1</AudioOutNum>
<!-- req, 语音对讲通道个数 -->
<AudioTalkNum>2</AudioTalkNum>
<!-- req, 支持 SD 卡, 1-支持, 不支持时不显示 -->
<SDSupport>1</SDSupport>
<!-- req, 支持 WiFi, 1-支持, 不支持时不显示 -->
<WiFiSupport>1</WiFiSupport>
<!-- req, 支持 POE(IPC 的网线供电), 1-支持, 不支持时不显示 -->
<POESupport>1</POESupport>
<!-- req, 支持红外, 1-支持, 不支持时不显示 -->
<IRSupport>1</IRSupport>
<!-- req, 支持本地输出, 1-支持, 不支持时不显示 -->
<VideoOutSupport>1</VideoOutSupport>
<!-- req, 支持恢复出厂设置, 1-支持, 不支持时不显示 -->
<ResetSupport>1</ResetSupport>
<!-- req, 模拟通道个数,和注册返回的 byChanNum 的值一致 -->
<AnalogChannelNum>255</AnalogChannelNum>
<!-- req, 数字通道个数 -->
<IPChannelNum>255</IPChannelNum>
<!-- req, 支持多网卡, 1-支持, 不支持时不显示 -->
<MultiNetworkCard>1</MultiNetworkCard>
<!-- req, 支持 bonding 功能, 1-支持, 不支持时无此项 -->
<BondingSupport>1</BondingSupport>
<!-- req, VGA 个数 -->
<VGANumber>255</VGANumber>
<!-- req, HDMI 个数 -->
<HDMINumber>255</HDMINumber>
<!-- req, SDI 个数 -->
<SDINumber>255</SDINumber>
<!-- req, CVBS 个数 -->
<CVBSNumber>255</CVBSNumber>
<!-- req, 辅助输出个数 -->
<AuxoutNumber>255</AuxoutNumber>
<!-- req, RAID 类型, 0-硬 RAID, 1-软 RAID -->
<RAIDType>1</RAIDType>
<RS485>
  <localRs485No min="" max="" /> <!-- req, 本地 485 口序号-->
  <expandRs485No min="" max="" /> <!-- req, 可扩展 485 口序号-->
  <fullDuplexRs485No min="" max="" /> <!-- req, 全双工 485 口序号, 相对于本地 485 口数-->
  <semiDuplexRs485No min="" max="" /> <!-- req, 半双工 485 口序号, 相对于本地 485 口数-->
  <RS485SlotNo min="" max="" /> <!-- req, 485 槽位序号-->
</RS485>
</HardwareCapability>

```

```

<!-- req, 软件支持能力 -->
<SoftwareCapability>
  <!-- req, 硬盘是否支持新的硬盘号映射方式:1-支持, 不支持时不显示 -->
  <!-- req, 在新的映射方式下, IPC 给 SDK 传的硬盘号: SD 卡是 0, NAS 盘从 8 开始; 之前的硬盘号表示为 0、1、2.....-->
  <NewHdNo>1</NewHdNo>
  <!-- req, 2012-5-14 支持网盘的最大个数(不支持时不显示),网盘包括 Nas 和 IPSAN -->
  <MaxNetworkHDNum>8</MaxNetworkHDNum>
  <!-- req, 支持 nas 存储功能, 1-支持, 不支持时不显示。兼容 IPC 不能删除 -->
  <NasSupport>1</NasSupport>
  <!-- req, nas 个数 -->
  <NasNumber>8</NasNumber>
  <!-- req, 支持的结构体 NET_DVR_NET_DISKCFG -->
  <NetDiskIdentification>
  <!-- req, 表示支持网盘接入支持认证 -->
  <NASIdentification>
  <!-- req, 表示支持 NAS 接入支持认证 -->
  <NFSMountType>true</NFSMountType>
  <!-- req, 表示支持 NAS 支持 NFS 方式接入 -->
  <CIFSMountType>
  <!-- req, 表示支持 NAS 支持 CIFS 方式接入 -->
  <usernameLen min="" max="" />
  <!-- req, NAS 认证上的 name 长度的最大值和最小值-->
  <passwordLen min="" max="" />
  <!-- req, NAS 认证上的 passwd 长度的最大值和最小值-->
  </CIFSMountType>
  </NASIdentification>
  </NetDiskIdentification>
  <!-- req, 最大字符串叠加个数 (针对模拟通道) -->
  <ShowStringNumber>8</ShowStringNumber>
  <!-- req, 支持移动侦测, 1-支持, 不支持时不显示 -->
  <MotionDetectAlarmSupport>1</MotionDetectAlarmSupport>
  <!-- req, 支持视频丢失报警, 1-支持, 不支持时不显示 (针对模拟通道) -->
  <VILostAlarmSupport>1</VILostAlarmSupport>
  <!-- req, 支持遮挡报警, 1-支持, 不支持时不显示 (针对模拟通道) -->
  <HideAlarmSupport>1</HideAlarmSupport>
  <!-- req, 支持遮盖, 1-支持, 不支持时不显示 (针对模拟通道) -->
  <ShelterSupport>1</ShelterSupport>
  <!-- req, 支持 rtsp 网络传输功能, 1-支持, 不支持时不显示 -->
  <RtspSupport>1</RtspSupport>
  <!-- req, 支持 rtp over rtsp 网络传输功能, 1-支持, 不支持时不显示 -->
  <RtpoverRtspSupport>1</RtpoverRtspSupport>
  <!-- req, 支持 rtsp over http 网络传输功能, 1-支持, 不支持时不显示 -->
  <RtspoverHttpSupport>1</RtspoverHttpSupport>

```

```

<!-- req, 支持 NTP 校时 , 1-支持, 不支持时不显示 -->
<NtpSupport>1</NtpSupport>
<!-- req, 支持电子云台, 1-支持, 不支持时不显示 -->
<EptzSupport>1</EptzSupport>
<!-- req, 支持云台控制, 1-支持, 不支持时不显示 -->
<PtzSupport>1</PtzSupport>
<!-- req, 支持 DDNS, 1-支持, 不支持时不显示。兼容 IPC 不能删除 -->
<DDNSSupport>1</DDNSSupport>
<!-- req, DDNS 服务器类型: 0- IP Server, 1-Dyndns, 2-PeanutHull(花生壳), 3-NO-IP, 4- hkDDNS -->
<DDNSHostType>0,1,2,3,4</DDNSHostType>
<SNMPSupport>1</SNMPSupport>
<!-- req, SNMP 版本, 1-v1, 2-v2, 3-v3 -->
<SNMPVersion>1,2,3</SNMPVersion>
<!-- req, 支持 UPNP, 1-支持, 不支持时不显示 -->
<UPNPSupport>1</UPNPSupport>
<!-- req, 支持 iSCSI, 1-支持, 不支持时不显示 -->
<iSCSISupport>1</iSCSISupport>
<!-- req, 2012-5-14 支持 IP SAN 个数 -->
<iSCSINum>1</iSCSINum>
<!-- req, 支持 ipv6, 1-支持, 不支持时不显示 -->
<Ipv6Support>0</Ipv6Support>
<!-- req, 支持智能, 1-支持, 不支持时不显示 -->
<VCASupport>1</VCASupport>
<!-- req, 支持复合流, 1-支持, 不支持时不显示 (针对模拟通道) -->
<MultipleStreamSupport>1</MultipleStreamSupport>
<!-- req, 支持子码流, 1-支持, 不支持时不显示 (针对模拟通道) -->
<SubStreamSupport>1</SubStreamSupport>
<!-- req, 支持 EMAIL, 1-支持, 不支持时不显示 -->
<EmailSupport>1</EmailSupport>
<!-- req, SADP 版本, 0-V1.0, 1-V3.0 -->
<SADPVersion>0,1</SADPVersion>
<ZeroChanNumber>1</ZeroChanNumber>
<!-- req, 支持备份, 1-支持, 不支持时不显示 -->
<BackupSupport>1</BackupSupport>
<!-- req, 支持按事件查找录像, 1-支持, 不支持时无此项 -->
<FindFileByEventSupport>1</FindFileByEventSupport>
<!-- req, 支持智能搜索录像, 1-支持, 不支持时不显示 -->
<SmartSearchSupport>1</SmartSearchSupport>
<!-- req, 支持 ATM 配置, 1-支持, 不支持时不显示 -->
<ATMSupport>1</ATMSupport>
<!-- req, 支持文件锁定/解锁, 1-支持, 不支持时不显示 -->
<FileLockSupport>1</FileLockSupport>
<!-- req, 支持主子码流码率动态限制, 1-支持, 不支持时不显示(此能力无对应配置) -->
<BitrateLimitSupport>1</BitrateLimitSupport>

```

```

<!-- req, 支持注册最大路数 -->
<MaxLoginNum>0</MaxLoginNum>
<!-- req, 支持预览最大路数 -->
<MaxPreviewNum>0</MaxPreviewNum>
<!-- req, 支持回放最大路数 -->
<MaxPlayBackNum>0</MaxPlayBackNum>
<!-- req, 通道支持最大连接路数 -->
<MaxChanLinkNum>6</MaxChanLinkNum>
<ShutDownSupport>1</ShutDownSupport>
<!-- req, 支持预置点冻结, 1-支持, 不支持时不显示 -->
<FrameFreezeSupport>1</FrameFreezeSupport>
<!-- req, 2012-03-22 是否支持硬盘盘组配置, 1-支持, 不支持时不显示 -->
<HDgroupConfig>1</HDgroupConfig>
<!-- req, 2012-03-22 是否支持 232 配置, 1-支持, 不支持时不显示 -->
<RS232Config>1</RS232Config>
<!-- req, 2012-03-22 是否支持 PPPoE 配置, 1-支持, 不支持时不显示 -->
<PPPoEConfig>1</PPPoEConfig>
<!-- req, 2012-03-22 是否支持冗余录像, 1-支持, 不支持时不显示 -->
<RedundancyRecord>1</RedundancyRecord>
<!-- req, 2012-03-22 是否支持主辅口缩放配置, 1-支持, 不支持时不显示 -->
<VideoOutScaleConfig>1</VideoOutScaleConfig>
<!-- req, 2012-03-22 是否支持 guest 用户, 1-支持, 不支持时不显示 -->
<GuestUser>1</GuestUser>
<!-- req, 2012-03-22 是否支持 FTP 上传图片, 1-支持, 不支持时不显示 -->
<UploadFTP>1</UploadFTP>
<!-- req, 2012-05-15 是否不支持磁盘配额, 录像和图片配额都支持时不显示, 1-录像和图片配额都不支持, 2-支持录像配额但不支持图片配额 -->
<NotSupportDiskQuota>1</NotSupportDiskQuota>
<!-- req, 2012-12-8 支持磁盘配额按比例分配, 1-支持, 不支持时不显示-->
<QuotaRatio>1</QuotaRatio>
<LocalVout>
  <mainVoutType opt="mainCVBS,HDMI,VGA,auto" />
  <!-- req,本地主口输入类型, mainCVBS-主 CVB. 非同源设备: 0-Auto 1-主 CVBS 2-HDMI 3-VGA 同源设备: 0-Auto
  1-主 CVBS 2-HDMI/VGASnetra2.2 以后支持选择,之前版本为 auto(能力集解析返回 auto)-->
</LocalVout>
<SingleStorageMaxCap>16</SingleStorageMaxCap> <!-- req,系统支持单个存储设备最大容量, 单位:T-->
<IPCPlug>1</IPCPlug> <!-- req 是否支持 IPC 即插即用, 1-支持, 不支持不显示-->
<DrawFrameRecord>1</DrawFrameRecord> <!-- req 是否支持抽帧录像, 1-支持, 不支持不显示-->
<DellInvalidDisk>1</DellInvalidDisk> <!-- req 是否支持删除无效磁盘, 1-支持, 不支持不显示-->
<MountOrUnmountDisk>1</MountOrUnmountDisk> <!-- req,是否支持加载或卸载磁盘, 1-支持, 不支持不显示-->
<MaxDvcsSubDevNumNum><!-- req,分布式设备支持的最大设备数, 不支持不显示--> </MaxDvcsSubDevNumNum>
<NotSupportInputOutputConfigFile>1</NotSupportInputOutputConfigFile>
<!-- req,不支持导入导出参数配置, 1-不支持-->
<NotSupportLogSearch>1</NotSupportLogSearch> <!-- req,不支持日志查询, 支持时不显示, 1-不支持-->

```

```

<DateUpLoadAndDownLoad>
  <!-- 数据上传下载功能-->
  <audioType opt="wave" /> <!-- 支持的语音类型-->
  <uploadAndDownLoad opt="upload,download" /> <!-- 支持上传下载-->
  <maxNum opt="8" /> <!-- type:支持的最大语音个数-->
</DateUpLoadAndDownLoad>
<!-- req,支持 IP 可视对讲主机能力集,对应 IpViewDevAbility 能力集-->
<IpViewDev>1</IpViewDev>
<DevModuleServerCfg>
  <!-- req,设备服务配置能力-->
  <!-- req,只允许 admin 用户设置 telnet 配置-->
  <telnetServer opt="disable,enable" />
  <!-- req, 0-disable(禁用), 1-enable(启用)-->
  <irLampServer opt="disable,enable" />
  <!-- req, 0-disable(禁用), 1-enable(启用)-->
  <abfServer opt="enable,disable," />
  <!-- req, 0-enable(启用), 1-disable(禁用) -->
</DevModuleServerCfg>
<GBT28181AccessAbilitySupport>1</GBT28181AccessAbilitySupport>
<!-- req,是否支持 GB/T28181 协议, 1-支持, 不支持不显示-->
<SearchLogAbilitySupport>1</SearchLogAbilitySupport>
<!-- req,是否支持 GB/T28181 协议, 1-支持, 不支持不显示-->
<AlarmTriggerRecordAbilitySupport>1</AlarmTriggerRecordAbilitySupport>
<!-- req,是否支持日志搜索能力, 1-支持, 不支持不显示-->
<CameraParaDynamicAbilitySupport>1</CameraParaDynamicAbilitySupport>
<!-- req,是否支持前端参数动态能力, 1-支持, 不支持不显示-->
<IOAbilitySupport>1</IOAbilitySupport>
<!-- req, IO 输入输出能力, 1-支持, 不支持不显示-->
<AccessProtocolAbility>1</AccessProtocolAbility>
<!-- req, 协议接入能力, 1-支持, 不支持不显示-->
<CameraMountAbility>1</CameraMountAbility>
<!-- req, 摄像机架设能力, 1-支持, 不支持不显示-->
<VehicleRecogAbility>1</VehicleRecogAbility>
<!-- req, 车辆二次检测能力, 1-支持, 不支持不显示-->
<VcaChanAbility>1</VcaChanAbility>
<!-- req, VCA 智能通道能力, 1-支持, 不支持不显示 -->
<Language>
  <!-- req, 支持的语言种类-->
  <supportType opt="0-noSupport,1-chinese,2-english" />
  <!-- req, 0 表示不支持这个功能区分, 1 表示中文, 2 表示英文-->
</Language>
<TransDevice>
  <transChannelNum min="" max="" /> <!-- req, 转码通道最大最小值-->
</TransDevice>

```

```

<MultiNetworkCardMode>0,1</MultiNetworkCardMode>
<!-- req,支持的多网卡工作模式,0-普通多网卡模式, 1-内外网隔离模式, 不支持模式设置不显示-->
<DeviceWorkMode>1,2,3</DeviceWorkMode>
<!-- req, 设备支持的工作模式, 1-通道模式, 2-流 ID 模式, 3-GB28281 模式, 不支持模式设置不显示-->
<AllBackupLog>
    <enabled>true</enabled> <!-- req, 是否支持全部日志导出-->
</AllBackupLog>
<VoiceTalkAsAudioIn>
    <enabled>true</enabled> <!-- req, 是否支持语音对讲作为音频输入-->
    <supportVoiceChan opt="1,2" /> <!-- opt, 支持的语音对讲通道号-->
</VoiceTalkAsAudioIn>
<NeedReboot>
    <!-- req, 导入配置文件是否自动重启, 1-重启, 2-需要提示重启, 不需要提示重启和设备不会自动重启时不显示 -->
    <ImportConfigurationFileReboot>1</ImportConfigurationFileReboot>
    <!-- req, ESATA 用途修改是否自动重启, 1-重启, 不重启时不显示 -->
    <EsataUseageChange>1</EsataUseageChange>
    <!-- req, 修改报警输入类型是否自动重启, 1-重启, 2-需要提示重启, 不需要提示重启和设备不会自动重启时不显示 -->
    <AlarmInTypeChange>1</AlarmInTypeChange>
    <!-- req, 模拟通道是否启用修改是否自动重启, 1-重启, 不重启时不显示 -->
    <AnalogChanEnableChange>1</AnalogChanEnableChange>
    <!-- req, 恢复默认参数是否自动重启, 1-重启, 2-需要提示重启, 不需要提示重启和设备不会自动重启时不显示 -->
    <RestoreConfig>1</RestoreConfig>
    <!-- req, 串口 232 传输模式更改是否需要重启, 1-重启, 不重启时不显示 -->
    <RS232workModeChange>1</RS232workModeChange>
    <!-- req, 网络传输端口更改是否需要重启, 1-重启, 不重启时不显示 -->
    <NetPortChange>1</NetPortChange>
    <!-- req, RTSP 端口更改是否需要重启, 1-重启, 不重启时不显示 -->
    <RtspPortChange>1</RtspPortChange>
    <!-- req, DHCP 启用状态更改是否需要重启, 1-重启, 不重启时不显示 -->
    <DhcpEnableChange>1</DhcpEnableChange>
    <!-- req, HTTP 端口更改是否需要重启, 1-重启, 不重启时不显示 -->
    <HttpPortChange>1</HttpPortChange>
    <!-- req, PPPoE 参数更改是否需要重启, 1-重启, 不重启时不显示 -->
    <PPPoEChange>1</PPPoEChange>
    <!-- req, 网络参数中的多播地址更改是否需要重启, 1-重启, 不重启时不显示 -->
    <NetMultiCastIPChange>1</NetMultiCastIPChange>
    <!-- req, 硬盘参数更改是否需要重启, 1-重启, 不重启时不显示 -->
    <HardDiskParamChange>1</HardDiskParamChange>
    <!-- req, 录像计划的时间更改是否需要重启, 1-重启, 不重启时不显示 -->
    <RecordTimeChange>1</RecordTimeChange>
    <!-- req, 压缩参数的视频编码类型更改是否需要重启, 1-重启, 不重启时不显示 -->

```

```

<VideoEncodeTypeChange>1</VideoEncodeTypeChange>
<!-- req, 压缩参数的音频编码类型更改是否需要重启, 1-重启, 不重启时不显示 -->
<AudioEncodeTypeChange>1</AudioEncodeTypeChange>
<!-- req, 制式更改是否需要重启, 1-重启, 不重启时不显示 -->
<StandardTypeChange>1</StandardTypeChange>
<!-- req, 去雾使能状态更改是否需要重启, 1-重启, 不需要重启不显示 -->
<DehazeEnableChange>1</DehazeEnableChange>
<!-- req, 行编码使能状态更改是否需要重启, 1-重启, 不需要重启不显示 -->
<LineCodingEnableChange>1</LineCodingEnableChange>
<!-- req, IPC 前端参数本地输出开关状态更改是否需要重启, 1-重启, 不需要重启不显示 -->
<LocalOutputEnableChange>1</LocalOutputEnableChange>
<!-- req, 切换指定主口是否需要重启, 1-重启, 不需要重启不显示 -->
<LocalMainVoutTypeChange>1</LocalMainVoutTypeChange>
<!-- req, NAS 认证参数修改是否需要重启, 1-重启, 不需要重启不显示 -->
<NASIdentificationChange>1</NASIdentificationChange>
<!-- req, 工作模式切换是否自动重启, 1-重启, 不重启不显示 -->
<DevWorkModeChange>1</DevWorkModeChange>
<!-- req, SIP 本地端口更改是否需要重启, 1-重启, 不重启时不显示 -->
<LocalPortChange>1</LocalPortChange>
<!-- req, SIP 服务器注册周期更改是否需要重启, 1-重启, 不重启时不显示 -->
<LoginCycleChange>1</LoginCycleChange>
<!-- req, RTP 端口更改是否需要重启, 1-重启, 不重启时不显示 -->
<RtpPortChange>1</RtpPortChange>
<!-- req, 音频编码优先级更改是否需要重启, 1-重启, 不重启时不显示 -->
<AudioEncodePriorityChange>1</AudioEncodePriorityChange>
<!-- req, 延迟预览时间修改是否需要重启, 1-重启, 不重启时不显示 -->
<PreviewDelayTimeChange>1</PreviewDelayTimeChange>
</NeedReboot>
</SoftwareCapability>
</BasicCapability>

```

## 7.157 ITDeviceAbility:智能终端能力集

### 输入描述:

```

<?xml version="1.0" encoding="UTF-8"?>
<!--req, 获取智能终端能力时 pInBuf 参数描述-->
<ITDeviceAbility version="2.0">
  <channelNO> 1
    <!--req, xs:inter,通道号-->
  </channelNO>
  <!--opt,可以指定节点返回-->
  <ITSAbility/>
  <!--opt,指定返回节点-->

```

```
</ITDeviceAbility>
```

### 能力集描述:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--req, 智能交通摄像机和智能终端能力集描述 -->
<ITDeviceAbility version="2.0">
  <channelNO>1</channelNO>
  <!--req, xs:inter,通道号-->
  <ITSAbility>
    <cameraAmount>
      <!--req, xs:inter, 最大支持的相机个数-->
    </cameraAmount>
    <storeAmount>
      <!--req, xs:inter, 最大支持的数据存储总量（单位：个）-->
    </storeAmount>
    <ImageCombine>
      <enabled>true</enabled>
      <!--req, 是否支持图片合成,不支持不显示-->
    </ImageCombine>
    <TrafficDataUpload>
      <UploadType1>
        <enabled> true</enabled>
        <!--req, 支持上传模式 1(即抓拍机支持的类型,布防方式的 COMM_UPLOAD_PLATE_RESULT),不支持不显示-->
      </UploadType1>
      <UploadType2>
        <enabled> true</enabled>
        <!--req, 支持上传模式 2(即终端支持的类型,监听方式的 COMM_ITS_PLATE_RESULT),不支持不显示-->
      </UploadType2>
      <UploadType3>
        <enabled> true</enabled>
        <!--req, 支持上传模式 2(即抓拍机支持布防上传(COMM_ITS_PLATE_RESULT/ NET_ITS_PLATE_RESULT),不支持不显示-->
      </UploadType3>
    </TrafficDataUpload>
    <VideoLocalRecord>
      <enabled> true<!--req, 是否支持本地录像,不支持不显示--></enabled>
    </VideoLocalRecord>
    <PicLocalStore>
      <enabled> true<!--req, 是否支持图片本地存储,不支持不显示--></enabled>
    </PicLocalStore>
    <PlateRecognise>
      <enabled> true<!--req, 是否支持车牌识别,不支持不显示--></enabled>
      <regionType opt="rect,polygon"/>
      <!-- 牌识区域支持的类型-->
      <plateTypeopt opt= "standard92, standard02, armedPolice, police, standard92Back, embassy, agricultural,
```

```

moto"/>
    <plateColor opt="blue,yellow,white,black,green"/>
    <vehicleType opt="small,big"/>
    <bodyworkColor opt="white,silver,gray,black,red,darkblue,blue,yellow,green,brown,pink,violet"/>
</PlateRecognise>
<imageOverlayString>
    <mode1> <!--opt, 原抓拍机的图片字符叠加,不支持不显示-->
        <overlayStringType opt="monitor,time,speed,speedRatio,speedLimit,plate,carColor,carLength,
carType,laneNum,milliSecond,illegalInfo,redOnTime"/>
        <!--opt, 监测点,时间,速度,超速比,限速标志,车牌,车身颜色,车身长度,车辆类型,车道号,毫秒,违规信息,红灯
已亮时间-->
    </mode1>
    <mode2>
        <!--opt, 终端和终端相机的图片字符叠加,不支持不显示-->
        <overlapType opt="monitor,site,roadNum,instrumentNum,directionNum,directionDes, laneNum,
laneDes,capTime,capTimeMilli,plate,carColor,carLength,carType,carBrand,speed,speedRatio,
speedLimit,illegalDes,redStart,redStop,redOnTime,securityCode,capCode"/>
        <!--opt, 监测点,地点,路口编号,设备编号,方向编号,方向描述,车道号,车道描述,抓拍时间,抓拍时间(毫秒),车
牌号,车身颜色,车身长度,车辆类型,车辆品牌,速度,超速比,限速标志,违规信息,红灯开始时间,红灯结束时间,红灯已亮时间,防伪
码,抓拍编号-->
        <itemLength max="" min="0"/>
        <!--opt, 字符叠加长度-->
        <changeLineNum max="10" min="0"/>
        <!--opt, 换行数-->
        <spaceNum max="255" min="0"/>
        <!--opt, 空格数-->
        <linePercent max="100" min="0"/>
        <!--opt, 叠加行百分比-->
        <itemsStlye opt="horizontal,vertical"/>
        <!--opt, 叠加方式, 横排, 竖排-->
        <startPosTop max="2448" min="0"/>
        <!--opt, 起始上坐标-->
        <startPosLeft max="2448" min="0"/>
        <!--opt, 起始左坐标-->
        <charStyle opt="SongTi,WeiTi"/>
        <!--opt, 字体, 宋体, 魏体-->
        <charSize opt="16*16,32*32,48*48,64*64"/>
        <!--req,字符大小 0-32*32(中)/16*32(英), 1-64*64(中)/32*64(英), 2-48*48 -->
        <charInterval max="16" min="0"/>
        <!--opt, 字符间距-->
        <ForeClorRGB>
            <enabled> true</enabled>
            <!--opt, 是否支持调节前景色 RGB,不支持不显示-->
        </ForeClorRGB>

```

```

    <BackClorRGB>
      <enabled> true</enabled>
      <!--opt, 是否支持调节背景色 RGB,不支持不显示-->
    </BackClorRGB>
    <ColorSelfAdapt>
      <enabled> true</enabled>
      <!--opt, 颜色是否自适应,不支持不显示-->
    </ColorSelfAdapt>
  </mode2>
</imageOverlayString>
<TPSAlarm>
  <!--req, 支持交通统计报警上传,不支持不显示-->
  <UploadType1>
    <enabled> true</enabled>
    <!--req, 支持上传模式 2(即终端支持的类型,监听方式的 COMM_ITS_TRAFFIC_COLLECT), 不支持不显示-->
  </UploadType1>
  <UploadType2>
    <enabled> true</enabled>
    <!--req, 支持上传模式 2 (即抓拍机支持布防上传(COMM_ITS_TRAFFIC_COLLECT/NET_ITS_TRAFFIC_COLLECT),
不支持不显示-->
  </UploadType2>
</TPSAlarm>
<CameraSetup>
  <!--req, 支持相机架设配置,不支持不显示-->
  <enabled> true</enabled>
</CameraSetup>
<supportMilliCheckTime>
  <!-- req, 支持毫秒校时-->
  <enabled>true</enabled>
</supportMilliCheckTime>
<CloudStorage>
  <poolID min="" max="" />
  <userNameLen min="" max="" />
  <passwdLen min="" max="" />
  <poolInfo opt="postVideoPool,illegalVideoPool" />
  <!-- req 数组 0 表示卡口录像池, 数组 1 表示违章录像池-->
  <supportURLUpload opt="true,false" />
  <!-- req 支持 URL 上传方式-->
</CloudStorage>

<!--req,停车场项目-->
<LampCtrlInfo>
  <lampCtrlMode opt="inlayLamp,externalLamp"/>
  <!--req, 控灯模式: 1 代表内置灯, 2 代表外接灯-->

```

```

<ctrlChannelIndex min="" max=""/>
<!--req 交替控制通道号-->

<inlayLampCtrlMode>
  <!--req 内置灯控制模式结构 -->
  <lampStateCtrlNum min="" max=""/>
  <!--req 停车位支持数范围-->
  <parkInlayInfo>
    <enable opt="false,true"/>
    <!--req,是否启用, true-启用, false-不启用-->
    <flicker opt="false,true"/>
    <!--req,是否闪烁, true-闪烁, false-不闪烁-->
    <lampColor opt="close,red,green,yellow,blue,magenta,cyan,white"/>
    <!--req 0- 灭 1-红 2-绿 3-黄 4-蓝 5-品红 6-青 7-白色-->
  </parkInlayInfo>
</inlayLampCtrlMode>

<externalLampCtrlMode>
  <!--req 接灯控制模式结构 -->
  <maxParkNum min="" max=""/>
  <!--req 停车位数量-->
  <parkInfoType opt="normalParkIOstate,normalNoParkIOstate,specialParkIOstate, noSpecialParkIOstate"/>
  <parkExternalSubinfo>
    <enable opt="false,true"/>
    <!--req,是否启用, true-启用, false-不启用-->
    <flicker opt="false,true"/>
    <!--req,是否闪烁, true-闪烁, false-不闪烁-->
    <IOstate opt="lowLevel,highLevel"/>
    <!--req,高低电平, 0-低电平, 1-高电平(外置灯起效)-->
  </parkExternalSubinfo>
</externalLampCtrlMode>
</LampCtrlInfo>

<parkSpaceAttributeParam>
  <maxParkNum min="" max=""/>
  <!--req 停车位数量-->
  <parkSpaceInfo>
    <parkSpaceAttribute opt="normalPack,specialPack"/>
    <!-- 0~普通车位 1~特殊车位-->
  </parkSpaceInfo>
</parkSpaceAttributeParam>

<lampExternalCfg>

```

```

    <enable opt="false,true"/>
    <!--req,是否启用, true-启用, false-不启用-->
    <lampState>
        <flicker opt="false,true"/>
        <!--req,是否闪烁, true-闪烁, false-不闪烁-->
        <IONo opt="IO1,IO2,IO3"/>
        <!--req 1~IO1,2~IO2,4~IO3 -->
    </lampState>
</lampExternalCfg>

<compelCaptureCfg>
    <parkIndex opt="parkPlace1,parkPlace2,parkPlace3,parkPlace4"/>
    <!-- 车位序号从左到右 1,2,3,4-->
</compelCaptureCfg>

<externalControlAlarm>
    <lampState>
        <flicker opt="false,true"/>
        <!--req,是否闪烁, true-闪烁, false-不闪烁-->
        <IONo opt="IO1,IO2,IO3"/>
        <!--req 1~IO1,2~IO2,4~IO3 -->
    </lampState>
    <externalBeginTime>true</externalBeginTime>
    <!--req 是否支持 外控开始时间的上传-->
</externalControlAlarm>
</ITSAbility>
</ITDeviceAbility>

```

## 7.158 VcaChanAbility:智能通道分析能力集

### 输入描述:

```

<?xml version="1.0" encoding="utf-8" ?>
<!-- req, 获取智能通道分析能力时 pinBuf 参数描述 -->
<VcaChanAbility version="2.0">
    <channelNO>1</channelNO><!-- req, 通道号 -->
</VcaChanAbility>

```

### 能力集描述:

```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- req, 智能通道分析能力集描述 -->
<VcaChanAbility>
    <channelNO>1</channelNO> <!-- 通道号 -->
    <restartLib>true</restartLib> <!-- 支持重启算法库 -->
    <vcaVersion>true</vcaVersion> <!-- 支持查看算法库版本 -->

```

```

<syncChannelName>true</syncChannelName> <!-- 同步分析仪的通道名为前端设备通道名 -->
<!-- 智能信息叠加 -->
<VcaDrawMode>
  <dspEncAddTarget opt="true,false" /> <!-- 编码叠加目标 -->
  <dspEncAddRule opt="true,false" /> <!-- 编码叠加规则 -->
  <dspPicAddTarget opt="true,false" /> <!-- 抓图叠加目标 -->
  <dspPicAddRule opt="true,false" /> <!-- 抓图叠加规则 -->
</VcaDrawMode>
<!-- 报警图片分辨率，index 为 SDK 定义的图片大小索引值，参见 NET_DVR_JPEGPARA -->
<AlarmPicResolutionList>
  <picResolutionEntry index="3" name="UXGA" resolution="1600*1200" />
  <picResolutionEntry index="4" name="SVGA" resolution="800*600" />
  <picResolutionEntry index="5" name="HD720P" resolution="1280*720" />
  <picResolutionEntry index="6" name="VGA" resolution="640*480" />
</AlarmPicResolutionList>
<!-- 全局尺寸过滤 -->
<GlobalSizeFilter>
  <mode opt="imagePixel,realWorld,default" default="imagePixel" />
  <!-- 过滤模式，imagePixel-像素大小，realWorld-实际大小，default-默认 -->
  <minRect>true</minRect> <!-- 最小目标框 -->
  <maxRect>true</maxRect> <!-- 最大目标框 -->
</GlobalSizeFilter>
<!-- 行为分析 -->
<Behavior>
  <EventType>
    <!-- 穿越警戒面 -->
    <TraversePlane>
      <planeBottom>true</planeBottom> <!-- 警戒面底边 -->
      <crossDirection opt="bothDirection,leftToRight,rightToLeft" default="bothDirection" />
      <!-- 穿越方向，bothDirection-双向，leftToRight-从左到右，rightToLeft-从右到左 -->
      <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->
      <planeHeight min="0" max="255" default="5" /> <!-- 警戒面高度 -->
    </TraversePlane>
    <!-- 进入区域 -->
    <EnterArea>
      <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
      </Region>
    </EnterArea>
    <!-- 离开区域 -->
    <ExitArea>
      <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
      </Region>
    </ExitArea>
  </EventType>
</Behavior>

```

```

</ExitArea>
<!-- 入侵参数 -->
<Intrusion>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <duration min="0" max="100" default="5" /> <!-- 行为事件触发时间阈值 -->
  <sensitivity min="1" max="100" default="50" /> <!-- 灵敏度 -->
  <rate min="1" max="100" default="5" /> <!-- 占比 -->
</Intrusion>
<!-- 徘徊 -->
<Loiter>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <duration min="1" max="120" default="10" /> <!-- 触发时间阈值 -->
</Loiter>
<!-- 物品放置拿取 -->
<LeftTake>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <duration min="1" max="120" default="10" /> <!-- 触发时间阈值 -->
</LeftTake>
<!-- 停车 -->
<Parking>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <duration min="1" max="100" default="10" /> <!-- 触发时间阈值 -->
</Parking>
<!-- 奔跑 -->
<Run>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <distance min="0.1" max="1.0" default="0.5" /> <!-- 人奔跑最大距离 -->
  <mode opt="imagePixel,realWorld" default="imagePixel" />
  <!-- imagePixel-像素模式, realWorld-实际模式 -->
</Run>
<!-- 人员聚集 -->
<HighDensity>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->

```

```

    </Region>
    <density min="0.1" max="1.0" default="0.5" /> <!-- 密度比率 -->
    <duration min="20" max="360" default="20" /> <!-- 触发人员聚集参数报警阈值 -->
</HighDensity>
<!-- 剧烈运动 -->
<ViolentMotion>
    <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
    <duration min="1" max="120" default="50" /> <!-- 触发事件阈值 -->
    <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->
    <mode opt="video,audio,videoAndAudio" default="video" />
    <!-- video-纯视频模式，audio-纯音频模式，videoAndAudio-音视频联合模式 -->
</ViolentMotion>
<!-- 攀高 -->
<ReachHeight>
    <planeBottom>true</planeBottom> <!-- 攀高警戒面 -->
    <duration min="1" max="120" default="10" /> <!-- 触发攀高报警阈值 -->
</ReachHeight>
<!-- 起身 -->
<GetUp>
    <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
    <duration min="1" max="100" default="10" /> <!-- 触发起床报警阈值 -->
    <sensitivity min="1" max="10" default="10" /> <!-- 灵敏度 -->
    <mode opt="overBed,areaMove,sitting" default="overBed" />
    <!-- overBed-大床通铺模式，areaMove-高低铺模式，sitting-大床通铺坐立起身模式 -->
</GetUp>
<!-- 物品放置 -->
<Left>
    <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
    <duration min="10" max="100" default="10" /> <!-- 触发物品遗留报警阈值 -->
    <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->
</Left>
<!-- 物品拿取 -->
<Take>
    <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
    <duration min="10" max="100" default="10" /> <!-- 触发物品拿取报警阈值 -->
    <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->

```

```

</Take>
<!-- 离岗 -->
<LeavePosition>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <leaveDelay min="1" max="1800" default="120" /> <!-- 无人报警时间 -->
  <staticDealy min="1" max="1800" default="120" /> <!-- 睡觉报警时间 -->
  <mode opt="leave,sleep,leaveAndSleep" default="leave" />
  <!-- leave-离岗事件, sleep-睡岗事件, leaveAndSleep-离睡岗事件 -->
  <personType opt="single,couple" default="single" />
  <!-- 值岗人数类型, single-单人值岗, couple-双人值岗 -->
</LeavePosition>
<!-- 尾随 -->
<Trail>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->
</Trail>
<!-- 重点人员起身 -->
<KeyPersonGetUp>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <duration min="1" max="100" default="20" /> <!-- 触发起床报警阈值 -->
  <sensitivity min="1" max="10" default="5" /> <!-- 灵敏度 -->
  <mode opt="overBed,areaMove,sitting" default="overBed" />
  <!-- overBed-大床通铺模式, areaMove-高低铺模式, sitting-大床通铺坐立起身模式 -->
</KeyPersonGetUp>
<!-- 倒地 -->
<FallDown>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <duration min="1" max="60" default="30" /> <!-- 触发事件阈值 -->
  <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->
</FallDown>
<!-- 声强突变 -->
<AudioAbnormal>
  <decibel min="1" max="255" default="50" /> <!-- 声音强度 -->
  <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->
  <audioMode opt="sensitivity,decibel,sensitivityAndDecibel" default="sensitivity" />
  <!-- sensitivity-灵敏度检测, decibel-分贝阈值检测, sensitivityAndDecibel-灵敏度与分贝阈值检测 -->

```

```

    <enabled>true</enabled>
    <!-- 使能标志 -->
    <threshold min="0" max="100" default="80" />
    <!-- 声音阈值 -->
</AudioAbnormal>
<!-- 折线攀高 -->
<ADVReachHeight>
    <Region>
        <vertexNum min="2" max="10" />
        <!-- 区域顶点数量 -->
    </Region>
    <crossDirection opt="bothDirection,leftToRight,rightToLeft" default="bothDirection" />
    <!-- 穿越方向, bothDirection-双向, leftToRight-从左到右,rightToLeft-从右到左 -->
</ADVReachHeight>
<!-- 如厕超时 -->
<ToiletTarry>
    <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
    <duration min="1" max="3600" default="600" /><!-- 如厕超时时间 -->
</ToiletTarry>
<!-- 放风场滞留 -->
<YardTarry>
    <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
    <duration min="1" max="120" default="60" /> <!-- 放风场滞留时间 -->
</YardTarry>
<!-- 折线警戒面 -->
<ADVTraversePlane>
    <Region>
        <vertexNum min="2" max="10" /> <!-- 区域顶点数量 -->
    </Region>
    <crossDirection opt="bothDirection,leftToRight,rightToLeft" default="bothDirection" />
    <!-- 穿越方向, bothDirection-双向, leftToRight-从左到右,rightToLeft-从右到左 -->
    <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->
</ADVTraversePlane>
<!-- 操作超时 -->
<OverTime>
    <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
    <duration min="4" max="60000" default="30000" /> <!-- 操作报警时间阈值 -->
</OverTime>

```

```

<!-- 贴纸条 -->
<StickUp>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <duration min="4" max="60" default="30" /> <!-- 触发时间阈值 -->
  <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->
</StickUp>
<!-- 安装读卡器 -->
<InstallScanner>
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
  <duration min="4" max="60" default="30" /> <!-- 读卡持续时间 -->
  <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->
</InstallScanner>
</EventType>
<!-- 行为规则 -->
<BehaviorRule>
  <picProcType opt="notProcess,upload" /> <!-- 图片处理方式,notProcess-不处理, upload-上传 -->
  <uploadLastAlarm opt="false,true" /> <!-- 上传最近一次的报警 -->
  <picRecordEnable opt="false,true" /> <!-- 图片存储 -->
  <JpegParam>
    <picSize>true</picSize> <!-- 图片大小 -->
    <picQuality opt="best,better,normal" /> <!-- 图片质量,best-最好, better-较好, normal-一般 -->
  </JpegParam>
  <maxRelSnapChanNum>3</maxRelSnapChanNum> <!-- 最大关联抓图通道数 -->
  <RuleEntryList>
    <maxRuleNum>8</maxRuleNum> <!-- 最大规则数目 -->
    <RuleEntry>
      <eventType opt="traversePlane,enterArea,exitArea,intrusion,loiter,leftAndTake,parking,run,
highDensity,violentMotion,reachHeight,getup,left,take,leavePosition,trail,
keyPersonGetup,fallDown,audioAbnormal,advReachHeight,toiletTarry,yardTarry,
advTraversePlane,humanEnter,overTime,stickup,installScanner" />
      <!-- 事件类型,traversePlane-穿越警戒面,enterArea-进入区域,exitArea-离开区域,intrusion-周界入侵,
loiter-徘徊,leftAndTake-物品放置拿取,parking-停车,run-奔跑,highDensity-区域内人员密
度,violentMotion-剧烈运动,reachHeight-攀高,getup-起身,left-物品放置,take-物品拿取,leavePosition-
离岗,trail-尾随,keyPersonGetup-重点人员起身,fallDown-倒地,audioAbnormal-声强突
变,advReachHeight-折线攀高,toiletTarry-如厕超时,yardTarry-放风场滞留,advTraversePlane-折线警戒
面,humanEnter-人靠近 ATM,overTime-操作超时,stickup-贴纸条,installScanner-安装读卡器 -->
      <ruleNameLength min="0" max="32" /> <!-- 规则名称长度 -->
      <SizeFilter>
        <mode opt="imagePixel,realWorld,default" default="imagePixel" />
        <!-- 过滤模式, imagePixel-像素大小, realWorld-实际大小, default-默认 -->
      </SizeFilter>
    </RuleEntry>
  </RuleEntryList>
</BehaviorRule>

```

```

        <minRect>true</minRect> <!-- 最小目标框 -->
        <maxRect>true</maxRect> <!-- 最大目标框 -->
    </SizeFilter>
    <AlarmTime>
        <timeSegNum min="2" max="8" /> <!-- 每天布防时间段数量 -->
    </AlarmTime>
    <alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp" />
    <!-- monitor-监视器上警告,audio-声音警告,center-上传中心,alarmout-触发报警输出,picture-jpeg 抓
    图并上传 email,wirelesslight-无线声光报警, uploadftp-抓图并上传 ftp -->
    <relRecordChan>true</relRecordChan> <!-- 报警触发通道录像 -->
    <alarmDelay opt="5,10,30,60,120,300,600" /> <!-- 报警延时 -->
    <FilterStrategy>
        <strategy opt="disabled,widthAndHeight,targetArea" default="disabled" />
        <!-- 过滤策略, disabled-不启用, widthAndHeight-高度和宽度过滤, targetArea-目标面积过滤
        -->
    </FilterStrategy>
    <TriggerParam>
        <tiggerMode opt="disabled,trackPoint,targetArea" default="disabled" />
        <!-- 触发方式, disabled-不启用, trackPoint-目标点, targetArea-目标面积 -->
        <triggerPoint opt="center,up,down" default="center" />
        <!-- 触发点,center-中, up-上, down-下 -->
        <triggerArea min="0" max="100" default="50" /> <!-- 触发目标面积百分比 -->
    </TriggerParam>
</RuleEntry>
</RuleEntryList>
</BehaviorRule>
<!-- 屏蔽区域 -->
<MaskRegion>
    <maxRegionNum>4</maxRegionNum> <!-- 区域个数 -->
    <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
</MaskRegion>
<!-- 进入区域 -->
<EnterRegion>
    <maxRegionNum>1</maxRegionNum> <!-- 区域个数 -->
    <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
</EnterRegion>
<!-- 标定 -->
<Calibration>
    <BehaviorIn>
        <!-- 行为室内场景标定 -->

```

```

    <enabled opt="true,false" /> <!-- 启用性 -->
    <calSampleNum min="2" max="5" /> <!-- 标定样本个数 -->
    <CalSample>
      <!-- 标定样本 -->
      <targetRect>true</targetRect> <!-- 目标框 -->
      <LineSegment>
        <lineMode opt="heightLine,lengthLine" />
        <!-- heightLine-高度样本线,lengthLine-长度样本线 -->
        <value min="1" max="1000" /> <!-- 高度或长度 -->
      </LineSegment>
    </CalSample>
    <CameraParam>
      <cameraHeight min="2" max="50" /> <!-- 摄像机高度 -->
      <cameraPitchAngle min="1" max="89" /> <!-- 摄像机俯仰角度 -->
      <horizonLine min="0.0" max="1.0" /> <!-- 场景中的地平线 -->
    </CameraParam>
  </BehaviorIn>
  <BehaviorOut>
    <!-- 行为室外场景标定 -->
    <enabled opt="true,false" /> <!-- 启用性 -->
    <lineSegNum min="4" max="8" /> <!-- 样本线个数 -->
    <LineSegment>
      <lineMode opt="heightLine,lengthLine" /> <!-- heightLine-高度样本线,lengthLine-长度样本线 -->
      <value min="1" max="1000" /> <!-- 高度或长度 -->
    </LineSegment>
    <CameraParam>
      <cameraHeight min="2" max="50" /> <!-- 摄像机高度 -->
      <cameraPitchAngle min="1" max="89" /> <!-- 摄像机俯仰角度 -->
      <horizonLine min="0.0" max="1.0" /> <!-- 场景中的地平线 -->
    </CameraParam>
  </BehaviorOut>
  <calibVerify>true</calibVerify> <!-- 支持标定校验 -->
</Calibration>
<Scene>
  <maxSceneNum>1</maxSceneNum> <!-- 最大场景数量 -->
</Scene>
</Behavior>
<!-- 交通 -->
<Traffic>
  <!-- 场景参数 -->
  <Scene>
    <maxSceneNum>16</maxSceneNum> <!-- 最大场景数量 -->
    <maxSceneTimeSegNum>16</maxSceneTimeSegNum> <!-- 最大场景时间段数量 -->
  </SceneParam>

```

```

    <direction>true</direction> <!-- 检测方向 -->
    <sceneID>true</sceneID> <!-- 场景 ID -->
    <sceneNameLength min="0" max="32" /> <!-- 场景名称长度 -->
    <ptzPos>true</ptzPos> <!-- ptz 坐标 -->
    <trackTime min="5" max="300" /> <!-- 球机跟随时间 -->
  </SceneParam>
</Scene>
<!-- 屏蔽区域 -->
<MaskRegion>
  <maxRegionNum>4</maxRegionNum> <!-- 区域个数 -->
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
</MaskRegion>
<!-- 参考区域 -->
<ReferenceRegion>
  <maxRegionNum>4</maxRegionNum> <!-- 区域个数 -->
  <Region>
    <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
  </Region>
</ReferenceRegion>
<!-- 标定 -->
<Calibration>
  <enabled opt="true,false" /> <!-- 启用性 -->
  <calibPointNum min="4" max="4" /> <!-- 标定点个数 -->
  <width>true</width> <!-- 宽度 -->
  <height>true</height> <!-- 高度 -->
  <calibVerify>true</calibVerify> <!-- 支持标定校验 -->
</Calibration>
<!-- 车道配置 -->
<LaneCfg>
  <maxLaneNum>8</maxLaneNum> <!-- 最大车道数量 -->
  <LaneParam>
    <laneNameLength min="0" max="32" /> <!-- 车道名称长度 -->
    <flowDirection>true</flowDirection> <!-- 车道内车流方向 -->
    <Region>
      <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
  </LaneParam>
</LaneCfg>
<!-- 交通事件规则 -->
<AidRule>
  <picProcType opt="notProcess,upload" /> <!-- 图片处理方式,notProcess-不处理, upload-上传 -->
  <JpegParam>

```

```

    <picSize>true</picSize> <!-- 图片大小 -->
    <picQuality opt="best,better,normal" /> <!-- 图片质量,best-最好, better-较好, normal-一般 -->
</JpegParam>
<RuleEntryList>
    <maxRuleNum>8</maxRuleNum> <!-- 最大规则数量 -->
    <RuleEntry>
        <ruleNameLength min="0" max="32" /> <!-- 规则名称长度 -->
        <eventType opt="congestion,parking,inverse,pedestrian,debris,smoke,overLine,vehicleControlList,
speed, illegalLaneChange, turnAround"/> <!-- 事件类型, congestion-拥堵, parking-停车, inverse-逆
行, pedestrian-行人, debris-遗留物, smoke-烟雾, overLine-压线, vehicleControlList-禁止名单, speed-
超速, illegalLaneChange-变道, turnAround-掉头 -->
        <SizeFilter>
            <mode opt="imagePixel,realWorld,default" default="imagePixel" />
            <!-- 过滤模式, imagePixel-像素大小, realWorld-实际大小, default-默认 -->
            <minRect>true</minRect> <!-- 最小目标框 -->
            <maxRect>true</maxRect> <!-- 最大目标框 -->
        </SizeFilter>
        <Region>
            <vertexNum min="3" max="10" /> <!-- 区域顶点数量-->
        </Region>
        <AidParam>
            <parkingDuration min="10" max="120" default="100" /> <!-- 违停检测灵敏度 -->
            <pedestrianDuration min="1" max="120" default="100" /> <!-- 行人持续时间 -->
            <debrisDuration min="10" max="120" default="100" /> <!-- 抛洒物持续时间 -->
            <congestionLength min="5" max="200" default="100" /> <!-- 拥堵长度阈值 -->
            <congestionDuration min="10" max="120" default="60" /> <!-- 拥堵持续时间 -->
            <inverseDuration min="1" max="10" default="5" /> <!-- 逆行持续时间 -->
            <inverseDistance min="2" max="100" default="50" /> <!-- 逆行距离阈值 -->
            <inverseAngleTolerance min="90" max="180" default="100" /> <!-- 允许角度偏差 -->
            <illegalParkingTime min="4" max="60" default="10" /> <!-- 违停时间 -->
            <illegalParkingPicNum min="1" max="6" default="4" /> <!-- 违停图片数量 -->
            <mergePic>true</mergePic> <!-- 支持图片拼接 -->
        </AidParam>
        <AlarmTime>
            <timeSegNum min="2" max="8" /> <!-- 每天布防时间段数量 -->
        </AlarmTime>
        <alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp" />
        <!-- monitor-监视器上警告,audio-声音警告,center-上传中心,alarmout-触发报警输出,picture-jpeg 抓
图并上传 email,wirelesslight-无线声光报警, uploadftp-抓图并上传 ftp -->
        <relRecordChan>true</relRecordChan> <!-- 报警触发通道录像 -->
        <laneNo min="1" max="99" /> <!-- 车道号 -->
    </RuleEntry>
</RuleEntryList>
</AidRule>

```

```

<!-- 交通数据统计规则 -->
<TpsRule>
  <RuleEntryList>
    <maxRuleNum>8</maxRuleNum> <!-- 最大规则数目 -->
    <RuleEntry>
      <laneID>true</laneID> <!-- 车道 ID -->
      <calcType opt="laneVolume,laneVelocity,timeHeadway,spaceHeadway,timeOccupancyRatio,
spaceOccupancyRatio,queue,vehicleType,trafficState" />
      <!-- 统计参数类型, laneVolume-车道流量, laneVelocity-车道速度, timeHeadway-车头时距,
spaceHeadway-车头间距, timeOccupancyRatio-车道占有率(时间上), spaceOccupancyRatio-车道占
占有率(空间上), queue-排队长度, vehicleType-车辆类型, trafficState-交通状态 -->
      <SizeFilter>
        <mode opt="imagePixel,realWorld,default" default="imagePixel" />
        <!-- 过滤模式, imagePixel-像素大小, realWorld-实际大小, default-默认 -->
        <minRect>true</minRect> <!-- 最小目标框 -->
        <maxRect>true</maxRect> <!-- 最大目标框 -->
      </SizeFilter>
      <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
      </Region>
      <AlarmTime>
        <timeSegNum min="2" max="8" /> <!-- 每天布防时间段数量 -->
      </AlarmTime>
      <alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp" />
      <!-- monitor-监视器上警告,audio-声音警告,center-上传中心,alarmout-触发报警输出,picture-jpeg 抓
图并上传 email,wirelesslight-无线声光报警, uploadftp-抓图并上传 ftp -->
    </RuleEntry>
  </RuleEntryList>
</TpsRule>
<forensicsMode opt="manual,auto" /> <!-- 取证方式, manual-手动取证, auto-自动取证 -->
<trafficSceneMode opt="freeway,tunnel,bridge" />
<!-- 交通场景模式, freeway-高速户外场景, tunnel-高速隧道场景, bridge-高速桥梁场景 -->

<!-- 视频触发参数 -->
<ITCTriggerCfg>
  <triggerMode opt="VIAVirtualCoil" />
  <!-- 触发模式, VIAVirtualCoil-VIA (Vehicle characteristics Intelligent Analysis) 虚拟线圈触发 -->

  <!-- VIA 虚拟线圈触发参数 -->
  <VIAVirtualCoilParam>
    <laneNum>true</laneNum> <!-- 车道数量 -->
    <laneBoundaryLine>true</laneBoundaryLine> <!-- 车道边界线,即最左边车道的左边界线 -->
    <VIALane>
      <maxLaneNum>6</maxLaneNum> <!-- 最大车道数量 -->
    </VIALane>
  </VIAVirtualCoilParam>
</ITCTriggerCfg>

```

```

<associatedLaneNO>true</associatedLaneNO> <!-- 关联车道号 -->
<LaneLogicParam>
  <laneUseage opt="unknown,carriageWay,bus,fast,slow,motor,nonMotor,reverse,banTrucks,mix" />
  <!-- 车道用途, unknown-未知,carriageWay-普通车道,bus-公交专用车道,fast-快车道,slow-慢车道,motor-摩托车道,nonMotor-非机动车道,reverse-反向车道,banTruck-禁止货车车道,mix-混合车道 -->
  <laneDirection opt="unknown,left,straight,leftStraight,right,leftRight,rightStraight,leftRightStraight, leftWait, straightWait,forward,backward,bothway" />
  <!-- 车道方向,unknown-未知, left-左转, straight-直行,leftStraight-左转+直行,right-右转,leftRight-左转+右转,rightStraight-右转+直行,leftRightStraight-左转+右转+直行,leftWait-左转待行,straightWait-直行待行,forward-正向行驶,backward-背向行驶,bothway-双向行驶 -->
  <carDriveDirection opt="unknown,uptodown,downtoup"/>
  <!-- 车辆行驶方向,unknown-未知,uptodown-下行,downtoup-上行 -->
</LaneLogicParam>
<laneLine>true</laneLine> <!-- 车道线 -->
<Region>
  <vertexNum min="3" max="20" /> <!-- 牌识区域, 区域顶点数量 -->
</Region>
</VIALane>
<PlateRecogParam>
  <provinceAbbreviation>true</provinceAbbreviation><!-- 设备运行省份的汉字简写 -->
  <!-- 识别的类型 -->
  <RecogMode>
    <plateRecogPos opt="fromFront,fromBack"/>
    <!-- 车牌识别位置, fromFront-正向识别,fromBack-背向识别 -->
    <plateRecogType opt="big,little" /> <!-- 车牌识别类型, big-大车牌,little-小车牌 -->
    <locationType opt="sceneLocation,frameLocation" />
    <!-- 定位类型, sceneLocation-场定位,frameLocation-帧定位 -->
    <recogType opt="sceneRecognition,frameRecognition" />
    <!-- 识别类型, sceneRecognition-场识别,frameRecognition-帧识别 -->
    <recogTime opt="daytime,night" /> <!-- 识别时间, daytime-白天,night-晚上 -->
    <sceneType opt="EPolice,gate" /> <!-- 场景类型, EPolice-电警,gate-卡口 -->
    <microPlateRecog opt="true,false" /> <!-- 微小车牌识别 -->
    <farmVehicleRecog opt="true,false" /> <!-- 农用车识别 -->
    <vechileColorRecog opt="true,false" /> <!-- 车身颜色识别 -->
    <motorCarRecog opt="true,false" /> <!-- 摩托车识别 -->
    <blurRecog opt="true,false" /> <!-- 模糊识别 -->
  </RecogMode>
  <vehicleLogoRecog opt="true,false" /> <!-- 车标识别 -->
</PlateRecogParam>
</VIAVirtualCoilParam>
</ITCTriggerCfg>
</Traffic>
<!-- 客流量统计 -->

```

```

<PDC>
  <PDCType opt="smart, professionalIntelligence" /> <!-- 客流智能类型，区分 Smart 设备和专业智能 iDS 设备 -->
  <PDCRule>
    <Region>
      <vertexNum min="4" max="10" /> <!-- 区域顶点数量 -->
    </Region>
    <enterDirection>true</enterDirection> <!-- 客流量进入方向 -->
    <dayStartTime>true</dayStartTime> <!-- 白天开始时间 -->
    <nightStartTime>true</nightStartTime> <!-- 夜晚开始时间 -->
    <AlarmTime>
      <timeSegNum min="2" max="8" /> <!-- 每天布防时间段数量 -->
    </AlarmTime>
    <alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp" /> <!-- req,处理类型 -->
    <detecteSensitive min="" max="" /> <!-- 目标检测灵敏度：范围 1-100，默认 50 -->
    <generatieSpeedSpace min="" max="" /> <!-- 目标生成速度（空域）：范围 1-100，默认 50 -->
    <generatieSpeedTime min="" max="" /> <!-- 目标生成速度（时域）：范围 1-100，默认 50 -->
    <countSpeed min="" max="" /> <!-- 计数速度：范围 1-100，默认 50 -->
    <detecteType opt="auto,head,shoulder" /> <!-- 目标检测类型：0-自动，1-人头，2-头肩，默认 0-自动 -->
    <targetSizeCorrect min="" max="" /> <!-- 目标尺寸修正：范围 1-100，默认 50 -->
  </PDCRule>
  <Calibration>
    <calibRectNum min="1" max="6" /> <!-- 标定框个数 -->
    <calibVerify>true</calibVerify> <!-- 支持标定校验 -->
  </Calibration>
  <resetCounter>true</resetCounter> <!-- 重置统计参数 -->
</PDC>
<HeatMapDetection>
<!-- req,热度图侦测 -->
  <enable opt="true,false" /> <!-- req,false:不使能，true:使能 -->
  <heatMapRegionNum>8</heatMapRegionNum> <!-- req,热度图侦测区域个数 -->
  <HeatMapParam>
    <!-- req,本节点会有相应的个数 -->
    <regionNum min="3" max="10" /> <!-- req,每个热度图侦测区域支持的有效点个数 -->
    <targetTrackEnable opt="true,false" /> <!-- req,目标跟随：0-关闭，1-开启，默认 0-关闭 -->
    <sensitivityLevel min="0" max="100" /> <!-- req,灵敏度 -->
    <backgroundUpdateRate min="1" max="100" /> <!-- req,背景更新速度：范围 1-100，默认 50 -->
    <sceneChangeLevel min="1" max="100" /> <!-- req,场景变化等级：范围 1-100，默认 50 -->
    <minTargetSize min="1" max="100" /> <!-- req,最小目标尺寸：范围 1-100，默认 50 -->
  </HeatMapParam>
  <alarmTime>8</alarmTime> <!-- req,布防时间段个数 -->
  <alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp" />
  <!-- req,处理类型,picture:抓图并邮件(此处由于早期能力缺少 uploadftp，后才加上，此处若解析不到该字符串，判断是否支持抓图传 FTP 再解析设备软硬件能力中的 UploadFTP 节点) -->
</HeatMapDetection>

```

```

<!-- 人脸 -->
<Face>
  <!-- 人脸检测 -->
  <FaceDetect>
    <eventType opt="abnormalFace,normalFace,multiFace" />
    <!-- abnormalFace-异常人脸, normalFace-正常人脸, multiFace-多张人脸 -->
    <uploadLastAlarm opt="false,true" /> <!-- 上传最近一次的报警 -->
    <uploadFacePic opt="false,true" /> <!-- 上传人脸子图 -->
    <picRecordEnable opt="false,true" /> <!-- 图片存储 -->
    <ruleNameLength min="0" max="32" /> <!-- 规则名称长度 -->
    <Region>
      <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
    <picProcType opt="notProcess,upload" /> <!-- 图片处理方式,no-process-不处理, upload-上传 -->
    <sensitivity min="1" max="100" default="10" /> <!-- 灵敏度 -->
    <duration min="4" max="60" default="30" /> <!-- 触发人脸报警时间阈值 -->
    <JpegParam>
      <picSize>true</picSize> <!-- 图片大小 -->
      <picQuality opt="best,better,normal" /> <!-- 图片质量,best-最好, better-较好, normal-一般 -->
    </JpegParam>
    <SizeFilter>
      <mode opt="imagePixel,realWorld,default" default="imagePixel" />
      <!-- 过滤模式, imagePixel-像素大小, realWorld-实际大小, default-默认 -->
      <minRect>true</minRect> <!-- 最小目标框 -->
      <maxRect>true</maxRect> <!-- 最大目标框 -->
    </SizeFilter>
    <AlarmTime>
      <timeSegNum min="2" max="8" /> <!-- 每天布防时间段数量 -->
    </AlarmTime>
    <alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp" />
    <!-- monitor-监视器上警告, audio-声音警告, center-上传中心, alarmout-触发报警输出, picture-jpeg 抓图并上传
    email,wirelesslight-无线声光报警, uploadftp-抓图并上传 ftp -->
    <relRecordChan>true</relRecordChan> <!-- 报警触发通道录像 -->
    <alarmDelay opt="5,10,30,60,120,300,600" /> <!-- 报警延时 -->

    <!-- 人脸画中画 -->
    <FaceInPicture>
      <enabled opt="true,false" /> <!-- 启用性 -->
      <backChannel>true</backChannel> <!-- 背景通道 -->
      <position opt="topLeft,topRight,bottomLeft,bottomRight" default="bottomRight" />
      <!-- 叠加位置, topLeft-左上,topRight-右上,bottomLeft-左下,bottomRight-右下 -->
      <division opt="1/4,1/9,1/16" /> <!-- 分屏系数 -->
    </FaceInPicture>
    <maxRelSnapChanNum>3</maxRelSnapChanNum> <!-- 最大关联抓图通道数 -->

```

```

</FaceDetect>
<!-- 人脸抓拍 -->
<FaceSnap>
  <snapTime min="0" max="10" default="5" /> <!-- 抓拍次数 -->
  <snapInterval min="0" max="255" default="24" /> <!-- 抓拍间隔, 单位: 帧 -->
  <snapThreshold min="0" max="100" default="80" /> <!-- 抓拍阈值 -->
  <generateRate min="1" max="5" default="3" /> <!-- 目标生成速度 -->
  <sensitivity min="1" max="5" default="5" /> <!-- 灵敏度 -->
  <referenceBright min="0" max="100" default="80" /> <!-- 参考亮度 -->
  <matchType opt="alarmRealtime,alarmAfterDisappear" default="alarmRealtime" />
  <!-- 比对报警模式, alarmRealtime-实时报警, alarmAfterDisappear-目标消失后报警 -->
  <matchThreshold min="0" max="100" default="80" /> <!-- 实时比对阈值 -->
  <JpegParam>
    <picSize>true</picSize> <!-- 图片大小 -->
    <picQuality opt="best,better,normal" /> <!-- 图片质量,best-最好, better-较好, normal-一般 -->
  </JpegParam>
  <RuleEntryList>
    <maxRuleNum>8</maxRuleNum> <!-- 最大规则数量 -->
    <RuleEntry>
      <SizeFilter>
        <mode opt="imagePixel,realWorld,default" default="imagePixel" />
        <!-- 过滤模式, imagePixel-像素大小, realWorld-实际大小, default-默认 -->
        <minRect>true</minRect> <!-- 最小目标框 -->
        <maxRect>true</maxRect> <!-- 最大目标框 -->
      </SizeFilter>
      <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
      </Region>
    </RuleEntry>
    <autoROI opt="false,true" /> <!-- 人脸自动 ROI 开关使能, false- 关闭, true- 开启 -->
  </RuleEntryList>
  <faceExposure opt="false,true" /> <!-- 人脸曝光使能 false- 关闭, true- 开启 -->
  <faceExposureMinDuration min="" max="" /> <!-- 人脸曝光最短时间 -->
</FaceSnap>
<!-- 抓拍接入参数 -->
<SnapDevAccess>
  <devIP opt="ipv4,ipv6" /> <!-- ip 地址 -->
  <devPort min="8000" max="65535" /> <!-- 端口 -->
  <usernameLength min="0" max="32" /> <!-- 用户名长度 -->
  <passwordLength min="0" max="16" /> <!-- 密码长度 -->
</SnapDevAccess>
<!-- 存储路径参数 -->
<SavePathCfg>
  <partitionNum>33</partitionNum> <!-- 分区个数 -->

```

```

<SinglePath>
  <type opt="snapFace,blocklistAlarm,snapFaceAndBlocklistAlarm" default="snapFaceAndBlocklistAlarm" />
  <!-- 存储类型, snapFace-抓拍人脸,blocklistAlarm-禁止名单报警,snapFaceAndBlocklistAlarm-抓拍人脸和、
  禁止名单报警 -->
  <saveAlarmPic>true</saveAlarmPic> <!-- 支持保存断网的报警图片 -->
  <diskDriver min="0" max="32" /> <!-- 盘符号 -->
  <reservedSpace min="10" max="20" default="10" /> <!-- 预留容量, 单位:G -->
</SinglePath>
</SavePathCfg>
</Face>
<!-- 视频质量诊断 -->
<VQD>
  <VQDEventType opt="blur,luma,chroma,snow,streak,freeze,signalLoss,ptzControl,sceneChange,videoAbnormal,
  videoBlock" />
  <!-- blur-图像模糊,luma-亮度异常,chroma-图像偏色,snow-雪花干扰,streak-条纹干扰,freeze-画面冻结,signalLoss-信号
  丢失,ptzControl-云台失控,sceneChange-场景突变,videoAbnormal-视频异常,videoBlock-视频遮挡 -->
  <VQDRule>
    <EventParam>
      <threshold min="0" max="100" /> <!-- 报警阈值 -->
      <triggerMode opt="continuous,single" /> <!-- 报警触发方式, continuous-持续触发,single-单次触发 -->
      <uploadPic opt="false,true" /> <!-- 报警图片上传 -->
      <timeInterval min="0" max="3600" default="120" /> <!-- 持续触发报警时间间隔, 单位: 秒 -->
    </EventParam>
    <AlarmTime>
      <timeSegNum min="2" max="8" /> <!-- 每天布防时间段数量 -->
    </AlarmTime>
    <alarmHandleType opt="monitor,audio,center,alarmout,picture,wirelesslight,uploadftp" />
    <!-- monitor-监视器上警告,audio-声音警告,center-上传中心,alarmout-触发报警输出,picture-jpeg 抓图并上传
    email,wirelesslight-无线声光报警, uploadftp-抓图并上传 ftp -->
    <relRecordChan>true</relRecordChan> <!-- 报警触发通道录像 -->
  </VQDRule>
  <downloadAlarmPic>true</downloadAlarmPic> <!-- 下载 VQD 报警图片 -->
</VQD>
<!-- 物体颜色 -->
<ObjectColor>
  <objectType opt="coat" /> <!-- 物体类型, coat-上衣 -->
  <colorMode opt="value,picture" default="value" /> <!-- 取色方式, value-颜色值, picture-图片 -->
  <colorValue>
    <brightness min="0" max="255" /> <!-- 亮度 -->
    <saturation min="0" max="255" /> <!-- 饱和度 -->
    <hue min="0" max="255" /> <!-- 色调 -->
  </colorValue>
  <colorPicture>
    <picType opt="jpg" /> <!-- 图片类型 -->
  </colorPicture>

```

```

        <picWidth min="100" max="400" /> <!-- 图片宽度 -->
        <picHeight min="100" max="300" /> <!-- 图片高度 -->
    </colorPicture>
</ObjectColor>
<!-- 辅助区域 -->
<AuxArea>
    <maxAreaNum>16</maxAreaNum> <!-- 区域个数 -->
    <areaType opt="overlapRegion,bedLocation" />
    <!-- 区域类型, overlapRegion-共同区域, bedLocation-床铺位置 -->
    <Region>
        <vertexNum min="3" max="10" /> <!-- 区域顶点数量 -->
    </Region>
</AuxArea>
<!-- 基准场景 -->
<BaselineScene>
    <operation opt="update" /> <!-- 操作, update-更新 -->
    <downloadBaselineScenePic>true</downloadBaselineScenePic> <!-- 基准场景图片下载 -->
</BaselineScene>
<channelWorkMode opt="independent,master,slave" />
<!-- 通道工作模式, independent-独立模式, master-主模式, slave-从模式 -->
<!-- 从通道 -->
<SlaveChannel>
    <maxChanNum>16</maxChanNum> <!-- 最大从通道个数 -->
    <chanType opt="local,remote" /> <!-- 从通道类型, local-本机通道, remote-远程设备通道 -->
    <RemoteChannel>
        <!-- 远程通道 -->
        <devAddress opt="ipv4,ipv6,domainName" /> <!-- 地址类型,domainName-设备域名 -->
        <devPort min="8000" max="65535" /> <!-- 端口 -->
        <userNameLength min="0" max="32" /> <!-- 用户名长度 -->
        <passwordLength min="0" max="16" /> <!-- 密码长度 -->
        <channelNumber>true</channelNumber> <!-- 通道号 -->
    </RemoteChannel>
</SlaveChannel>
<!-- 算法库关键字参数, 当前默认都是整型, 其他类型如浮点、字符串等需额外增加接口 -->
<!-- 翻译后缀含义: cn-中文,en-英文 -->
<VcaKeyParam>
    <VcaKeyParamEntry>
        <index>101</index> <!-- 关键字索引值 -->
        <PlainText>
            <translationCn>关键参数 1</translationCn> <!-- 中文名称 -->
            <translationEn>Key Parameter 1</translationEn> <!-- 英文名称 -->
        </PlainText>
        <value min="1" max="100" default="50" /> <!-- 取值范围 -->
    </VcaKeyParamEntry>

```

```

<VcaKeyParamEntry>
  <index>102</index> <!-- 关键字索引值 -->
  <PlainText>
    <translationCn>关键参数 2</translationCn> <!-- 中文名称 -->
    <translationEn>Key Parameter 2</translationEn> <!-- 英文名称 -->
  </PlainText>
  <value min="1" max="100" default="50" />
</VcaKeyParamEntry>
<VcaKeyParamEntry>
  <index>103</index> <!-- 关键字索引值 -->
  <PlainText>
    <translationCn>关键参数 3</translationCn> <!-- 中文名称 -->
    <translationEn>Key Parameter 3</translationEn> <!-- 英文名称 -->
  </PlainText>
  <value min="1" max="100" default="50" />
</VcaKeyParamEntry>
</VcaKeyParam>
</VcaChanAbility>

```

## 7.159 VehicleRecogAbility:智能通道分析能力集

### 输入描述:

```

<?xml version="1.0" encoding="utf-8" ?>
<!-- req, 获取车辆二次识别检测能力时 pinBuf 参数描述 -->
<VehicleRecogAbility version="2.0">
  <channelNO>1</channelNO>
  <!-- req,通道号-->
</VehicleRecogAbility>

```

### 能力集描述:

```

<?xml version="1.0" encoding="utf-8" ?>
<!-- req, 车辆二次识别检测能力集描述-->
<VehicleRecogAbility version="2.0">
  <channelNO>1</channelNO>
  <!-- req,通道号-->
  <dataIndex>true</dataIndex>
  <!-- req,数据流水号-->
  <taskNo>true</taskNo>
  <!-- req,任务处理号-->
  <plateRect>true</plateRect>
  <!-- req,车牌位置-->
  <plateNo>true</plateNo>
  <!-- req,车牌号码-->
  <recogOperate opt="plate,vehicle,carColor,vehicleLogo, copilotSafebelt, pilotSafebelt, pilotSunVisor, copilotSunVisor" />

```

```

<!-- req,车辆图片二次识别操作: 车牌、车辆类型、车身颜色、车辆品牌、安全带识别(副驾驶)、安全带识别(主驾驶)、遮
挡板识别(主驾驶)、遮挡板识别(副驾驶)-->
<picUploadMethods opt="sdk,URL" />
<!-- req,图片数据上传类型-->
</VehicleRecogAbility>

```

## 7.160 设备日志信息

### dwMajorType 主类型

宏定义	宏定义值	含义
MAJOR_ALARM	0x1	报警
MAJOR_EXCEPTION	0x2	异常
MAJOR_OPERATION	0x3	操作
MAJOR_INFORMATION	0x4	日志附加信息

### dwMinorType 次类型

主类型的宏定义	宏定义值	含义
MAJOR_ALARM	0x1	报警
次类型的宏定义	宏定义值	含义
MINOR_ALARM_IN	0x1	报警输入
MINOR_ALARM_OUT	0x2	报警输出
MINOR_MOTDET_START	0x3	移动侦测报警开始
MINOR_MOTDET_STOP	0x4	移动侦测报警结束
MINOR_HIDE_ALARM_START	0x5	遮挡报警开始
MINOR_HIDE_ALARM_STOP	0x6	遮挡报警结束
MINOR_VCA_ALARM_START	0x7	智能报警开始
MINOR_VCA_ALARM_STOP	0x8	智能报警结束
MINOR_ITS_ALARM_START	0x09	交通事件报警开始
MINOR_ITS_ALARM_STOP	0x0a	交通事件报警结束
MINOR_NETALARM_START	0x0b	网络报警开始
MINOR_NETALARM_STOP	0x0c	网络报警结束
MINOR_NETALARM_RESUME	0x0d	网络报警恢复

主类型的宏定义	宏定义值	含义
MAJOR_EXCEPTION	0x2	异常
次类型的宏定义	宏定义值	含义
MINOR_RAID_ERROR	0x20	阵列异常

MINOR_VI_LOST	0x21	视频信号丢失
MINOR_ILLEGAL_ACCESS	0x22	非法访问
MINOR_HD_FULL	0x23	硬盘满
MINOR_HD_ERROR	0x24	硬盘错误
MINOR_DCD_LOST	0x25	MODEM 掉线(保留)
MINOR_IP_CONFLICT	0x26	IP 地址冲突
MINOR_NET_BROKEN	0x27	网络断开
MINOR_REC_ERROR	0x28	录像出错
MINOR_IPC_NO_LINK	0x29	IPC 连接异常
MINOR_VI_EXCEPTION	0x2a	视频输入异常(只针对模拟通道)
MINOR_IPC_IP_CONFLICT	0x2b	IPC 的 IP 地址冲突
MINOR_SENCE_EXCEPTION	0x2c	场景异常
MINOR_PIC_REC_ERROR	0x2d	抓图出错,获取图片文件失败
MINOR_VI_MISMATCH	0x2e	视频制式不匹配

主类型的宏定义	宏定义值	含义
MAJOR_OPERATION	0x3	操作
次类型的宏定义	宏定义值	含义
MINOR_START_DVR	0x41	开机
MINOR_STOP_DVR	0x42	关机
MINOR_STOP_ABNORMAL	0x43	异常关机
MINOR_REBOOT_DVR	0x44	本地重启设备
MINOR_LOCAL_LOGIN	0x50	本地登陆
MINOR_LOCAL_LOGOUT	0x51	本地注销登陆
MINOR_LOCAL_CFG_PARM	0x52	本地配置参数
MINOR_LOCAL_PLAYBYFILE	0x53	本地按文件回放或下载
MINOR_LOCAL_PLAYBYTIME	0x54	本地按时间回放或下载
MINOR_LOCAL_START_REC	0x55	本地开始录像
MINOR_LOCAL_STOP_REC	0x56	本地停止录像
MINOR_LOCAL_PTZCTRL	0x57	本地云台控制
MINOR_LOCAL_PREVIEW	0x58	本地预览(保留不使用)
MINOR_LOCAL_MODIFY_TIME	0x59	本地修改时间(保留不使用)
MINOR_LOCAL_UPGRADE	0x5a	本地升级
MINOR_LOCAL_RECFILE_OUTPUT	0x5b	本地备份录像文件
MINOR_LOCAL_FORMAT_HDD	0x5c	本地初始化硬盘
MINOR_LOCAL_CFGFILE_OUTPUT	0x5d	导出本地配置文件
MINOR_LOCAL_CFGFILE_INPUT	0x5e	导入本地配置文件

MINOR_LOCAL_COPYFILE	0x5f	本地备份文件
MINOR_LOCAL_LOCKFILE	0x60	本地锁定录像文件
MINOR_LOCAL_UNLOCKFILE	0x61	本地解锁录像文件
MINOR_LOCAL_DVR_ALARM	0x62	本地手动清除和触发报警
MINOR_IPC_ADD	0x63	本地添加 IPC
MINOR_IPC_DEL	0x64	本地删除 IPC
MINOR_IPC_SET	0x65	本地设置 IPC
MINOR_LOCAL_START_BACKUP	0x66	本地开始备份
MINOR_LOCAL_STOP_BACKUP	0x67	本地停止备份
MINOR_LOCAL_COPYFILE_START_TIME	0x68	本地备份开始时间
MINOR_LOCAL_COPYFILE_END_TIME	0x69	本地备份结束时间
MINOR_LOCAL_ADD_NAS	0x6a	本地添加网络硬盘
MINOR_LOCAL_DEL_NAS	0x6b	本地删除 NAS 盘
MINOR_LOCAL_SET_NAS	0x6c	本地设置 NAS 盘
MINOR_REMOTE_LOGIN	0x70	远程登录
MINOR_REMOTE_LOGOUT	0x71	远程注销登陆
MINOR_REMOTE_START_REC	0x72	远程开始录像
MINOR_REMOTE_STOP_REC	0x73	远程停止录像
MINOR_START_TRANS_CHAN	0x74	开始透明传输
MINOR_STOP_TRANS_CHAN	0x75	停止透明传输
MINOR_REMOTE_GET_PARM	0x76	远程获取参数
MINOR_REMOTE_CFG_PARM	0x77	远程配置参数
MINOR_REMOTE_GET_STATUS	0x78	远程获取状态
MINOR_REMOTE_ARM	0x79	远程布防
MINOR_REMOTE_DISARM	0x7a	远程撤防
MINOR_REMOTE_REBOOT	0x7b	远程重启
MINOR_START_VT	0x7c	开始语音对讲
MINOR_STOP_VT	0x7d	停止语音对讲
MINOR_REMOTE_UPGRADE	0x7e	远程升级
MINOR_REMOTE_PLAYBYFILE	0x7f	远程按文件回放
MINOR_REMOTE_PLAYBYTIME	0x80	远程按时间回放
MINOR_REMOTE_PTZCTRL	0x81	远程云台控制
MINOR_REMOTE_FORMAT_HDD	0x82	远程格式化硬盘
MINOR_REMOTE_STOP	0x83	远程关机
MINOR_REMOTE_LOCKFILE	0x84	远程锁定文件
MINOR_REMOTE_UNLOCKFILE	0x85	远程解锁文件
MINOR_REMOTE_CFGFILE_OUTPUT	0x86	远程导出配置文件

MINOR_REMOTE_CFGFILE_INTPUT	0x87	远程导入配置文件
MINOR_REMOTE_RECFILE_OUTPUT	0x88	远程导出录像文件
MINOR_REMOTE_DVR_ALARM	0x89	远程手动清除和触发报警
MINOR_REMOTE_IPC_ADD	0x8a	远程添加 IPC
MINOR_REMOTE_IPC_DEL	0x8b	远程删除 IPC
MINOR_REMOTE_IPC_SET	0x8c	远程设置 IPC
MINOR_REBOOT_VCA_LIB	0x8d	重启智能库
MINOR_REMOTE_ADD_NAS	0x8e	远程添加 NAS 盘
MINOR_REMOTE_DEL_NAS	0x8f	远程删除 NAS 盘
MINOR_REMOTE_SET_NAS	0x90	远程设置 NAS 盘
MINOR_LOCAL_START_REC_CDRW	0x91	本地开始刻录
MINOR_LOCAL_STOP_REC_CDRW	0x92	本地停止刻录
MINOR_REMOTE_START_REC_CDRW	0x93	远程开始刻录
MINOR_REMOTE_STOP_REC_CDRW	0x94	远程停止刻录
MINOR_LOCAL_PIC_OUTPUT	0x95	本地备份图片文件
MINOR_REMOTE_SET_NAS	0x96	远程备份图片文件
MINOR_LOCAL_INQUEST_RESUME	0x97	本地恢复审讯事件
MINOR_REMOTE_INQUEST_RESUME	0x98	远程恢复审讯事件
MINOR_REMOTE_BYPASS	0xd0	远程旁路
MINOR_REMOTE_UNBYPASS	0xd1	远程旁路恢复
MINOR_REMOTE_SET_ALARMIN_CFG	0xd2	远程设置报警输入参数
MINOR_REMOTE_GET_ALARMIN_CFG	0xd3	远程获取报警输入参数
MINOR_REMOTE_SET_ALARMOUT_CFG	0xd4	远程设置报警输出参数
MINOR_REMOTE_GET_ALARMOUT_CFG	0xd5	远程获取报警输出参数
MINOR_REMOTE_ALARMOUT_OPEN_MAN	0xd6	远程手动开启报警输出
MINOR_REMOTE_ALARMOUT_CLOSE_MAN	0xd7	远程手动关闭报警输出
MINOR_REMOTE_ALARM_ENABLE_CFG	0xd8	远程设置报警主机的 RS485 串口使能状态
MINOR_DBDATA_OUTPUT	0xd9	导出数据库记录
MINOR_DBDATA_INPUT	0xda	导入数据库记录
MINOR_MU_SWITCH	0xdb	级联切换
MINOR_MU_PTZ	0xdc	级联 PTZ 控制
MINOR_LOCAL_CONF_REB_RAID	0x101	本地配置自动重建
MINOR_LOCAL_CONF_SPARE	0x102	本地配置热备
MINOR_LOCAL_ADD_RAID	0x103	本地创建阵列
MINOR_LOCAL_DEL_RAID	0x104	本地删除阵列
MINOR_LOCAL_MIG_RAID	0x105	本地迁移阵列
MINOR_LOCAL_REB_RAID	0x106	本地手动重建阵列

MINOR_LOCAL_QUICK_CONF_RAID	0x107	本地一键配置
MINOR_LOCAL_ADD_VD	0x108	本地创建虚拟磁盘
MINOR_LOCAL_DEL_VD	0x109	本地删除虚拟磁盘
MINOR_LOCAL_RP_VD	0x10a	本地修复虚拟磁盘
MINOR_LOCAL_FORMAT_EXPANDVD	0x10b	本地扩展虚拟磁盘扩容
MINOR_LOCAL_RAID_UPGRADE	0x10c	本地 raid 卡升级
MINOR_LOCAL_STOP_RAID	0x10d	本地暂停 RAID 操作(即安全拔盘)
MINOR_REMOTE_CONF_REB_RAID	0x111	远程配置自动重建
MINOR_REMOTE_CONF_SPARE	0x112	远程配置热备
MINOR_REMOTE_ADD_RAID	0x113	远程创建阵列
MINOR_REMOTE_DEL_RAID	0x114	远程删除阵列
MINOR_REMOTE_MIG_RAID	0x115	远程迁移阵列
MINOR_REMOTE_REB_RAID	0x116	远程手动重建阵列
MINOR_REMOTE_QUICK_CONF_RAID	0x117	远程一键配置
MINOR_REMOTE_ADD_VD	0x118	远程创建虚拟磁盘
MINOR_REMOTE_DEL_VD	0x119	远程删除虚拟磁盘
MINOR_REMOTE_RP_VD	0x11a	远程修复虚拟磁盘
MINOR_REMOTE_FORMAT_EXPANDVD	0x11b	远程虚拟磁盘扩容
MINOR_REMOTE_RAID_UPGRADE	0x11c	远程 raid 卡升级
MINOR_REMOTE_STOP_RAID	0x11d	远程暂停 RAID 操作(即安全拔盘)
MINOR_LOCAL_START_PIC_REC	0x121	本地开始抓图
MINOR_LOCAL_STOP_PIC_REC	0x122	本地停止抓图
MINOR_LOCAL_SET_SNMP	0x125	本地配置 SNMP
MINOR_LOCAL_TAG_OPT	0x126	本地标签操作
MINOR_REMOTE_START_PIC_REC	0x131	远程开始抓图
MINOR_REMOTE_STOP_PIC_REC	0x132	远程停止抓图
MINOR_REMOTE_SET_SNMP	0x135	远程配置 SNMP
MINOR_REMOTE_TAG_OPT	0x136	远程标签操作
MINOR_REMOTE_TAG_OPT	0x136	远程标签操作
MINOR_LOCAL_VOUT_SWITCH	0x140	本地输出口切换操作
MINOR_SET_TRIGGERMODE_CFG	0x1001	设置触发模式参数
MINOR_GET_TRIGGERMODE_CFG	0x1002	获取触发模式参数
MINOR_SET_IOOUT_CFG	0x1003	设置 IO 输出参数
MINOR_GET_IOOUT_CFG	0x1004	获取 IO 输出参数
MINOR_GET_TRIGGERMODE_DEFAULT	0x1005	获取触发模式推荐参数
MINOR_GET_ITCSTATUS	0x1006	获取状态检测参数
MINOR_SET_STATUS_DETECT_CFG	0x1007	设置状态检测参数

MINOR_GET_STATUS_DETECT_CFG	0x1008	获取状态检测参数
MINOR_GET_VIDEO_TRIGGERMODE_CFG	0x1009	获取视频电警模式参数
MINOR_SET_VIDEO_TRIGGERMODE_CFG	0x100a	设置视频电警模式参数
MINOR_LOCAL_ADD_CAR_INFO	0x2001	本地添加车辆信息
MINOR_LOCAL_MOD_CAR_INFO	0x2002	本地修改车辆信息
MINOR_LOCAL_DEL_CAR_INFO	0x2003	本地删除车辆信息
MINOR_LOCAL_FIND_CAR_INFO	0x2004	本地查找车辆信息
MINOR_LOCAL_ADD_MONITOR_INFO	0x2005	本地添加布控信息
MINOR_LOCAL_MOD_MONITOR_INFO	0x2006	本地修改布控信息
MINOR_LOCAL_DEL_MONITOR_INFO	0x2007	本地删除布控信息
MINOR_LOCAL_FIND_MONITOR_INFO	0x2008	本地查询布控信息
MINOR_LOCAL_FIND_NORMAL_PASS_INFO	0x2009	本地查询正常通行信息
MINOR_LOCAL_FIND_ABNORMAL_PASS_INFO	0x200a	本地查询异常通行信息
MINOR_LOCAL_FIND_PEDESTRIAN_PASS_INFO	0x200b	本地查询正常通行信息
MINOR_LOCAL_PIC_PREVIEW	0x200c	本地图片预览
MINOR_LOCAL_SET_GATE_PARM_CFG	0x200d	设置本地配置出入口参数
MINOR_LOCAL_GET_GATE_PARM_CFG	0x200e	获取本地配置出入口参数
MINOR_LOCAL_SET_DATAUPLOAD_PARM_CFG	0x200f	设置本地配置数据上传参数
MINOR_LOCAL_GET_DATAUPLOAD_PARM_CFG	0x2010	获取本地配置数据上传参数
MINOR_LOCAL_DEVICE_CONTROL	0x2011	本地设备控制(本地开关闸)
MINOR_LOCAL_ADD_EXTERNAL_DEVICE_INFO	0x2012	本地添加外接设备信息
MINOR_LOCAL_MOD_EXTERNAL_DEVICE_INFO	0x2013	本地修改外接设备信息
MINOR_LOCAL_DEL_EXTERNAL_DEVICE_INFO	0x2014	本地删除外接设备信息
MINOR_LOCAL_FIND_EXTERNAL_DEVICE_INFO	0x2015	本地查询外接设备信息
MINOR_LOCAL_ADD_CHARGE_RULE	0x2016	本地添加收费规则
MINOR_LOCAL_MOD_CHARGE_RULE	0x2017	本地修改收费规则
MINOR_LOCAL_DEL_CHARGE_RULE	0x2018	本地删除收费规则
MINOR_LOCAL_FIND_CHARGE_RULE	0x2019	本地查询收费规则
MINOR_LOCAL_COUNT_NORMAL_CURRENTINFO	0x2020	本地统计正常通行信息
MINOR_LOCAL_EXPORT_NORMAL_CURRENTINFO_REPORT	0x2021	本地导出正常通行信息统计报表
MINOR_LOCAL_COUNT_ABNORMAL_CURRENTINFO	0x2022	本地统计异常通行信息
MINOR_LOCAL_EXPORT_ABNORMAL_CURRENTINFO_REPORT	0x2023	本地导出异常通行信息统计报表
MINOR_LOCAL_COUNT_PEDESTRIAN_CURRENTINFO	0x2024	本地统计行人通行信息
MINOR_LOCAL_EXPORT_PEDESTRIAN_CURRENTINFO_REPORT	0x2025	本地导出行人通行信息统计报表
MINOR_LOCAL_FIND_CAR_CHARGEINFO	0x2026	本地查询过车收费信息
MINOR_LOCAL_COUNT_CAR_CHARGEINFO	0x2027	本地统计过车收费信息
MINOR_LOCAL_EXPORT_CAR_CHARGEINFO_REPORT	0x2028	本地导出过车收费信息统计报表

MINOR_LOCAL_FIND_SHIFTINFO	0x2029	本地查询交接班信息
MINOR_LOCAL_FIND_CARDINFO	0x2030	本地查询卡片信息
MINOR_LOCAL_ADD_RELIEF_RULE	0x2031	本地添加减免规则
MINOR_LOCAL_MOD_RELIEF_RULE	0x2032	本地修改减免规则
MINOR_LOCAL_DEL_RELIEF_RULE	0x2033	本地删除减免规则
MINOR_LOCAL_FIND_RELIEF_RULE	0x2034	本地查询减免规则
MINOR_LOCAL_GET_ENDETCFG	0x2035	本地获取出入口控制机离线检测配置
MINOR_LOCAL_SET_ENDETCFG	0x2036	本地设置出入口控制机离线检测配置
MINOR_LOCAL_SET_ENDEV_ISSUEDDATA	0x2037	本地设置出入口控制机下发卡片信息
MINOR_LOCAL_DEL_ENDEV_ISSUEDDATA	0x2038	本地清空出入口控制机下发卡片信息
MINOR_REMOTE_DEVICE_CONTROL	0x2101	远程设备控制
MINOR_REMOTE_SET_GATE_PARM_CFG	0x2102	设置远程配置出入口参数
MINOR_REMOTE_GET_GATE_PARM_CFG	0x2103	获取远程配置出入口参数
MINOR_REMOTE_SET_DATAUPLOAD_PARM_CFG	0x2104	设置远程配置数据上传参数
MINOR_REMOTE_GET_DATAUPLOAD_PARM_CFG	0x2105	获取远程配置数据上传参数
MINOR_REMOTE_GET_BASE_INFO	0x2106	远程获取终端基本信息
MINOR_REMOTE_GET_OVERLAP_CFG	0x2107	远程获取字符叠加参数配置
MINOR_REMOTE_SET_OVERLAP_CFG	0x2108	远程设置字符叠加参数配置
MINOR_REMOTE_GET_ROAD_INFO	0x2109	远程获取路口信息
MINOR_REMOTE_START_TRANSCHAN	0x210a	远程建立同步数据服务器
MINOR_REMOTE_GET_ECTWORKSTATE	0x210b	远程获取出入口终端工作状态
MINOR_REMOTE_GET_ECTCHANINFO	0x210c	远程获取出入口终端通道状态
MINOR_REMOTE_ADD_EXTERNAL_DEVICE_INFO	0x210d	远程添加外接设备信息
MINOR_REMOTE_MOD_EXTERNAL_DEVICE_INFO	0x210e	远程修改外接设备信息
MINOR_REMOTE_GET_ENDETCFG	0x210f	远程获取出入口控制机离线检测配置
MINOR_REMOTE_SET_ENDETCFG	0x2110	远程设置出入口控制机离线检测配置
MINOR_REMOTE_ENDEV_ISSUEDDATA	0x2111	远程设置出入口控制机下发卡片信息
MINOR_REMOTE_DEL_ENDEV_ISSUEDDATA	0x2112	远程清空出入口控制机下发卡片信息
MINOR_REMOTE_ON_CTRL_LAMP	0x2115	开启远程控制车位指示灯
MINOR_REMOTE_OFF_CTRL_LAMP	0x2116	关闭远程控制车位指示灯

主类型的宏定义	宏定义值	含义
MAJOR_INFORMATION	0x4	附加信息
次类型的宏定义	宏定义值	含义
MINOR_HDD_INFO	0xa1	硬盘信息
MINOR_SMART_INFO	0xa2	S.M.A.R.T 信息
MINOR_REC_START	0xa3	开始录像

MINOR_REC_STOP	0xa4	停止录像
MINOR_REC_OVERDUE	0xa5	过期录像删除
MINOR_LINK_START	0xa6	连接前端设备
MINOR_LINK_STOP	0xa7	断开前端设备
MINOR_NET_DISK_INFO	0xa8	网络硬盘信息
MINOR_RAID_INFO	0xa9	raid 相关信息
MINOR_PIC_REC_START	0xb3	开始抓图
MINOR_PIC_REC_STOP	0xb4	停止抓图
MINOR_PIC_REC_OVERDUE	0xb5	过期图片文件删除

## 7.161 违章类型国标定义

违法代码	违法行为描述	简称	补充说明
1349	驾驶中型以上载客汽车在高速公路、城市快速路以外的道路上行驶超过规定时速未达 20%的	超速行驶	1. 对道路设置属性： 高速公路/城市快速路/其他道路； 2. 基于车型识别区分车辆类型： 黄牌货车=中型以上载货车； 黄牌客车=中型以上载客车； 校车暂不能识别，归类为客车； 危险品车辆暂不能识别，归类为货车； 其他车辆=“中型以上载客载货汽车、校车、危险物品运输车辆以外的”
1350	驾驶中型以上载货汽车在高速公路、城市快速路以外的道路上行驶超过规定时速未达 20%的	超速行驶	
1352	驾驶中型以上载客载货汽车、危险物品运输车辆以外的其他机动车行驶超过规定时速未达 20%的	超速行驶	
1632	驾驶中型以上载客汽车在高速公路、城市快速路以外的道路上行驶超过规定时速 20%以上未达到 50%的	超速行驶	
1633	驾驶中型以上载货汽车在高速公路、城市快速路以外的道路上行驶超过规定时速 20%以上未达到 50%的	超速行驶	
1634	驾驶校车在高速公路、城市快速路以外的道路上行驶超过规定时速 20%以上未达到 50%的		
1636	驾驶中型以上载客载货汽车、校车、危险物品运输车辆以外的其他机动车行驶超过规定时速 20%以上未达到 50%的	超速行驶	
1628	驾驶中型以上载客汽车在城市快速路上行驶超过规定时速未达 20%的	超速行驶	
1629	驾驶中型以上载货汽车在城市快速路上行驶超过规定时速未达 20%的	超速行驶	
1630	驾驶校车在城市快速路上行驶超过规定时速未达 20%的		
1631	驾驶危险物品运输车辆在快速路上行驶超过规定时速未达 20%的		
1722	驾驶中型以上载客汽车在城市快速路上行驶超过规定时速 20%以上未达 50%的	超速行驶	
1723	驾驶中型以上载货汽车在城市快速路上行驶超过规定时速 20%以上未达 50%的	超速行驶	
1724	驾驶校车在城市快速路上行驶超过规定时速 20%以上未达 50%	超速行驶	

	的		
1721	驾驶中型以上载客载货汽车、校车、危险物品运输车辆以外的机动车行驶超过规定时速 50% 以上的	超速行驶	
1726	驾驶中型以上载客汽车在高速公路以外的道路上行驶超过规定时速 50% 的	超速行驶	
1727	驾驶中型以上载货汽车在高速公路以外的道路上行驶超过规定时速 50% 的	超速行驶	
1728	驾驶校车在高速公路以外的道路上行驶超过规定时速 50% 的	超速行驶	
4710	驾驶中型以上载客汽车在高速公路上行驶超过规定时速 50% 的	超速行驶	
4711	驾驶中型以上载货汽车在高速公路上行驶超过规定时速 50% 的	超速行驶	
4712	驾驶校车在高速公路上行驶超过规定时速 50% 的		
4713	驾驶危险物品运输车辆在高速公路上行驶超过规定时速 50% 的		
4609	驾驶中型以上载客汽车在高速公路上行驶超过规定时速未达 20% 的	超速行驶	
4610	驾驶中型以上载货汽车在高速公路上行驶超过规定时速未达 20% 的	超速行驶	
4706	驾驶中型以上载客汽车在高速公路上行驶超过规定时速 20% 以上未达 50% 的	超速行驶	
4707	驾驶中型以上载货汽车在高速公路上行驶超过规定时速 20% 以上未达 50% 的	超速行驶	
1625	驾驶机动车违反道路交通信号灯通行的	违反信号灯	
4615	低能见度气象条件下，驾驶机动车在高速公路上不按规定行驶的	不按规定行驶	
1344	机动车违反禁令标志指示的	违反禁令标志	
13451	机动车违反禁止标线指示的	压线	
13453	机动车违反禁止标线指示的	违法变道	
1018	机动车不在机动车道内行驶的	机占非	
1019	机动车违反规定使用专用车道的	占用专用车道	
1208	机动车通过有灯控路口时，不按所需行进方向驶入导向车道的	不按导向车道行驶	
1211	通过路口遇停止信号时，停在停止线以内或路口内的	路口停车	
1301	机动车逆向行驶的	逆行	
4306	在高速公路上正常情况下以低于规定最低时速行驶的	低速行驶	
16251	信号灯为绿灯信号时，车辆未通过路口停车	绿灯停车	
1228	路口遇有交通阻塞时未依次等候的	路口滞留	
1039	机动车未在规定地点停放	城市违停	
1357	遇行人正在通过人行横道时未停车让行的	未礼让行人	
1240	驾驶人未按规定使用安全带的	未系安全带	
4613	高速公路违法停车	违法停车	
4701	违法倒车	违法倒车	

