

# 设备(智能交通摄像机)

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 .....	17
3.1 SDK 接口调用主要流程 .....	17
3.2 报警（布防）流程 .....	18
3.3 报警（监听）流程 .....	19
4 函数调用实例 .....	20
4.1 参数配置模块的示例代码 .....	20
4.2 抓拍图片上传示例代码 .....	21
4.3 抓拍机实时状态检测结果上传 .....	27
5 函数说明 .....	29
5.1 SDK 初始化 .....	29
5.1.1 初始化 SDK NET_DVR_Init .....	29
5.1.2 释放 SDK 资源 NET_DVR_Cleanup .....	29
5.2 SDK 本地功能 .....	29
SDK 本地参数配置 .....	29
5.2.1 获取 SDK 本地参数 NET_DVR_GetSDKLocalCfg .....	29
5.2.2 设置 SDK 本地参数 NET_DVR_SetSDKLocalCfg .....	30
连接和接收超时时间及重连设置 .....	31
5.2.3 设置网络连接超时时间和连接尝试次数 NET_DVR_SetConnectTime .....	31
5.2.4 设置重连功能 NET_DVR_SetReconnect .....	31
5.2.5 设置接收超时时间 NET_DVR_SetRecvTimeOut .....	31
多网卡绑定 .....	31
5.2.6 获取所有 IP，用于支持多网卡接口 NET_DVR_GetLocalIP .....	31
5.2.7 设置 IP 绑定 NET_DVR_SetValidIP .....	32
SDK 版本、状态和能力 .....	32
5.2.8 获取 SDK 的版本号和 build 信息 NET_DVR_GetSDKBuildVersion .....	32
5.2.9 获取当前 SDK 的状态信息 NET_DVR_GetSDKState .....	32
5.2.10 获取当前 SDK 的功能信息 NET_DVR_GetSDKAbility .....	32
SDK 启用写日志 .....	33
5.2.11 启用写日志文件 NET_DVR_SetLogToFile .....	33
异常消息回调 .....	33
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 .....	33
获取错误信息 .....	35
5.2.13 返回最后操作的错误码 NET_DVR_GetLastError .....	35
5.2.14 返回最后操作的错误码信息 NET_DVR_GetErrorMsg .....	35
5.3 用户注册 .....	36
5.3.1 激活设备 NET_DVR_ActivateDevice .....	36
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 .....	36
5.3.3 用户注册设备 NET_DVR_Login_V40 .....	37

5.3.4	用户注销 NET_DVR_Logout .....	37
5.4	获取设备能力集 .....	37
5.4.1	获取设备能力集 NET_DVR_GetDeviceAbility .....	37
5.4.2	获取设备能力集 NET_DVR_GetSTDAbility .....	39
5.5	实时预览 .....	39
5.5.1	主码流动态产生一个关键帧 NET_DVR_MakeKeyFrame .....	39
5.5.2	子码流动态产生一个关键帧 NET_DVR_MakeKeyFrameSub .....	39
5.5.3	实时预览 NET_DVR_RealPlay_V40 .....	40
5.5.4	停止预览 NET_DVR_StopRealPlay .....	40
5.5.5	获取预览时用来解码和显示的播放库句柄 NET_DVR_GetRealPlayerIndex .....	41
5.6	预览时播放声音控制 .....	41
5.6.1	设置声音播放模式 NET_DVR_SetAudioMode .....	41
5.6.2	独占声卡模式下开启声音 NET_DVR_OpenSound .....	41
5.6.3	独占声卡模式下开启声音 NET_DVR_CloseSound .....	41
5.6.4	共享声卡模式下开启声音 NET_DVR_OpenSoundShare .....	42
5.6.5	共享声卡模式下关闭声音 NET_DVR_CloseSoundShare .....	42
5.6.6	调节播放音量 NET_DVR_Volume .....	42
5.7	实时预览数据捕获 .....	42
5.7.1	注册回调函数，捕获实时码流数据 NET_DVR_SetRealDataCallBack .....	42
5.7.2	捕获数据并保存到指定的文件中 NET_DVR_SaveRealData .....	43
5.7.3	停止数据捕获 NET_DVR_StopSaveRealData .....	43
5.8	预览时抓图 .....	44
5.8.1	设置抓图模式 NET_DVR_SetCapturePictureMode .....	44
5.8.2	预览时，单帧数据捕获图片并保存成文件 NET_DVR_CapturePicture .....	44
5.9	布防、撤防 .....	44
	设置报警等信息上传的回调函数 .....	44
5.9.1	注册报警回调函数 NET_DVR_SetDVRMessageCallBack_V31 .....	44
	布防撤防 .....	46
5.9.2	建立报警上传通道，获取报警等信息 NET_DVR_SetupAlarmChan_V41 .....	46
5.9.3	撤销报警上传通道 NET_DVR_CloseAlarmChan_V30 .....	46
5.10	监听报警 .....	46
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 .....	46
5.10.2	停止监听（支持多线程） NET_DVR_StopListen_V30 .....	48
5.11	远程参数配置 .....	48
	通用参数配置 .....	48
5.11.1	获取设备的配置信息 NET_DVR_GetDVRConfig .....	48
5.11.2	设置设备的配置信息 NET_DVR_SetDVRConfig .....	49
5.11.3	批量获取配置信息 NET_DVR_GetDeviceConfig .....	50
5.11.4	批量设置配置信息 NET_DVR_SetDeviceConfig .....	51
	抓拍参数配置 .....	52
5.11.5	获取设备的配置信息 NET_DVR_GetDVRConfig .....	52
5.11.6	设置设备的配置信息 NET_DVR_SetDVRConfig .....	53
5.11.7	批量获取配置信息 NET_DVR_GetDeviceConfig .....	55
5.11.8	批量设置配置信息 NET_DVR_SetDeviceConfig .....	56

5.11.9	获取设备参数 NET_DVR_GetSTDConfig .....	57
5.11.10	设置设备参数 NET_DVR_SetSTDConfig .....	58
	长连接参数配置 .....	58
5.11.11	启动长连接远程配置 NET_DVR_StartRemoteConfig .....	58
5.11.12	逐个获取查找到的结果信息 NET_DVR_GetNextRemoteConfig .....	59
5.11.13	发送长连接数据 NET_DVR_SendRemoteConfig .....	60
5.11.14	关闭长连接配置 NET_DVR_StopRemoteConfig .....	61
	获取设备支持的云台协议 .....	61
5.11.15	获取设备支持的云台协议 NET_DVR_GetPTZProtocol .....	61
5.12	证书管理 .....	61
	证书创建、删除 .....	61
5.12.1	远程控制 NET_DVR_RemoteControl .....	61
	证书信息获取 .....	62
5.12.2	批量获取配置信息 NET_DVR_GetDeviceConfig .....	62
	证书上传下载 .....	62
5.12.3	上传文件 NET_DVR_UploadFile_V40 .....	62
5.12.4	获取文件上传的进度和状态 NET_DVR_GetUploadState .....	63
5.12.5	停止文件上传 NET_DVR_UploadClose .....	63
5.12.6	开始下载文件 NET_DVR_StartDownload .....	63
5.12.7	获取文件下载的进度和状态 NET_DVR_GetDownloadState .....	64
5.12.8	停止文件下载 NET_DVR_StopDownload .....	64
5.13	远程控制 .....	64
5.13.1	远程控制 NET_DVR_RemoteControl .....	64
5.14	手动触发抓拍 .....	65
5.14.1	网络触发抓拍 NET_DVR_ContinuousShoot .....	65
5.14.2	手动抓拍 NET_DVR_ManualSnap .....	65
5.15	雨刷控制 .....	65
5.15.1	云台控制操作 NET_DVR_PTZControl_Other .....	65
5.16	图片的查找、下载 .....	66
	查找图片 .....	66
5.16.1	根据类型和时间查找图片 NET_DVR_FindPicture .....	66
5.16.2	逐个获取查找到的图片 NET_DVR_FindNextPicture .....	66
5.16.3	关闭图片查找，释放资源 NET_DVR_CloseFindPicture .....	67
	图片下载 .....	67
5.16.4	获取图片数据并存放在指定的内存空间中 NET_DVR_GetPicture_V30 .....	67
5.16.5	图片下载 NET_DVR_GetPicture .....	67
5.17	文件上传 .....	68
5.17.1	上传文件 NET_DVR_UploadFile_V40 .....	68
5.17.2	获取文件上传的进度和状态 NET_DVR_GetUploadState .....	68
5.17.3	停止文件上传，释放资源 NET_DVR_UploadClose .....	68
5.18	获取设备状态 .....	69
5.18.1	获取设备状态信息 NET_DVR_GetDeviceStatus .....	69
5.19	设备维护管理 .....	69
	在线状态检测 .....	69

5.19.1	设备在线状态检测 NET_DVR_RemoteControl.....	69
	远程升级.....	70
5.19.2	设置远程升级时网络环境 NET_DVR_SetNetworkEnvironment.....	70
5.19.3	远程升级 NET_DVR_Upgrade .....	70
5.19.4	获取远程升级的进度 NET_DVR_GetUpgradeProgress .....	71
5.19.5	获取远程升级的状态 NET_DVR_GetUpgradeState.....	71
5.19.6	关闭远程升级句柄, 释放资源 NET_DVR_CloseUpgradeHandle.....	71
	日志查找.....	71
5.19.7	查找设备的日志信息 NET_DVR_FindDVRLog_V30 .....	71
5.19.8	逐条获取查找到的日志信息 NET_DVR_FindNextLog_V30.....	72
5.19.9	释放查找日志的资源 NET_DVR_FindLogClose_V30.....	72
	恢复设备默认参数.....	72
5.19.10	恢复设备默认参数 NET_DVR_RestoreConfig.....	72
5.19.11	完全恢复出厂默认参数 NET_DVR_RemoteControl.....	72
	导入/导出配置文件 .....	73
5.19.12	导出配置文件 NET_DVR_GetConfigFile_V30.....	73
5.19.13	导出配置文件 NET_DVR_GetConfigFile .....	73
5.19.14	导入配置文件 NET_DVR_SetConfigFile_EX.....	73
5.19.15	导入配置文件 NET_DVR_SetConfigFile.....	74
	远程重启.....	74
5.19.16	重启设备 NET_DVR_RebootDVR .....	74
6	错误代码及说明.....	75
6.1	网络通讯库错误码.....	75
6.2	RTSP 通讯库错误码.....	81
6.3	软解码库错误码.....	81
7	附录:结构体.....	83

# 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、解码器、多屏控制器、报警主机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摄像机（抓拍机）相关的功能，设备型号如下：

(i)DS-2CD9361(-S)、(i)DS-2CD9282(-S)、DS-2CD9111(-S)、(i)DS-2CD9120(-H/-HS)、(i)DS-2CD9121(-S/-V/A/A-S)、DS-2CD9122(-S/-H/-HS)、(i)DS-2CD9131(-S/-IS/-KS)、(i)DS-2CD9151A(-S)、(i)DS-2CD9152(-H/-S/-HS/-HIS/-HKS)、(i)DS-2CD9182(-H/-HS)、DS-2CD986A、DS-2CD986C、DS-2CD977、DS-2CD976(-V)、DS-2CD966(-V)等(i)DS-2CD93xx、(i)DS-2CD92xx、(i)DS-2CD91xx、和 DS-2CD9xx 系列智能交通摄像机。

设备网络 SDK 主要功能：实时码流预览、摄像机参数配置、网络触发抓拍以及获取上传的抓拍图片等功能。

**注意：**该文档附录中结构体说明仅供参考，更多说明及更新请参见《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
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		HCIndustry.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 5.1.6.25 (build20160401)

- 更新原因：智能交通摄像机 V4.2.0
- 新增报警控制配置功能（对应接口：[NET\\_DVR\\_GetSTDConfig](#) 和 [NET\\_DVR\\_SetSTDConfig](#)）：  
配置类型：NET\_DVR\_GET\_ALARMCTRL\_CFG、NET\_DVR\_SET\_ALARMCTRL\_CFG。
- 新增报警控制能力集（对应接口：[NET\\_DVR\\_GetSTDAbility](#)）：  
配置类型：NET\_DVR\_GET\_ALARMCTRL\_CAPABILITIESG。
- 新增车辆子品牌文件上传功能（对应接口：[NET\\_DVR\\_UploadFile\\_V40](#)）：  
UPLOAD\_SUBBRAND\_FILE。
- [NET\\_ITS\\_PICTURE\\_INFO](#)(抓拍图片信息)使用 5 个保留字节新增参数：dwUTCTime(UTC 时间)、byCompatibleAbility(兼容能力字段)。
- [NET\\_ITS\\_PLATE\\_RESULT](#)(识别结果)使用 5 个保留字节新增参数：byIllegalFromatType(违章信息格式类型)、pIllegalInfoBuf(违法代码字符信息)。
- [NET\\_DVR\\_MANUALSNAP](#)(手动抓拍参数)使用 1 个保留字节新增参数：byLaneNo(车道号)。
- [ITS\\_OVERLAP\\_ITEM\\_TYPE](#)(字符叠加类型)新增枚举类型：OVERLAP\_ITEM\_CAR\_SUBBRAND(车辆子品牌)。
- [NET\\_DVR\\_SNAP\\_CAMERAPARAMCFG](#)(CCD 参数)使用 8 个保留字节新增参数：byMCEEnabled(记忆色增强使能)、byMCELevel(记忆色增强强度)、byAutoContrastEnabled(自动对比度使能)、byAutoContrastLevel(自动对比等级)、byLSEDetailEnabled(细节增强使能)、byLSEDetailLevel(细节增强等级)、byLPDEEnabled(车牌增强使能)、byLPDELevel(车牌增强等级)。
- [NET\\_ITC\\_ICR\\_AOTOSWITCH\\_PARAM](#)(ICR 自动切换参数)使用 1 个保留字节新增参数：byICRAutoSwitch(ICR 自动切换加阈值选项)。
- 触发参数配置扩展：
  - 1) [ITC\\_TRIGGERMODE\\_TYPE](#)(触发模式)新增枚举类型：ITC\_REDLIGHT\_PEDESTRIAN\_TYPE(行人闯红灯触发)、ITC\_NOCOMITY\_PEDESTRIAN\_TYPE(不礼让行人触发)。
  - 2) [NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION](#)(触发参数联合体)中新增：  
[NET\\_ITC\\_REDLIGHT\\_PEDESTRIAN\\_PARAM](#)(行人闯红灯参数)、[NET\\_ITC\\_NOCOMITY\\_PEDESTRIAN\\_PARAM](#)(不礼让行人参数)。
  - 3) [NET\\_ITC\\_PLATE\\_RECOG\\_PARAM](#)(牌识参数)中 dwRecogMode(识别的类型)新增取值：bit18-使馆车牌识别、bit19-车辆子品牌识别。
  - 4) [NET\\_ITC\\_SINGLE\\_IOSPEED\\_PARAM](#)(单组 IO 测速参数)、[NET\\_ITC\\_LANE\\_PARAM](#)(卡口 RS485 触发模式车道参数)、[NET\\_ITC\\_SINGLEIO\\_PARAM](#)(单 IO 参数)分别使用 1 个保留字节新增参数：byEmergencyCapEn(是否启用应急车道抓拍)。
- 视频电警触发参数扩展：
  - 1) [NET\\_ITC\\_VIOLATION\\_DETECT\\_PARAM](#)(违规检测参数)使用 1 个保留字节新增参数：byPostSnapTimes(卡口抓拍张数)。
  - 2) [NET\\_ITC\\_LANE\\_VIDEO\\_EPOLICE\\_PARAM](#)(车道参数)使用 20 个保留字节新增参数：struInterval(抓拍间隔参数)。
- 设备通用能力集(对应接口：[NET\\_DVR\\_GetDeviceAbility](#)，能力集类型：DEVICE\_ABILITY\_INFO)，其中智能交通设备能力(ITDeviceAbility)扩展：
  - 1) <ITSAbility>中<triggerMode>新增取值：postNoComityPed(不礼让行人触发)、postRedLightPed(行人闯红灯触发)。

2) <ITSAbility>中<ManualSnap>新增子节点: <laneNo>(车道号)。

- 新增错误码: 1605~1620。

### Version 5.1.3.25 (build20150916)

- 更新原因: 智能交通摄像机 V4.1.1
- [NET\\_DVR\\_SNAPCFG](#)(触发参数)使用 4 个保留字节新增参数: dwSnapVehicleNum(抓拍匹配序号)。

### Version 5.1.1.17 (build20150602)

- 更新原因: 智能交通摄像机 V4.1.0
- [NET\\_ITC\\_POST\\_IMT\\_PARAM](#)(智慧监控配置)使用 1 个保留字节新增参数: bySnapMode(抓拍类型)。
- [NET\\_ITC\\_EPOLICE\\_LANE\\_PARAM](#)(电警 RS485 车检器触发模式车道参数)使用 1 个保留字节新增参数: bySnapPicPreRecord(抓拍图片预录时间点)。
- [NET\\_ITC\\_POST\\_MPR\\_PARAM](#)(卡口多帧检测(MPR)触发参数)使用 1 个保留字节新增参数: byPicUploadType(图片上传类型)。
- [NET\\_DVR\\_PICCFG\\_V30](#) 和 [NET\\_DVR\\_PICCFG\\_V40](#)(通道图像参数)分别使用 1 个保留字节新增参数: byOSDMilliSecondEnable(视频叠加时间支持毫秒)。
- [NET\\_ITC\\_LANE\\_VIDEO\\_EPOLICE\\_PARAM](#)(视频电警触发模式车道参数)使用 1 个保留字节新增参数: bySnapPicPreRecord(抓拍图片预录时间点)。
- [NET\\_ITC\\_IOOUT\\_PARAM](#)(IO 输出配置参数)中 byIOWorkMode(IO 输出口工作模式)新增取值: 2- 常亮灯。
- [NET\\_DVR\\_SNAPENABLECFG](#)(使能参数)使用 1 个保留字节新增参数: bySnapPicResolution(抓拍图片分辨率)。
- [NET\\_DVR\\_SNAP\\_ABILITY](#)(智能交通摄像机能力集)的 bySupport2 新增取值: bySupport2&0x02, 表示是否支持时间毫秒视频叠加功能。
- 设备通用能力集(DEVICE\_ABILITY\_INFO), 其中智能交通设备能力(ITDeviceAbility)扩展, <ITCAbility>中 <imageOverlayString>新增子节点<itemTypeCustomLength>(自定义类型长度)。
- 设备图像参数能力集(DEVICE\_VIDEOPIC\_ABILITY)扩展, <OSD>新增子节点: <OSDMilliSecond>(频叠加时间支持毫秒)。
- 新增支持完全恢复出厂默认参数功能 (对应接口: [NET\\_DVR\\_RemoteControl](#)):  
命令: NET\_DVR\_COMPLETE\_RESTORE\_CTRL。

### Version 5.0.3.15 (build20150129)

- 更新原因: 智能交通摄像机 V4.0.0
- 新增字符叠加参数配置扩展 (对应接口: [NET\\_DVR\\_GetDeviceConfig](#) 和 [NET\\_DVR\\_SetDeviceConfig](#)):  
命令: NET\_ITS\_GET\_OVERLAP\_CFG\_V50、NET\_ITS\_SET\_OVERLAP\_CFG\_V50, 兼容  
NET\_ITS\_GET\_OVERLAP\_CFG、NET\_ITS\_SET\_OVERLAP\_CFG。
- 新增上传违法字典文件功能 (对应接口: [NET\\_DVR\\_UploadFile\\_V40](#))  
上传文件类型: UPLOAD\_VIOLATION\_FILE。
- 新增雷达报警上传 (对应接口: [NET\\_DVR\\_SetDVRMessageCallBack\\_V31](#)):  
COMM\_ITS\_RADARINFO。
- 触发参数配置扩展:
  - 1) [ITC\\_TRIGGERMODE\\_TYPE](#) (触发模式)新增枚举类型: ITC\_POST\_HVT\_TYPE\_V50(混行卡口视频触发模式)、ITC\_POST\_IMT\_TYPE(智慧监控配置)。
  - 2) [NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION](#)(触发参数联合体)中新增结构体参数:  
NET\_ITC\_POST\_HVT\_PARAM\_V50(混行卡口视频触发)、[NET\\_ITC\\_POST\\_IMT\\_PARAM](#)(智慧监控配置)。

- 3) [ITC\\_LANE\\_USEAGE\\_TYPE](#)(车道用途)新增枚举类型: ITC\_LANE\_BAN\_LEFT(禁左车道)、ITC\_LANE\_BAN\_RIGHT(禁右车道)。
- 4) [ITC\\_LANE\\_DIRECTION\\_TYPE](#)(车道方向)新增枚举类型: ITC\_LANE\_FORWARD(正向行驶)、ITC\_LANE\_BACKWARD(背向行驶)、ITC\_LANE\_BOTHWAY(双向行驶)、ITC\_LANE\_STRAIGHT\_WAIT\_RIGHT(直行待行+右转)。
- 5) [NET\\_ITC\\_LANE\\_LOGIC\\_PARAM](#)(车道属性)使用 1 个保留字节新增参数: byCarDriveDirect(车辆行驶方向)。
- 6) [NET\\_ITC\\_LANE\\_PARAM](#)(车道参数)、[NET\\_ITC\\_SINGLE\\_IOSPEED\\_PARAM](#)(单组 IO 测速参数)分别使用 4 个保留字节新增参数: byRelaLaneDirectionType(关联车道方向类型)、byLowSpeedLimit(小车限低速值)、byBigCarLowSpeedLimit(大车限低速值)、byLowSpeedCapEn(是否启用低速抓拍)。
- 7) [NET\\_ITC\\_VTLANE\\_PARAM](#)(虚拟线圈触发模式车道参数)、[NET\\_ITC\\_EPOLICE\\_LANE\\_PARAM](#)(电警 RS485 车检器触发模式车道参数)、[NET\\_ITC\\_LANE\\_MPR\\_PARAM](#)(多帧检测模式车道参数)、[NET\\_ITC\\_LANE\\_PRS\\_PARAM](#)(视频检测触发模式车道参数)分别使用 1 个保留字节新增参数: byRelaLaneDirectionType(关联车道方向类型)。
- 8) [NET\\_ITC\\_POST\\_RS485\\_PARAM](#)(卡口 RS485 车检器触发参数)中的 byTriggerSpareMode(触发备用模式)新增取值: 2- 卡口混合车道模式。
- [NET\\_ITS\\_IMGMERGE\\_CFG](#)(图片合成配置)中 dwTwoMergeType(两张图合成方式)、dwThreeMergeType(三张图合成方式)新增合成编码取值: 203~210、313~320。
- 车牌识别上传信息扩展:
  - 1) [NET\\_DVR\\_VEHICLE\\_INFO](#)(车辆信息)中 byIllegalType(违法类型)新增取值: 18-禁右、19-禁左、20-压黄线、21-未系安全带。
  - 2) [NET\\_ITS\\_PICTURE\\_INFO](#)(抓拍图片信息)使用 1 个保留字节新增参数: byPicRecogMode(图片背向识别)。
  - 3) [NET\\_ITS\\_PLATE\\_RESULT](#)(交通识别结果)使用 5 个保留字节新增参数: byRelaLaneDirectionType(关联车道方向类型)、dwCustomIllegalType(自定义违章类型)。
  - 4) [NET\\_DVR\\_PLATE\\_RESULT](#)(交通识别结果老报警信息类型)使用 1 个保留字节新增参数: byRelaLaneDirectionType(关联车道方向类型)。
- [NET\\_ITC\\_VIOLATION\\_DETECT\\_PARAM](#)(违规检测参数)使用 2 个保留字节新增参数: byTurnAroundEnable(违章掉头使能)、byThirdPlateRecogTime(第三张牌识时间)。
- [NET\\_DVR\\_CAMERAPARAMCFG\\_EX](#)(前端参数配置)使用 64 个保留字节新增参数: struSnapCCD(抓拍机 CCD 参数)。
- [NET\\_DVR\\_SETUPALARM\\_PARAM](#)(报警布防参数)中 byLevel(布防优先级)增加支持三级布防。
- [NET\\_DVR\\_SNAP\\_ABILITY](#)(智能交通摄像机能力集参数)使用 1 个保留字节新增参数: bySupportWhiteBalance(白平衡能力)。
- 设备通用能力集(DEVICE\_ABILITY\_INFO), 其中智能交通设备能力(ITDeviceAbility)扩展, <ITCAbility>中 <imageOverlayString>新增子节点<itemTypeCustomLength>(自定义类型长度)。

### Version 4.3.1.5 (build20141017)

- 更新原因: 智能交通摄像机 V3.8.2
- 新增交通统计信息上传报警类型 (对应接口: [NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#) 和 [NET\\_DVR\\_StartListen\\_V30](#)):  
COMM\_ALARM\_TPS\_REAL\_TIME、COMM\_ALARM\_TPS\_STATISTICS。
- 新增支持监听方式获取车牌识别结果 (对应接口: [NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#) 和 [NET\\_DVR\\_StartListen\\_V30](#)): COMM\_ITS\_PLATE\_RESULT。
- 新增支持码流附加信息配置 (对应接口: [NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)):

命令：NET\_DVR\_GET\_STREAM\_ATTACHINFO\_CFG、NET\_DVR\_SET\_STREAM\_ATTACHINFO\_CFG。

- 车牌识别信息上传扩展：
  - 1) [NET\\_DVR\\_SETUPALARM\\_PARAM](#)(报警布防参数)使用 2 个保留字节新增参数：bySupport(bit0 表示二级布防是否上传图片)。
  - 2) [NET\\_ITS\\_PLATE\\_RESULT](#)(识别结果)，使用 6 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)，参数 byVehicleType 新增取值：9-SUV/MPV、10-中型客车。
  - 3) [NET\\_ITS\\_PICTURE\\_INFO](#)(抓拍图片信息)中 byType 新增取值：6-人脸子图(主驾驶)、7-人脸子图(副驾驶)。
- 闪光灯状态检测功能：
  - 1) [NET\\_DVR\\_EXDEVDET\\_COND](#)(外设状态检测条件)中 byExternalDevType(外接设备类型)新增取值：2-闪光灯。
  - 2) [NET\\_DVR\\_ALARMINFO\\_V30](#)(报警信息)中 dwAlarmType(报警类型)新增取值：17-闪光灯异常。
- 视频电警触发模式扩展：
  - 1) [ITC\\_VIOLATION\\_DETECT\\_TYPE](#)(违规检测类型)新增枚举类型：ITC\_VIOLATION\_TURN\_AROUND(违法掉头)、ITC\_VIOLATION\_CONGESTION(拥堵)，ITC\_VIOLATION\_INTERSECTION\_PARK 类型含义从"路口停车"修改为"红灯越线"。
  - 2) [NET\\_ITC\\_PLATE\\_RECOG\\_PARAM](#)(牌识参数)中 dwRecogMode(识别的类型)新增取值：bit15-遮阳板检测、bit16-黄标车检测、bit17-危险品车辆检测。
- [ITS\\_OVERLAP\\_ITEM\\_TYPE](#)(字符叠加类型)新增枚举类型：OVERLAP\_ITEM\_YELLOW\_SIGN\_CAR(黄标车)、OVERLAP\_ITEM\_DANGEROUS\_CAR(危险品车)。
- [NET\\_DVR\\_SNAP\\_ABILITY](#)(智能交通摄像机能力集)扩展：
  - 1) bySupport1 新增取值：bySupport1&0x80 表示是否支持 tps 报警数据上传。
  - 2) 使用 1 个保留字节新增 bySupport2：bySupport2&0x01 表示是否支持码流附加信息配置。
- 设备通用能力集(DEVICE\_ABILITY\_INFO)，其中智能交通设备能力(ITDeviceAbility)扩展，<ITCAbility>新增子节点<FlashLampDet>。

### Version 4.3.1.3 (build20140912)

- 更新原因：DS-2CD9121A-S(B)牌识出入口
- 新增视频检测触发模式：
  - 1) [ITC\\_TRIGGERMODE\\_TYPE](#) 新增枚举类型：ITC\_POST\_PRS\_TYPE(视频检测触发配置)。
  - 2) [NET\\_ITC\\_PLATE\\_RECOG\\_PARAM](#)(牌识参数)中 dwRecogMode(识别的类型)新增取值：bit12-民航车牌识别、bit13-车牌过渡倾斜处理、bit14-超大车牌识别。
  - 3) [NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION](#)(触发参数联合体)新增结构体参数：NET\_ITC\_POST\_PRS\_PARAM(视频检测触发)。
- 设备通用能力集(DEVICE\_ABILITY\_INFO)，其中智能交通设备能力(ITDeviceAbility)扩展，<ITSAbility>中 <triggerMode>新增取值：postPRS。

### Version 4.3.0.6 (build20140722)

- 更新原因：智能交通摄像机 V3.8.0
- 新增 GPS 参数配置功能（对应接口：[NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)）：[NET\\_DVR\\_GPS\\_DATACFG](#)。
- 新增车辆允许名单信息同步功能（对应接口：[NET\\_DVR\\_StartRemoteConfig](#) 和 [NET\\_DVR\\_SendRemoteConfig](#)）：

命令：NET\_DVR\_VEHICLELIST\_CTRL\_START。

- [NET\\_DVR\\_CAMERA\\_SETUPCFG](#)(相机架设参数)使用 8 个保留参数新增：fHorFieldAngle(相机水平视场角角度)、fVerFieldAngle(相机垂直视场角角度)。
- [NET\\_ITS\\_PLATE\\_RESULT](#)(车牌识别结果)使用 2 个保留参数新增：byBarrierGateCtrlType(道闸控制类型)、byAlarmDataType(报警数据类型)。
- [NET\\_DVR\\_VEHICLE\\_INFO](#)(车辆信息)的参数 byIllegalType(违章类型)新增取值：15-违章停车、16-违章掉头、17-占用应急车道。
- [NET\\_ITC\\_IOOUT\\_PARAM](#)(IO 输出配置)使用 1 个保留字节新增参数：byIOWorkMode(IO 输出口工作模式)。
- [NET\\_ITS\\_IMGMERGE\\_CFG](#)(图片合成配置)扩展，使用 1 个保留字节新增参数：byCloseupProportion(特写图比例)，新增合成方式：309、310、311、312。
- [ITS\\_OVERLAP\\_ITEM\\_TYPE](#) (叠加信息类型)新增枚举类型：OVERLAP\_ITEM\_MONITOR\_ID(监测点编号)、OVERLAP\_ITEM\_SUN\_VISOR(遮阳板)、OVERLAP\_ITEM\_LANE\_DIRECTION(车道行驶方向)。
- [NET\\_DVR\\_ENTRANCE\\_CFG](#)(出入口控制参数)使用 1 个保留字节新增参数：byBarrierGateCtrlMode(道闸控制模式)。
- 触发参数配置扩展：
  - 1) [ITC\\_LANE\\_USEAGE\\_TYPE](#)(车道用途)新增枚举类型：ITC\_LANE\_MIX(混合车道)、ITC\_LANE\_EMERGENCY(应急车道)。
  - 2) [NET\\_ITC\\_SINGLEIO\\_PARAM](#)(单 IO 参数)、[NET\\_ITC\\_SINGLE\\_IOSPEED\\_PARAM](#)(单组 IO 测速参数)、[NET\\_ITC\\_LANE\\_PARAM](#)(RS485 触发模式车道参数)分别使用 1 个保留参数新增：byUseageType(车道用途类型)。
  - 3) [NET\\_ITC\\_POST\\_MPR\\_PARAM](#)(卡口多帧检测触发参数)使用 1 个保留参数新增：bySourceType(信号源类型)，[NET\\_ITC\\_LANE\\_MPR\\_PARAM](#)(多帧检测模式车道参数)使用 4 个字节新增联合体参数 uTssParamInfo。
  - 4) [ITC\\_VIOLATION\\_DETECT\\_TYPE](#)(违规检测类型)新增枚举类型：ITC\_VIOLATION\_HIGH\_SPEED(超速)、ITC\_VIOLATION\_LOW\_SPEED(低速)。
  - 5) [NET\\_ITC\\_VIOLATION\\_DETECT\\_PARAM](#)(违规检测参数)使用 1 个保留参数新增：bySpeedTimes(违法超速抓拍张数)。
  - 6) [NET\\_ITC\\_LANE\\_VIDEO\\_EPOLICE\\_PARAM](#)(视频电警触发的单个车道参数)使用 2 个保留参数新增：byCarSpeedLimit(车速限制值)、byCarSignSpeed(标志限速)。
- ICR 切换新增算法自动模式：
  - 1) [NET\\_ITC\\_ICRCFG](#)(ICR 参数配置)的参数 bySwitchType(切换模式)新增取值：4- 算法自动。
  - 2) [NET\\_ITC\\_ICR\\_PARAM\\_UNION](#)(ICR 切换配置)新增结构体参数：[NET\\_ITC\\_ICR\\_ALGAOTOSWITCH\\_PARAM](#)(算法自动)。
- [NET\\_DVR\\_SNAP\\_ABILITY](#)(智能交通摄像机能力集)的 bySupport1 新增取值：
  - bySupport1&0x20, 表示是否支持 GPS 模块参数配置；
  - bySupport1&0x40, 表示是否支持 NTP 校时间隔支持分钟设置。
- 设备通用能力集(DEVICE\_ABILITY\_INFO)，其中智能交通设备能力(ITDeviceAbility)扩展，<ITCAbility>中<，新增节点：<GPSDATA CFG>；ITCIOOutParam>新增子节点<ioWorkMode>；<CameraSetUpCfg>新增子节点：<horFieldAngle>、<verFieldAngle>；<ICRCFG>新增子节点：<switchType>、<AlgorithmAutoSwitch>；<ITCEntranceCfg>新增子节点<barrierGateCtrlMode>。
- 设备网络应用参数能力(DEVICE\_NETAPP\_ABILITY)扩展，<IPAddrFilter>中新增子节点<IPTest>。

## Version 4.2.8.5 (build20140317)

- 更新原因：出入口抓拍单元(一体机) DS-TCG113 V3.7
- 新增出入口禁止名单数据同步功能（对应接口：[NET\\_DVR\\_StartRemoteConfig](#)）：

- 命令：NET\_DVR\_VEHICLE\_CONTROL\_LIST\_START。
- 新增获取出入口禁止名单数据功能（对应接口：[NET\\_DVR\\_StartRemoteConfig](#)）：  
命令：NET\_DVR\_GET\_ALL\_VEHICLE\_CONTROL\_LIST。
  - 新增远程控制功能（对应接口：[NET\\_DVR\\_RemoteControl](#)）：  
命令：NET\_DVR\_VEHICLE\_DELINFO\_CTRL(删除设备禁止名单数据库信息)、  
NET\_DVR\_BARRIERGATE\_CTRL(远程控制道闸)、NET\_DVR\_GATELAMP\_CTRL(常亮灯功能)。
  - 新增出入口参数配置功能（对应接口：[NET\\_DVR\\_GetDeviceConfig](#) 和 [NET\\_DVR\\_SetDeviceConfig](#)）：  
命令：NET\_DVR\_GET\_ENTRANCE\_PARAMCFG、NET\_DVR\_SET\_ENTRANCE\_PARAMCFG。
  - 新增报警信息上传（对应接口：[NET\\_DVR\\_SetDVRMessageCallBack V30](#) 和 [NET\\_DVR\\_StartListen V30](#)）：  
NET\_DVR\_VEHICLE\_CONTROL\_LIST\_DSALARM(允许名单数据需要同步报警)、  
NET\_DVR\_VEHICLE\_CONTROL\_ALARM(允许名单车辆报警)、NET\_DVR\_FIRE\_ALARM(消防报警)。
  - [NET\\_DVR\\_SNAP\\_ABILITY](#)(智能交通摄像机能力集)的 bySupport1 新增取值：bySupport1&0x10 表示是否支持出入口配置。
  - 设备通用能力集(DEVICE\_ABILITY\_INFO)，其中智能交通设备能力(ITDeviceAbility)扩展，新增节点：  
<ITCGateNum>、<ITCEntranceCfg>。

### Version 4.2.8.5 (build20140317)

- 更新原因：智能交通摄像机 V3.7.0
- 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。
- 新增获取设备当前触发模式的功能（对应接口：[NET\\_DVR\\_GetDVRConfig](#)）：  
[NET\\_DVR\\_CURTRIGGERMODE](#)。
- 新增(MPR)多帧检测触发模式：联合体 [NET\\_ITC\\_TRIGGER\\_PARAM\\_UNION](#) 中新增  
NET\_ITC\_POST\_MPR\_PARAM，ITC\_TRIGGERMODE\_TYPE 新增枚举类型 ITC\_POST\_MPR\_TYPE。
- 新增获取信号灯状态功能（对应接口：[NET\\_DVR\\_GetDeviceStatus](#)）：  
命令：NET\_DVR\_GET\_SIGNALLAMP\_STATUS，状态结构体：[NET\\_DVR\\_EXDEVDET\\_CFG](#)
- 新增上传报警信息类型(对应接口：[NET\\_DVR\\_SetDVRMessageCallBack V30](#) 和 [NET\\_DVR\\_StartListen V30](#))：  
信号灯异常检测报警：COMM\_SIGNAL\_LAMP\_ABNORMAL，对应结构体：[NET\\_DVR\\_SIGNALLAMP\\_DETCFG](#)。
- [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：3-特写图、4-二值图、5-码流。
- [NET\\_DVR\\_VEHICLE\\_INFO](#)(车辆信息)中 byVehicleType(车辆类型)新增取值：4-二轮车触发、5-三轮车触发。
- [NET\\_ITS\\_PLATE\\_RESULT](#)(智能交通识别结果)中 byVehicleType(车辆类型)新增取值：6-行人、7-二轮车、8-三轮车。
- [NET\\_ITC\\_PLATE\\_RECOG\\_PARAM](#)(车牌识别参数)中 dwRecogMode(识别类型)新增取值：bit8-摩托车识别、bit9-场景模式，原“微小车牌”识别类型修正为：bit10-微小车牌。
- [NET\\_ITC\\_RADAR\\_PARAM](#)(雷达参数)中 byRadarType(雷达类型)新增取值：5-安道雷(无雷达控制器)。
- 卡口视频触发模式配置扩展：
  - 1) [NET\\_ITC\\_VTLANE\\_PARAM](#)(虚拟线圈触发模式车道参数)使用 2 个保留字节新增参数：  
byLowSpeedLimit(小车低速限速)、byBigCarLowSpeedLimit(大车低速限速)。
  - 2) [NET\\_ITC\\_VTCOIL\\_INFO](#)(虚拟线圈参数)使用 2 个保留字节新增参数：byUseageType(车道用途类型)、  
byCarDriveDirect(车辆行驶方向)。
  - 3) [NET\\_ITC\\_POST\\_VTCOIL\\_PARAM](#)(卡口虚拟线圈触发参数)使用 6 个保留字节新增参数：  
bySpeedDetector(测速方式)、dwVioDetectType(违规检测类型)、byDebugMode(调试模式)。
- 卡口 IO 测速模式配置扩展，[NET\\_ITC\\_SINGLE\\_IOSPEED\\_PARAM](#)(单组 IO 测速参数)使用 2 个保留字节新增

参数: byBigCarSignSpeed(大车标志限速)、byCarSignSpeed(小车标志限速)。

- 视频电警触发模式配置扩展:
  - 1) [NET\\_ITC\\_VIOLATION\\_DETECT\\_PARAM](#)(违规检测参数)使用 1 个保留字节新增参数: byDriveLineSnapSen(压线灵敏度)。
  - 2) [NET\\_ITC\\_LANE\\_VIDEO\\_EPOLICE\\_PARAM](#)(单个车道视频电警触发参数)使用 1 个保留字节新增参数: byEnableRadar(启用雷达测试)。
  - 3) [NET\\_ITC\\_SINGLE\\_VIDEO\\_DETECT\\_LIGHT\\_PARAM](#)(单组视频检测交通信号灯参数)使用 1 个保留字节新增参数: byYellowLightTime(黄灯持续时间)。
- OSD 字符叠加配置扩展:
  - 1) [NET\\_ITS\\_OVERLAPCFG\\_COND](#)(字符叠加配置条件参数)使用 1 个保留字节新增参数: byPicModeType(大小图 图片模式)。
  - 2) [ITS\\_OVERLAP\\_ITEM\\_TYPE](#) (叠加字符选项)新增类型: OVERLAP\_ITEM\_SEATBELT(26-有无系安全带)。
  - 3) [NET\\_ITS\\_OVERLAP\\_SINGLE\\_ITEM\\_PARAM](#)(单条字符叠加信息)使用 5 个保留字节新增参数: byEnablePos(是否启用坐标显示)、wStartPosTop(起始上坐标)、wStartPosLeft(起始左坐标)。
  - 4) [NET\\_ITS\\_OVERLAP\\_ITEM\\_PARAM](#)(字符叠加参数配置)使用 7 个保留字节新增参数: byParamFillZeroEnable(参数补零使能)、byPlateLeftCornerEnable(车牌小图叠加左上角使能)、wStartSPicPosTop(起始上坐标)、wStartSPicPosLeft(起始左坐标)、byOsdLocate(OSD 叠加位置)。
- 10. FTP 参数配置扩展:
  - 1) [NET\\_DVR\\_PICTURE\\_NAME](#)(图片命名规则)中 byItemOrder 新增取值类型: PICNAME\_ITEM\_CAR\_COLOR(车身颜色)、PICNAME\_ITEM\_PLATE\_COORDINATE(车牌坐标)、PICNAME\_ITEM\_CAR\_TYPE(车辆类型)、PICNAME\_ITEM\_VIOLATION\_TYPE(违规类型)、PICNAME\_ITEM\_CUSTOM(自定义)。
  - 2) [NET\\_ITC\\_FTP\\_CFG](#)(FTP 配置)使用 33 个保留字节新增参数: byUploadPlateEnable(车牌小图是否上传)、szPicNameCustom[32](自定义图片命名)。
- 11. 图片查找扩展:
  - 1) [NET\\_DVR\\_FIND\\_PICTURE\\_PARAM](#)(图片查找条件)使用 13 个保留字节新增参数: dwTrafficType(图片检索生效项)、dwVehicleType(车辆类型)、dwIllegalType(违规检测类型)、byLaneNo(车道号)。
  - 2) [NET\\_DVR\\_FIND\\_PICTURE](#)(图片查找结果)使用 2 个保留字节新增参数: byPlateColor(车牌颜色)、byVehicleLogo(汽车品牌)。
  - 3) VCA\_PLATE\_COLOR(车牌颜色)新增枚举类型: VCA\_GREEN\_PLATE(绿色车牌)、VCA\_BKAIR\_PLATE(民航黑色车牌)、VCA\_OTHER(其他)。
  - 4) VLR\_VEHICLE\_CLASS(汽车品牌)新增枚举类型: 15~62。
- [NET\\_DVR\\_COMPRESSION\\_INFO\\_V30](#)(码流压缩参数)中 byResolution(分辨率)新增取值: 60-2752\*2208。
- [NET\\_DVR\\_SNAP\\_ABILITY](#)(智能交通摄像机能力集), 使用 2 个保留字节新增参数: byRelayNum(支持继电器的个数)、bySupport1(设备能力扩展, 视频触发优化、DDNS 配置、获取当前触发模式等能力), 参数 bySupport 新增能力取值: bySupport&0x80 表示是否支持云存储功能。
- 设备通用能力集(DEVICE\_ABILITY\_INFO), 其中智能交通设备能力(ITDeviceAbility)扩展, 节点<ITCAbility>:
  - 1) 新增子节点<ParamFillZero>、<CloudStorage>、<SignalLampDet>、<CurTriggerMode>、<ManualSnap>、<supportMilliCheckTime>新增子节点<customSetTimeSignLen>; <ItcFtpCfg>新增子节点<supportUploadPlate>、<picNameCustomLen>。
  - 2) <triggerMode>新增取值: postMPR; <support>新增取值: CloudStorage; <overlapType>新增取值: seatBelt; <charSize>新增取值: 128\*128; <radarType>新增取值: AndaoleiWithoutCtrl; <itemOrder>新增取值: carColor,plateCoordinate,carType,violationType,custom。
- 新增错误号: 1414~1499、1600、1601



- 原有接口或者结构，V3.7 新增支持：
  - 1) 图片查找、下载：[NET\\_DVR\\_FindPicture](#)、[NET\\_DVR\\_FindNextPicture](#)、[NET\\_DVR\\_CloseFindPicture](#)、[NET\\_DVR\\_GetPicture\\_V30](#)、[NET\\_DVR\\_GetPicture](#)。
  - 2) DDNS 网络参数配置，对应接口：[NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)，对应命令：NET\_DVR\_GET\_DDNSCFG\_V30、NET\_DVR\_SET\_DDNSCFG\_V30。
  - 3) ROI 区域检测，对应接口：[NET\\_DVR\\_GetDeviceConfig](#) 和 [NET\\_DVR\\_SetDeviceConfig](#)，对应命令：NET\_DVR\_GET\_ROI\_DETECT\_NUM、NET\_DVR\_GET\_ROI\_DETECT、NET\_DVR\_SET\_ROI\_DETECT。
  - 4) 多码流压缩参数配置(支持三码流)，对应接口：[NET\\_DVR\\_GetDeviceConfig](#) 和 [NET\\_DVR\\_SetDeviceConfig](#)，对应命令：NET\_DVR\_GET\_MULTI\_STREAM\_COMPRESSIONCFG、NET\_DVR\_SET\_MULTI\_STREAM\_COMPRESSIONCFG。

### Version 4.2.7.3 (build20131209)

- 更新原因：智能交通摄像机 V3.6.0
- 新增参数配置功能（对应接口：[NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)）：  
[NET\\_ITC\\_ICRCFG](#)(ICR 切换配置)、[NET\\_ITC\\_EXCEPTION](#)(异常参数配置)
- 新增参数批量配置功能（对应接口：[NET\\_DVR\\_GetDeviceConfig](#) 和 [NET\\_DVR\\_SetDeviceConfig](#)）：  
RS485 关联接入设备配置：NET\_ITC\_GET\_RS485\_ACCESSINFO、NET\_ITC\_SET\_RS485\_ACCESSINFO  
FTP 信息配置(扩展支持四级目录)：NET\_ITC\_GET\_FTPCFG、NET\_ITC\_SET\_FTPCFG
- [NET\\_DVR\\_SNAP\\_ABILITY](#) 使用 7 个保留字节新增参数：wSupportMultiRadar(RS485 口关联雷达能力)、byICRPresetNum(支持的 ICR 预置点数)、byICRTimeSlot(支持的 ICR 的时间段数)、bySupportRS485Num(支持的 RS485 口数)、byExpandRs485SupportSensor(RS485 口关联车检器能力)、byExpandRs485SupportSignalLampDet(RS485 口关联信号灯检测器能力)，原有参数 bySupport 新增取值定义：bySupport&0x4、bySupport&0x8、bySupport&0x10、bySupport&0x20、bySupport&0x40。
- [NET\\_ITC\\_POST\\_SINGLEIO\\_PARAM](#) 参数，struSingleIO[MAX\_IOIN\_NUM]扩展为 struSingleIO[MAX\_IOIN\_NUMEX]，byRes[32]扩展为 byRes[360]。
- [NET\\_ITC\\_SINGLEIO\\_PARAM](#)、[NET\\_ITC\\_SINGLE\\_IOSPEED\\_PARAM](#)、[NET\\_ITC\\_LANE\\_PARAM](#)、[NET\\_ITC\\_VTLANE\\_PARAM](#)、[NET\\_ITC\\_SINGLE\\_IOTL\\_PARAM](#)、[NET\\_ITC\\_EPOLICE\\_LANE\\_PARAM](#) 分别使用 1 个保留字节新增参数：byRelatedIOOutEx(关联的 IO 输出口扩展)。
- [NET\\_DVR\\_CAMERA\\_SETUPCFG](#) 分别使用 3 个保留字节新增参数：wVideoDetCoefficient(视频测试系数)、byErectMethod(架设方式)。
- 设备通用能力集(DEVICE\_ABILITY\_INFO)，其中智能交通设备能力(ITDeviceAbility)，节点<ITCAbility>新增子节点：<support>、<supportMultiRadar>、<ICRPresetNum>、<ICRTimeSlot>、<expandRs485SupportSensor>、<expandRs485SupportSignalLampDet>、<videoDetCoefficient>、<erectMethod>、<ExceptionAlarmITC>、<ICRCFG>、<ITCFTP>。
- 设备软硬件能力集(DEVICE\_SOFTHARDWARE\_ABILITY)新增节点：<RS485>。
- 新增错误号：1413
- 原有接口或者结构，V3.6 新增支持：
  - 1) 多网卡相关配置，对应接口：[NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)，对应命令：NET\_DVR\_GET\_NETCFG\_MULTI、NET\_DVR\_SET\_NETCFG\_MULTI、NET\_DVR\_GET\_NETWORK\_BONDING、NET\_DVR\_SET\_NETWORK\_BONDING
  - 2) 硬盘满、读写硬盘出错等异常报警信息上传，对应接口：[NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#) 和 [NET\\_DVR\\_StartListen\\_V30](#)，对应报警信息类型：COMM\_ALARM\_V30
  - 3) 雨刷控制，对应接口：[NET\\_DVR\\_PTZControl\\_Other](#)，对应命令：WIPER\_PWRON

### Version 4.2.5.6 (build20130808)

- 更新原因：智能交通摄像机 V3.5.0
- 新增远程配置功能（对应接口：[NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)）：  
[NET\\_DVR\\_CABINET\\_CFG](#)(机柜参数配置)、[NET\\_DVR\\_CALIBRATE\\_TIME](#)(时间校时扩展)、  
[NET\\_DVR\\_CAMERAPARAMCFG\\_EX](#)(前端参数配置扩展)。
- 新增报警布防上传信息类型（对应接口：[NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#)）：[NET\\_ITS\\_ROADINFO](#)(路口设备异常)
- 新增交通灯状态检测：[NET\\_ITC\\_TRAFFIC\\_LIGHT\\_COLOR](#)(常量矫正参数)、[NET\\_ITC\\_TRAFFIC\\_LIGHT\\_TURN](#)(常量矫正参数)
- [NET\\_DVR\\_STATUS\\_DETECTCFG](#) 使用 1 个保留字节新增参数：`byEnableTrafficLightDetect`(开启交通灯状态检测)。
- [NET\\_ITC\\_RADAR\\_PARAM](#) 使用 10 个保留字节新增参数：`wRadarSpeedValidTime`(雷达速度有效时间)、`fLineCorrectParam`(线性矫正参数)、`iConstCorrectParam`(常量矫正参数)，`byRadarType` 新增取值：4- 雷达接 IO 扩展盒。
- [NET\\_ITC\\_VTLANE\\_PARAM](#) 使用 16 个保留字节新增参数：`struLine`(车道线)。
- [NET\\_ITC\\_POST\\_VTCOIL\\_PARAM](#) 使用 17 个保留字节新增参数：`bySnapMode`(抓拍模式)、`struLine`(右车道线)。
- [NET\\_ITC\\_EPOLICE\\_LANE\\_PARAM](#) 使用 2 个保留字节新增参数：`byRedTrafficLightChan`(红灯通道号)、`byYellowTrafficLightChan`(黄灯通道号)。
- [NET\\_ITC\\_EPOLICE\\_RS485\\_PARAM](#) 使用 1 个保留字节新增参数：`byTrafficLightSignalSrc`(交通灯信号来源)。
- [NET\\_ITC\\_VIOLATION\\_DETECT\\_PARAM](#) 使用 4 个保留字节新增参数：`byChangeLaneTimes`(违法变道抓拍张数)、`bybanTimes`(违法禁令抓拍张数)、`wSnapPosFixPixel`(抓拍位置最小偏移)，参数 `dwVioDetectType` 新增违规检测类型：`ITC_VIOLATION_INTERSECTION_PARK`、`ITC_VIOLATION_GREEN_PARK`。
- [NET\\_ITC\\_PLATE\\_RECOG\\_PARAM](#) 使用 1 个保留字节新增参数：`byVehicleLogoRecog`(汽车品牌识别)，`dwRecogMode` 新增识别类型：`bit8`-微小车牌。
- [NET\\_DVR\\_FTPCFG](#) 使用 1 个保留字节新增参数：`byEnableAnony`(启用匿名)，`wTopDirMode` 和 `wSubDirMode` 新增取值：`0x6`- 自定义、`0x7`- 违规类型、`0x8`- 方向、`0x9`- 地点。
- [NET\\_DVR\\_SNAPENABLECFG](#) 使用 2 个保留字节新增参数：`wJpegPicSize`(JPEG 图片大小)。
- [NET\\_ITC\\_IOOUT\\_PARAM](#) 使用 1 个保留字节新增参数：`byAutoPlateBrightness`(车牌亮度自动使能闪光灯)。
- [NET\\_DVR\\_VEHICLE\\_INFO](#) 使用 1 个保留字节新增参数：`byVehicleLogoRecog`(汽车品牌)，`byVehicleType` 新增取值：3- 行人触发、6- 机动车触发，`byColor` 新增取值：`0xff`- 未进行车身颜色识别。
- [NET\\_ITS\\_PLATE\\_RESULT](#) 使用 16 个保留字节新增参数：`struSnapFirstPicTime`(端点时间)、`dwIllegalTime`(违法持续时间)。
- [NET\\_DVR\\_CAMERA\\_SETUPCFG](#) 使用 5 个保留字节新增参数：`bySetupHeightUnit`(架设高度单位)、`dwSceneDis`(图像下边缘位置与相机之间的水平距离)。
- [NET\\_DVR\\_PLCCFG](#) 使用 2 个保留字节新增参数：`wPlcBrightOffSet`(车牌亮度补偿灵敏度)。
- `NET_DVR_PICTURE_NAME` 的参数 `byItemOrder` 新增元素：`PICNAME_ITEM_CAR_NUMBER`(车辆序号)。
- [NET\\_DVR\\_SNAP\\_ABILITY](#) 的参数 `bySupport` 新增取值：`bySupport&0x2` 表示是否支持扩展的校时配置结构。
- 前端参数能力集 `IPC_FRONT_PARAMETER_V20` 新增能力节点：`<CorridorMode>`、`<BrightCompensate>`、`<ExposureSegment>`；智能交通能力集 `DEVICE_ABILITY_INFO` 新增新增能力节点：`<supportMilliCheckTime>`、`<SnapStatusDetectCFG>`、`<SnapRadarParam>`、`<SnapEnableCfg>`、`<ITCIOoutParam>`、`<CameraSetUpCfg>`、`<PlcCfg>`、`<CabinetAlarmParamCfg>`。

- 智能交通摄像机 V3.4.0
- [NET\\_ITC\\_LANE\\_PARAM](#)、[NET\\_ITC\\_VTCOIL\\_INFO](#) 使用 1 个保留字节新增参数 byLaneType(车道类型)
- [NET\\_ITC\\_SINGLE\\_IOSPEED\\_PARAM](#) 使用 2 个保留字节新增参数: byLaneType(车道类型)、byBigCarSpeedLimit(大车限速)。
- [NET\\_ITS\\_PLATE\\_RESULT](#)、[NET\\_DVR\\_PLATE\\_RESULT](#) 使用 1 个保留字节新增参数 byVehicleType(车辆类型)。
- [NET\\_ITC\\_VIDEO\\_EPOLICE\\_PARAM](#) 使用 1 个保留字节新增参数 byLogicJudge(闯红灯违规判断逻辑)。
- [NET\\_ITC\\_SINGLE\\_RS485\\_LIGHT\\_PARAM](#) 使用 1 个保留字节新增参数 byRelatedYLightChan(黄灯关联的红绿灯检测器通道号)。
- [NET\\_ITC\\_VIOLATION\\_DETECT\\_PARAM](#) 使用 1 个保留字节新增参数 byNonDriveSnapTimes(机占非抓拍张数)
- [NET\\_ITC\\_VTLANE\\_PARAM](#) 使用 2 个保留字节新增参数: byBigCarSignSpeed(大车标志限速)、byBigCarSpeedLimit(大车限速值)。
- [NET\\_ITC\\_EPOLICE\\_LANE\\_PARAM](#) 使用 2 个保留字节新增参数: byBigCarSignSpeed(大车标志限速值)、byBigCarSpeedLimit(大车限速值)。
- [ITC\\_VIOLATION\\_DETECT\\_TYPE](#) 新增枚举类型: ITC\_VIOLATION\_BAN(违法禁令)。
- [NET\\_DVR\\_VEHICLE\\_INFO](#) 中参数 byIllegalType 新增取值类型: 10-机动车违反规定使用专用车道, 11-黄牌车禁限, “4-闯红灯” 类型更改为违反交通灯指示。

### Version 4.2.1.7 (2013-1-11)

- 智能交通摄像机 V3.3.0
- 新增布防上传的报警信息类型 (对应接口: [NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#)): [NET\\_ITS\\_TRAFFIC\\_COLLECT](#)。
- 新增远程配置功能 (对应接口: [NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)): 交通统计报警参数([NET\\_DVR\\_TPS\\_ALARMCFG](#))、相机架设参数([NET\\_DVR\\_CAMERA\\_SETUPCFG](#))、图片合成配置参数([NET\\_ITS\\_IMGMERGE\\_CFG](#))。
- 新增批量配置功能 (对应接口: [NET\\_DVR\\_GetDeviceConfig](#) 和 [NET\\_DVR\\_SetDeviceConfig](#)): 命令: NET\_ITS\_GET\_OVERLAP\_CFG、NET\_ITS\_SET\_OVERLAP\_CFG。
- 新增获取智能交通摄像机能力集 (对应接口: [NET\\_DVR\\_GetDeviceAbility](#)): 前端参数能力 (IPC\_FRONT\_PARAMETER\_V20)、智能抓拍设备能力 (DEVICE\_ABILITY\_INFO)。
- 新增报警录像下载功能: [NET\\_DVR\\_StartDownload](#)。
- [NET\\_DVR\\_PLATE\\_RESULT](#) 使用 2 个保留字节新增参数: wAlarmRecordID(报警录像 ID), 参数 byResultType 增加类型取值: 2-长录像(支持查询)。
- [ITC\\_VIOLATION\\_DETECT\\_TYPE](#) 增加机占非抓拍和违法变道的枚举类型: ITC\_VIOLATION\_NONDRIVEWAY、ITC\_VIOLATION\_CHANGELANE。
- [NET\\_ITC\\_VIOLATION\\_DETECT\\_PARAM](#) 使用 6 个保留字节增加参数: byDriveLineSnapTimes(压车道线抓拍张数)、byReverseSnapTimes(逆行抓拍张数)、wStayTime(机占非停留时间)和 wStopLineDis(电警第 2 张违规图片与停止线的最短距离)。
- [ITC\\_LANE\\_USEAGE\\_TYPE](#) 增加禁止货车车道的枚举类型: ITC\_LANE\_BAN\_TRUCKS。
- [NET\\_ITC\\_LIGHT\\_ACCESSPARAM\\_UNION](#) 新增视频检测信号灯参数: [NET\\_ITC\\_VIDEO\\_DETECT\\_LIGHT\\_PARAM](#)。
- [NET\\_ITC\\_TRAFFIC\\_LIGHT\\_PARAM](#) 的参数 bySource 增加类型取值: 2-视频检测。
- [NET\\_ITC\\_LANE\\_VIDEO\\_EPOLICE\\_PARAM](#) 使用 6 个保留字节增加参数: bySensitivity(线圈灵敏度)和 byRecordEnable(闯红灯周期录像标志)、byRecordType(闯红灯录像类型)、byPreRecordTime(闯红灯录像预录时间)、byRecordDelayTime(闯红灯录像延时时间)、byRecordTimeOut(闯红灯录像超时时间)。
- [ITC\\_SERIAL\\_PROTOCOL](#) 为闯红灯违规的逻辑抓拍类型枚举结构, 增加类型:

ITC\_DOUBLE\_COIL\_PROTOCOL\_MODE4、ITC\_DOUBLE\_COIL\_PROTOCOL\_MODE5、ITC\_DOUBLE\_COIL\_PROTOCOL\_MODE6 和 ITC\_SINGLE\_COIL\_PROTOCOL1。

- [NET ITC SERIAL INFO](#) 使用 3 个保留字节增加参数：byNormalPassProtocol(正常过车抓拍协议)、byInverseProtocol(逆行抓拍协议)和 bySpeedProtocol(超速抓拍协议)。
- [NET DVR VIDEOEFFECT](#) 使用 3 个保留字节新增参数：byEnableFunc(防过曝、低照度或强光抑制使能)、byLightInhibitLevel(强光抑制等级)和 byGrayLevel(灰度值域)。
- [NET DVR WHITEBALANCE](#) 的参数 byWhiteBalanceMode 新增类型取值：3-锁定白平衡，4-白炽灯，5-暖光灯，6-自然光，7-日光灯。
- [NET DVR DAYNIGHT](#) 使用 5 个保留字节新增参数(定时模式的起止时间以及报警输入触发状态)：byBeginTimeMin、byBeginTimeSec、byEndTimeMin、byEndTimeSec 和 byAlarmTrigState。
- [NET ITC LANE PARAM](#) 使用 3 个保留字节新增参数：byOverlayDriveWay(OSD 叠加的车道号)、byCartSignSpeed(大车标志限速)和 byCartSpeedLimit(大车限速值)。
- [NET ITC EPOLICE LANE PARAM](#) 使用 2 个保留字节新增参数：byOverlayDriveWay(OSD 叠加的车道号)、bySerialType(车检器类型)。
- [NET ITC POST RS485 PARAM](#) 使用 2 个保留字节新增参数：byTriggerSpareMode(触发备用模式)、byFaultToleranceTime(容错时间)。
- [NET ITC IOOUT PARAM](#) 使用 1 个保留字节新增参数：byMode(闪光灯工作方式)。
- [NET DVR VEHICLE INFO](#) 的参数 byIllegalType 增加违法类型取值：7-路口滞留，8-机占非，9-违法变道。
- [NET DVR SINGLE HD](#) 使用 12 个保留字节新增参数：dwStorageType(存储类型)、dwPictureCapacity(硬盘图片容量)和 dwFreePictureSpace(剩余图片空间)。
- [NET DVR SNAP ABILITY](#) 使用 1 个保留字节新增能力参数：bySupport。
- 新增智能交通摄像机错误码：1400~1499。

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

- 智能交通摄像机 V3.2.2
- 新增布防上传的报警信息类型（对应接口 [NET DVR SetDVRMessageCallBack V30](#)）：[NET ITS PLATE RESULT](#)
- [NET DVR DEVICEINFO V30](#) 参数 bySupport1 新增是否支持多磁盘数、rtsp over http 以及车牌新报警信息的能力，并且使用 2 个保留字节增加参数 byMultiStream、byMultiStreamProto 表示多码流能力。
- [NET DVR SETUPALARM PARAM](#) 使用 1 个保留字节增加参数 byAlarmInfoType，支持选择智能交通摄像机上传的报警信息类型。
- [NET DVR VEHICLE INFO](#) 使用 16 个保留字节增加数组参数 byCustomInfo[16]，上传自定义信息，并且 byIllegalType 增加违法类型：5-压车道线、6-不按导向。

### Version 4.1.8 (2012-9-28)

- 新增智能交通摄像机设备型号：iDS-2CD9121(-S)、iDS-2CD9152(-S)、DS-2CD9121(-S)、DS-2CD9122(-S)、DS-2CD9151A(-S)、DS-2CD9152(-S)。
- 新增 v3.1 及之后版本设备视频电警触发参数配置功能：[NET ITC VIDEO TRIGGER PARAM](#)，对应接口：[NET DVR GetDeviceConfig](#)、[NET DVR SetDeviceConfig](#)。

### Version 4.1.4 (2012-4-20)

- 能力集 [NET DVR SNAP ABILITY](#) 增加能力参数，用于区分新老设备的功能差异（比如触发模式）：byPreTriggerSupport、dwAbilityType、byIoSpeedGroup、byIoLightGroup、byRecogRegionType，对应接口：[NET DVR GetDeviceAbility](#)。

- 新增获取智能 IPC 设备 (v3.1 及之后版本) 的能力集功能: [NET\\_ITC\\_TRIGGERMODE\\_ABILITY](#), 对应接口: [NET\\_DVR\\_GetDeviceAbility](#)。
- 新增 V3.1 及以后版本设备支持的配置功能 (对应接口: [NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)): IO 输出配置 ([NET\\_ITC\\_IOOUT\\_PARAM](#))、触发参数配置 ([NET\\_ITC\\_TRIGGERCFG](#))、获取触发模式推荐参数 ([NET\\_ITC\\_SINGLE\\_TRIGGERCFG](#))、获取状态检测使能参数 ([NET\\_DVR\\_STATUS\\_DETECTCFG](#))。
- 新增状态检测上传功能 (对应接口: [NET\\_DVR\\_StartListen\\_V30](#)): [NET\\_ITC\\_STATUS\\_DETECT\\_RESULT](#)
- 新增布防(扩展)接口: [NET\\_DVR\\_SetupAlarmChan\\_V41](#), 支持设置布防优先级。
- 抓拍机使能配置 ([NET\\_DVR\\_SNAPENABLECFG](#)) 中增加是否上传抓拍附加信息到 FTP (byUploadInfoFTP) 和是否自动格式化 SD 卡 (byAutoFormatSD) 配置参数。
- 设备状态结构 ([NET\\_DVR\\_DEVICESTATECFG](#)) 中增加 SD 卡状态 (bySDStatus) 和布防等级 (byFortifyLevel)。
- 增加日志类型 (对应接口: [NET\\_DVR\\_FindDVRLog\\_V30](#)): MINOR\_SET\_TRIGGERMODE\_CFG、MINOR\_GET\_TRIGGERMODE\_CFG、MINOR\_SET\_IOOUT\_CFG、MINOR\_GET\_IOOUT\_CFG、MINOR\_GET\_TRIGGERMODE\_DEFAULT、MINOR\_GET\_ITCSTATUS、MINOR\_SET\_STATUS\_DETECT\_CFG、MINOR\_GET\_STATUS\_DETECT\_CFG、MINOR\_GET\_VIDEO\_TRIGGERMODE\_CFG、MINOR\_SET\_VIDEO\_TRIGGERMODE\_CFG。

### Version 4.0.3 (2011-3-21)

- 新增智能交通摄像机设备型号: DS-2CD966(-V)、DS-2CD986C、DS-2CD986A。
- 新增配置功能 (对应接口: [NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)): 车牌亮度补偿配置功能 ([NET\\_DVR\\_PLCCFG](#))、卡口电警参数配置功能 ([NET\\_DVR\\_POSTEPOLICECFG](#))。
- 闪光灯时控配置功能 ([NET\\_DVR\\_FLASH\\_OUTCFG](#)) 中新增时间段: byStartHour、byStartMinute、byEndHour、byEndMinute、byFlashLightEnable。
- 获取设备状态 ([NET\\_DVR\\_DEVICESTATECFG](#)) 增加获取获取车检器状态 (byDetectorLinkState)。

### Version 4.0.2 (2010-12-30)

- SDK 新增智能交通摄像机设备型号: DS-2CD966(-V)、DS-2CD986C、DS-2CD986A。
- 新增配置功能 (对应接口: [NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)): 车牌亮度补偿配置功能 ([NET\\_DVR\\_PLCCFG](#))、卡口电警参数配置功能 ([NET\\_DVR\\_POSTEPOLICECFG](#))。
- 闪光灯时控配置功能 ([NET\\_DVR\\_FLASH\\_OUTCFG](#)) 中新增时间段: byStartHour、byStartMinute、byEndHour、byEndMinute、byFlashLightEnable。
- 获取设备状态 ([NET\\_DVR\\_DEVICESTATECFG](#)) 增加获取获取车检器状态 (byDetectorLinkState)。

### Version 4.0.1 (2010-9-8)

- 新增配置功能 (对应接口: [NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)): 车牌识别配置 ([NET\\_DVR\\_SPRCFG](#))、获取设备状态信息 ([NET\\_DVR\\_DEVICESTATECFG](#))。
- 使能配置功能 ([NET\\_DVR\\_SNAPENABLECFG](#)) 新增配置信号灯同步和图像翻转 (byFrameFlip、wFlipAngle、byLightSyncPower)。
- 手动触发连拍 ([NET\\_DVR\\_ContinuousShoot](#))。

### Version 4.0.0 (2010-3-9)

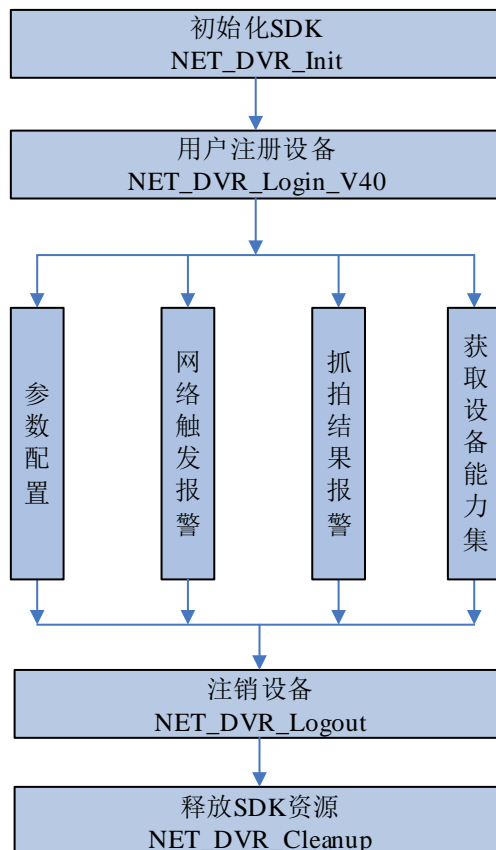
- SDK 新增支持智能交通摄像机 (抓拍机), 设备型号: DS-2CD976、DS-977。
- 新增配置功能 (对应接口: [NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)): 前端参数配置功能 ([NET\\_DVR\\_CAMERAPARAMCFG](#))、IO 输入配置功能 ([NET\\_DVR\\_IO\\_INCFG](#))、IO 输出配置功能

- ([NET\\_DVR\\_IO\\_OUTCFG](#))、闪光灯配置功能 ([NET\\_DVR\\_FLASH\\_OUTCFG](#))、红绿灯配置功能 ([NET\\_DVR\\_LIGHTSNAPCFG](#))、测速配置功能 ([NET\\_DVR\\_MEASURESPEEDCFG](#))、图像叠加配置功能 ([NET\\_DVR\\_IMAGEOVERLAYCFG](#))、单 IO 触发配置功能 ([NET\\_DVR\\_SNAPCFG](#))、虚拟线圈配置功能 ([NET\\_DVR\\_VTPARAM](#))、使能配置功能 ([NET\\_DVR\\_SNAPENABLECFG](#))。
- 新增手动触发抓拍 ([NET\\_DVR\\_ManualSnap](#))。
  - 新增能力集类型: [SNAPCAMERA\\_ABILITY](#), 对应接口: [NET\\_DVR\\_GetDeviceAbility](#)。
  - 新增智能交通摄像机报警上传功能 ([NET\\_DVR\\_PLATE\\_RESULT](#)), 对应接口: [NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#)。

## 3 函数调用顺序

### 3.1 SDK 接口调用主要流程

图 3.1 SDK 调用主要流程



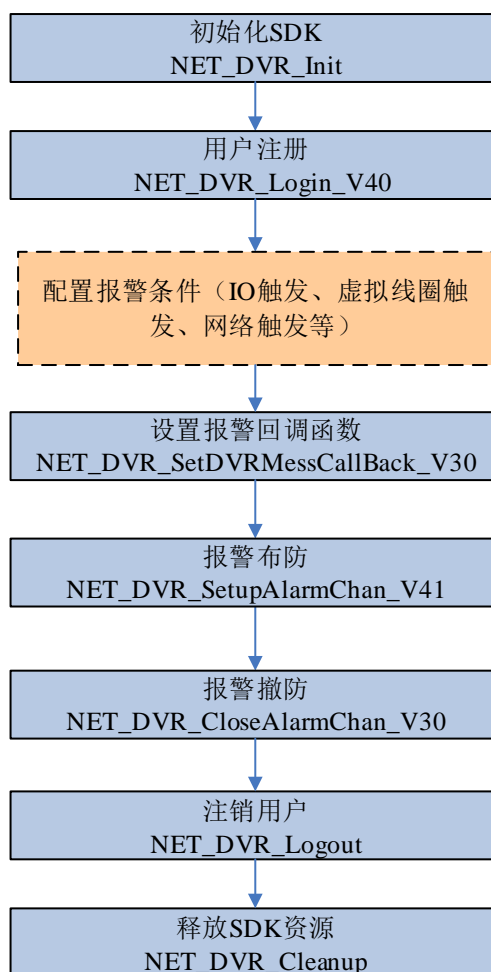
智能交通摄像机功能模块包括参数配置、抓拍结果报警和获取设备能力集等功能。该模块的所有功能都需要先进行用户注册，[NET\\_DVR\\_Login\\_V40](#) 接口返回的用户 ID 号作为其他功能接口的参数。

- **抓拍结果报警：**智能交通摄像机抓拍（IO 触发、视频触发、网络触发等）通过“布防”报警的方式上传，在报警回调函数中获取抓拍结果，包括抓拍图片、车牌信息等。详见 [3.2 报警（布防）流程](#)。
- **参数配置：**主要完成对智能交通摄像机的基本参数的配置功能，调用的接口为 [NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)。如果已经通过 IE 或者配套客户端软件完成配置，不需要调用这些接口再次配置。
- **网络触发报警：**通过网络触发设备抓拍，调用的接口为 [NET\\_DVR\\_ContinuousShoot](#)。
- **获取设备能力集：**通过 [NET\\_DVR\\_GetDeviceAbility](#) 接口获取智能交通摄像机能力信息。

[调用实例代码](#)

## 3.2 报警（布防）流程

图 3.2 报警布防流程



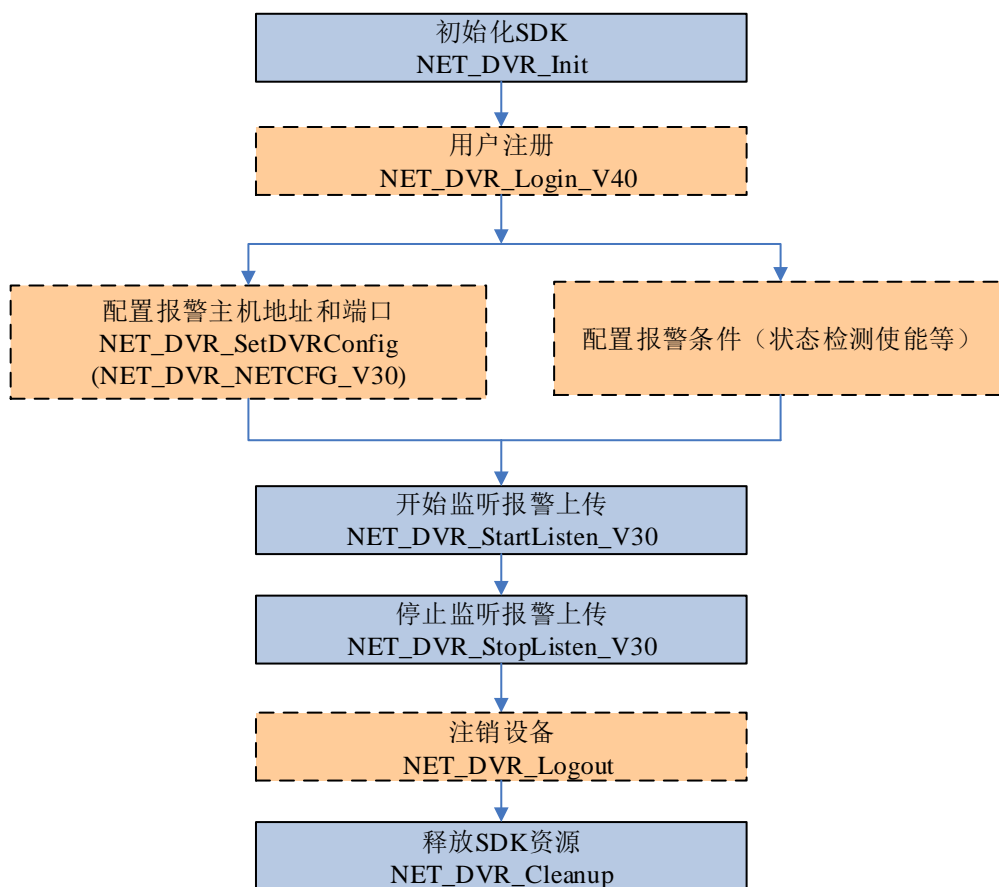
- “布防”报警方式：是指 SDK 主动连接设备，并发起报警上传命令，设备发生报警立即发送给 SDK。
- “布防”方式需要先进行用户注册（[NET\\_DVR\\_Login\\_V40](#)）。虚线框部分是实现报警信息上传的必要条件，主要完成相关的报警条件和处理方法的配置，参数配置的接口为 [NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)，或者是网络触发连拍（[NET\\_DVR\\_ContinuousShoot](#)）。IO 触发、虚拟线圈触发、红绿灯触发等模式，如果参数事先已经配置完成，那么虚线框部分可以省略。接下来就是设置报警回调函数（[NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#)），调用成功后还需要设置布防（[NET\\_DVR\\_SetupAlarmChan\\_V41](#)）。
- [NET\\_DVR\\_CloseAlarmChan\\_V30](#) 撤销报警上传通道，设备将不再上传报警信息。
- 抓拍机的抓拍结果报警信息（`COMM_UPLOAD_PLATE_RESULT`）只能通过布防方式获取，交通统计数据（`COMM_ITS_TRAFFIC_COLLECT`）只能通过布防而且是一级布防才能获取。

[调用实例代码](#)



### 3.3 报警（监听）流程

图 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。
- 这个过程需要远程配置设备的报警主机地址（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，但事先没有配置，就必须调用参数配置接口（[NET\\_DVR\\_GetDVRConfig](#) 和 [NET\\_DVR\\_SetDVRConfig](#)）对设备的网络参数（[NET\\_DVR\\_NETCFG\\_V30](#)）进行配置。而虚线框“配置报警条件和处理方法”部分与“布防”中的一致。对以上需要配置的参数都设置完后，调用 [NET\\_DVR\\_StartListen\\_V30](#) 函数，开启 SDK 的监听端口，准备接收设备上传的报警信息。
- 该方式适用于多个设备向一台客户端上传报警，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；缺点是设备只支持一个报警主机地址和端口号的配置。

[调用实例代码](#)

## 4 函数调用实例

### 4.1 参数配置模块的示例代码

[相关模块流程图](#)

#### 配置 IO 触发参数 (NET\_DVR\_SNAPCFG)

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

void main() {
    //-----
    // 初始化
    NET_DVR_Init();
    //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
    LONG IUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    IUserID = NET_DVR_Login_V30("192.0.0.64", 8000, "admin", "12345", &struDeviceInfo);
    if (IUserID < 0)
    {
        printf("Login error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }
    int iRet;
    //获取 IO 触发参数
    DWORD dwReturnLen;
    NET_DVR_SNAPCFG struParams = {0};
    iRet = NET_DVR_GetDVRConfig(IUserID, NET_DVR_GET_SNAPCFG, struDeviceInfo.byStartChan, \
        &struParams, sizeof(NET_DVR_SNAPCFG), &dwReturnLen);

    if (!iRet)
    {
        printf("NET_DVR_GetDVRConfig NET_DVR_GET_SNAPCFG error.\n");
        NET_DVR_Logout(IUserID);
    }
}
```

```

NET_DVR_Cleanup();
return;
}

//设置 IO 触发参数
struParams.bySnapTimes = 2;
struParams.wSnapWaitTime = 1000;
struParams.wIntervalTime[0] = 1000;

iRet = NET_DVR_SetDVRConfig(IUserID, NET_DVR_SET_SNAPCFG, struDeviceInfo.byStartChan, \
    &struParams, sizeof(NET_DVR_SNAPCFG));
if (!iRet)
{
    printf("NET_DVR_GetDVRConfig NET_DVR_SET_SNAPCFG error.\n");
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}
printf("Video Bitrate is %d\n", struParams.struNormHighRecordPara.dwVideoBitrate);
//注销用户
NET_DVR_Logout(IUserID);
//释放 SDK 资源
NET_DVR_Cleanup();
return;
}

```

## 4.2 抓拍图片上传示例代码

[相关模块流程图](#)

布防报警的示例代码:

```

#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。

    switch (ICommand)
    {
        case COMM_UPLOAD_PLATE_RESULT:
        {
            NET_DVR_PLATE_RESULT struPlateResult={0};
            memcpy(&struPlateResult, pAlarmInfo, sizeof(struPlateResult));
            printf("车牌号: %s\n", struPlateResult.struPlateInfo.sLicense);//车牌号

            switch(struPlateResult.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;
            }
            //场景图
            if (struPlateResult.dwPicLen != 0 && struPlateResult.byResultType == 1 )
            {
                sprintf(filename,"testpic_%d.jpg",iNum);
                fSnapPic=fopen(filename,"wb");
                fwrite(struPlateResult.pBuffer1,struPlateResult.dwPicLen,1,fSnapPic);
                iNum++;
                fclose(fSnapPic);
            }
            //车牌图

```

```

if (struPlateResult.dwPicPlateLen != 0 && struPlateResult.byResultType == 1)
{
    sprintf(filename, "testPicPlate_%d.jpg", iNum);
    fSnapPicPlate=fopen(filename, "wb");
    fwrite(struPlateResult.pBuffer1, struPlateResult.dwPicLen, 1, fSnapPicPlate);
    iNum++;
    fclose(fSnapPicPlate);
}
//其他信息处理.....
break;
}
case COMM_ITS_PLATE_RESULT:
{
    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
        if ((struITSPlateResult.struPicInfo[i].dwDataLen != 0) && (struITSPlateResult.struPicInfo[i].byType ==
1) | (struITSPlateResult.struPicInfo[i].byType == 2))
        {
            sprintf(filename, "testITSpic%d_%d.jpg", iNum, i);
            fSnapPic=fopen(filename, "wb");
            fwrite(struITSPlateResult.struPicInfo[i].pBuffer, struITSPlateResult.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;
}
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
    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;
    }
    //设置报警回调函数
    NET_DVR_SetDVRMessageCallBack_V30(MSesGCallback, NULL);
    //启用布防
    NET_DVR_SETUPALARM_PARAM struSetupParam={0};
    struSetupParam.dwSize=sizeof(NET_DVR_SETUPALARM_PARAM);

    //上传报警信息类型: 0- 老报警信息(NET_DVR_PLATE_RESULT), 1- 新报警信息(NET_ITS_PLATE_RESULT)
    struSetupParam.byAlarmInfoType=1;

    LONG IHandle = NET_DVR_SetupAlarmChan_V41(IUserID,&struSetupParam);
    if (IHandle < 0)
    {
        printf("NET_DVR_SetupAlarmChan_V41 error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Logout(IUserID);
        NET_DVR_Cleanup();
        return;
    }
    //-----
    //启动预览
    LONG IRealPlayHandle;
    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, 仅取流不解码时可为空(设置回调函数)
    struPlayInfo.lChannel = 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
    struPlayInfo.bBlocked = 1; //0- 非阻塞取流, 1- 阻塞取流

    IRealPlayHandle = NET_DVR_RealPlay_V40(IUserID, &struPlayInfo, NULL, NULL);
    if (IRealPlayHandle < 0) //预览失败
    {
        printf("NET_DVR_RealPlay_V40 error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Logout(IUserID);
        NET_DVR_Cleanup();
        return;
    }
    Sleep(5000);
    //关闭预览
    NET_DVR_StopRealPlay(IRealPlayHandle);
    printf("\n");
    printf("\n");
    printf("\n");

```

```
////////////////////////////////////
//网络触发抓拍
NET_DVR_SNAPCFG struSnapCfg;
memset(&struSnapCfg, 0, sizeof(NET_DVR_SNAPCFG));

//结构体大小
struSnapCfg.dwSize = sizeof(NET_DVR_SNAPCFG);

//线圈抓拍次数, 0-不抓拍, 非 0-连拍次数, 目前最大 5 次
struSnapCfg.bySnapTimes = 3;

//抓拍等待时间, 单位 ms, 取值范围[0,60000]
struSnapCfg.wSnapWaitTime = 1000;

//连拍间隔时间, 单位 ms, 取值范围[67,60000]
struSnapCfg.wIntervalTime[0] = 1000;
struSnapCfg.wIntervalTime[1] = 1000;

//触发 IO 关联的车道号, 取值范围[0,9]
struSnapCfg.byRelatedDriveWay = 0;

//网络触发连拍
if (!NET_DVR_ContinuousShoot(IUserID, &struSnapCfg))
{
    printf("NET_DVR_ContinuousShoot error, %d\n", NET_DVR_GetLastError());
    return;
}
//-----
//退出
Sleep(20000);
//撤销布防上传通道
if (!NET_DVR_CloseAlarmChan_V30(IHandle))
{
    printf("NET_DVR_CloseAlarmChan_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;
}
```



## 4.3 抓拍机实时状态检测结果上传

[相关模块流程图](#)

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

void CALLBACK MessageCallback(LONG ICommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD dwBufLen, void *pUser)
{
    int i;
    NET_ITC_STATUS_DETECT_RESULT struAlarmInfo; //状态监测结果结构体
    memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_ITC_STATUS_DETECT_RESULT));

    switch(ICommand)
    {
    case COMM_ITC_STATUS_DETECT_RESULT:
        {
            switch (struAlarmInfo.dwStatusType)
            {
            case 1: //IO 触发
                for (i=0; i<8; i++) //define MAX_IOIN_NUM 8 //最大 IO 输入口个数
                {
                    if (struAlarmInfo.uStatusParam.struTrigIO.byTriggerIOIndex[i] == 1)
                    {
                        printf("IO 触发, 触发 IO 号: %d\n", i+1);
                    }
                }
                break;
            default:
                break;
            }
        }
        break;
    default:
        break;
    }
}

void main() {
    //-----
    // 初始化
    NET_DVR_Init();
}
```

```
//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
LONG IUserID;
NET_DVR_DEVICEINFO_V30 struDeviceInfo;
IUserID = NET_DVR_Login_V30("172.0.0.100", 8000, "admin", "12345", &struDeviceInfo);
if (IUserID < 0)
{
    printf("Login error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}
//启用监听
LONG IHandle;
IHandle = NET_DVR_StartListen_V30(NULL, 7200, MessageCallback, NULL);
if (IHandle < 0)
{
    printf("NET_DVR_StartListen_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}
Sleep(5000);
//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, 参数默认值为 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 状态信息结构, 请参见结构体: [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 pSDKAbI)

参 数: [out] pSDKAbI 功能信息, 请参见结构体: [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_RECONNECTSUCESS	0x8008	透明通道重连成功
EXCEPTION_PLAYBACK	0x8010	回放异常
EXCEPTION_DISKFMT	0x8011	硬盘格式化
PREVIEW_RECONNECTSUCESS	0x8015	预览时重连成功
ALARM_RECONNECTSUCESS	0x8016	报警时重连成功
RESUME_EXCHANGE	0x8017	用户交互恢复
NETWORK_FLOWTEST_EXCEPTION	0x8018	网络流量检测异常

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_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，默认：8000
[in]lpActivateCfg	激活参数，包括激活使用的初始密码，详见结构体： <a href="#">NET_DVR_ACTIVATECFG</a>

**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:** 建议使用此接口实现注销功能。

[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\_V40 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, 详见表 5.5  
[in] dwInLength 输入缓冲区的长度  
[out] pOutBuf 输出缓冲区指针, 详见表 5.5  
[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_ENCODE_ALL_ABILITY_V20	0x008	设备所有编码能力
IPC_FRONT_PARAMETER_V20	0x009	设备前端参数
DEVICE_USER_ABILITY	0x00c	设备用户管理参数能力
DEVICE_NETAPP_ABILITY	0x00d	设备网络应用参数能力
DEVICE_VIDEOPIC_ABILITY	0x00e	设备图像参数能力
DEVICE_SERIAL_ABILITY	0x010	设备 RS232 和 RS485 串口能力
DEVICE_ABILITY_INFO	0x011	智能交通摄像机和智能终端设备能力
SNAPCAMERA_ABILITY	0x300	智能交通摄像机抓拍能力
ITC_TRIGGER_MODE_ABILITY	0x301	智能交通摄像机（V3.1 及以后版本）的触发模式能力

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 plnBuf 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	plnBuf	pOutBuf
DEVICE_SOFTWARE_ABILITY	获取设备软硬件能力	无	设备软硬件能力 XML 描述
DEVICE_ENCODE_ALL_ABILITY_V20	获取设备所有编码能力	编码能力获取输入 XML 描述	设备所有编码能力 XML 描述
IPC_FRONT_PARAMETER_V20	获取前端参数能力	前端参数能力输入 XML 描述	前端参数能力 XML 描述
DEVICE_USER_ABILITY	获取用户管理参数能力	用户管理参数能力输入 XML 描述	设备用户管理参数能力 XML 描述
DEVICE_NETAPP_ABILITY	获取网络应用参数能力	网络应用参数能力输入 XML 描述	设备网络应用参数能力 XML 描述
DEVICE_VIDEOPIC_ABILITY	获取图像参数能力	图像参数能力输入 XML 描述	设备图像参数能力 XML 描述
DEVICE_SERIAL_ABILITY	获取 RS232 和 RS485 串口能力	串口能力获取输入 XML 描述	设备串口能力 XML 描述
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+	获取安全认证配置能力集	安全认证配置能力 XML 描述 (SecurityAbility)
		获取智能交通摄像机输入 XML 描述	智能交通摄像机和智能终端的能力(ITDeviceAbility)
SNAPCAMERA_ABILITY	获取智能交通摄像机的抓拍能力	无	<a href="#">NET_DVR_SNAP_ABILITY</a>
ITC_TRIGGER_MODE_ABILITY	获取智能交通摄像机（V3.1 及以后版本）触发模式能力	无	<a href="#">NET_ITC_TRIGGERMODE_ABILITY</a>

注：能力集 XML 描述详细内容请参见《设备网络 SDK 使用手册.chm》。

[返回目录](#)

## 5.4.2 获取设备能力集 **NET\_DVR\_GetSTDAbility**

函数: BOOL NET\_DVR\_GetSTDAbility(LONG IUserID, DWORD dwAbilityType, LPNET\_DVR\_STD\_ABILITY lpAbilityParam)

参数: [in]IUserID 用户 ID 号, NET\_DVR\_Login\_V40 的返回值  
 [in]dwAbilityType 能力类型, 具体定义见表 5.6  
 [in&out]lpAbilityParam 设备能力集参数(包括输入和输出参数), 不同的能力集对应不同的输入输出参数

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 5.6 获取设备能力集

宏定义	宏定义值	含义	lpCondBuffer	lpOutBuffer
NET_DVR_GET_ALARMCTRL_CAPABILITIES	3146	获取报警控制能力集	NULL	<a href="#">AlarmCtrlParam</a>

[返回目录](#)

## 5.5 实时预览

### 5.5.1 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\_V40 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\\_CLIENTINFO](#))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](#)

### 5.5.2 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\_V40 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\\_CLIENTINFO](#))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](#)

### 5.5.3 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\_V40 的返回值  
 [in] lpPreviewInfo 预览参数, 包括码流类型、取流协议、通道号、预览窗口句柄等, 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

```
typedef void(CALLBACK *REALDATACALLBACK) (LONG IRealHandle, DWORD dwDataType, BYTE *pBuffer, DWORD dwBufSize, void *pUser);
```

[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]pUser 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)
NET_DVR_AUDIOSTREAMDATA	3	音频数据

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:
- 该接口预览参数结构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。
  - 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。
  - fRealDataCallBack\_V30 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。
  - 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](#)

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

函数: LONG 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](#)

### 5.5.5 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 预览时播放声音控制

### 5.6.1 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](#)

### 5.6.2 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](#)

### 5.6.3 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](#)

### 5.6.4 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](#)

### 5.6.5 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](#)

### 5.6.6 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](#)

## 5.7 实时预览数据捕获

### 5.7.1 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 5.8  
 [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 5.8 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）
NET_DVR_AUDIOSTREAMDATA	3	音频数据

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。
- `cbRealDataCallBack` 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](#)

## 5.7.2 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

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](#)

## 5.7.3 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](#)

## 5.8 预览时抓图

### 5.8.1 设置抓图模式 `NET_DVR_SetCapturePictureMode`

函数: `BOOL NET_DVR_SetCapturePictureMode(DWORD dwCaptureMode)`

参数: `[in]dwCaptureMode` 抓图模式

```
enum tagPDC_PARAM_KEY{
    BMP_MODE    = 0, // BMP 模式
    JPEG_MODE   = 1  // JPEG 模式
}CAPTURE_MODE
```

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: 调用该接口设置抓图模式后, `NET_DVR_CapturePicture` 可抓取相应的图片。

[返回目录](#)

### 5.8.2 预览时, 单帧数据捕获图片并保存成文件 `NET_DVR_CapturePicture`

函数: `BOOL NET_DVR_CapturePicture(LONG IRealHandle, char *sPicFileName)`

参数: `[in] IRealHandle` `NET_DVR_RealPlay_V40` 的返回值  
`[in] sPicFileName` 保存图象的文件路径, 包含文件名。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 (包括文件名在内)。

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:

- 在调用该接口之前可以调用 [NET\\_DVR\\_SetCapturePictureMode](#) 设置抓图模式, 默认为 `BMP` 模式。如果抓图模式为 `BMP` 模式, 抓取的是 `BMP` 图片, 保存路径后缀应为 `.bmp`; 如果抓图模式为 `JPEG` 模式, 抓取的是 `JPEG` 图片, 保存路径后缀应为 `.jpg`。
- 调用该接口进行抓图, 实际是播放库解码抓图, 要求在调用 [NET\\_DVR\\_RealPlay\\_V40](#) 等接口时传入非空的播放句柄 (播放库解码显示), 否则时接口会返回失败 (调用次序错误)。
- 智能交通摄像机车牌抓拍请参考 [3.2 报警 \(布防\) 模块流程](#)。

[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 注册报警回调函数 `NET_DVR_SetDVRMessageCallBack_V31`

函数: `BOOL NET_DVR_SetDVRMessageCallBack_V31(MSGCallBack_V31 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_V31)(LONG ICommand, NET_DVR_ALARMER *pAlarmer,
char *pAlarmInfo, DWORD dwBufLen, void *pUser)
```

[out]ICommand	上传的消息类型, 详见表 5.9
[out]pAlarmer	报警设备信息, 详见 <a href="#">NET_DVR_ALARMER</a>
[out]pAlarmInfo	报警信息, 详见表 5.10
[out]dwBufLen	报警信息缓存大小
[out]pUser	用户数据

表 5.9 布防报警信息类型

宏定义	宏定义值	含义
COMM_UPLOAD_PLATE_RESULT	0x2800	交通抓拍结果上传(老报警信息类型)
COMM_ITS_PLATE_RESULT	0x3050	交通抓拍结果上传(新报警信息类型)
COMM_ITS_TRAFFIC_COLLECT	0x3051	交通统计数据上传
COMM_VEHICLE_CONTROL_LIST_DSALARM	0x3058	允许名单数据需要同步报警上传
COMM_VEHICLE_CONTROL_ALARM	0x3059	允许名单车辆报警上传
COMM_FIRE_ALARM	0x3060	消防报警上传
COMM_ITS_RADARINFO	0x3079	雷达报警上传
COMM_SIGNAL_LAMP_ABNORMAL	0x3080	信号灯异常检测上传
COMM_ALARM_TPS_REAL_TIME	0x3081	TPS 实时过车数据上传
COMM_ALARM_TPS_STATISTICS	0x3082	TPS 统计过车数据上传
COMM_ALARM_V30	0x4000	设备普通报警信息上传
COMM_ITS_ROAD_EXCEPTION	0x4500	路口设备异常报警上传

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 5.10 所示。

表 5.10 报警信息结构

消息类型 (ICommand)	上传内容	pAlarmInfo 对应的结构体
COMM_UPLOAD_PLATE_RESULT	交通抓拍结果(老报警信息类型)	<a href="#">NET_DVR_PLATE_RESULT</a>
COMM_ITS_PLATE_RESULT	交通抓拍结果(新报警信息类型)	<a href="#">NET_ITS_PLATE_RESULT</a>
COMM_ITS_TRAFFIC_COLLECT	交通统计数据上传	<a href="#">NET_ITS_TRAFFIC_COLLECT</a>
COMM_VEHICLE_CONTROL_LIST_DSALARM	允许名单数据需要同步报警上传	<a href="#">NET_DVR_VEHICLE_CONTROL_LIST_DSALARM</a>
COMM_VEHICLE_CONTROL_ALARM	允许名单车辆报警上传	<a href="#">NET_DVR_VEHICLE_CONTROL_ALARM</a>
COMM_FIRE_ALARM	消防报警上传	<a href="#">NET_DVR_FIRE_ALARM</a>
COMM_ITS_RADARINFO	雷达报警上传	<a href="#">NET_DVR_ALARM_RADARINFO</a>
COMM_SIGNAL_LAMP_ABNORMAL	信号灯异常检测上传	<a href="#">NET_DVR_SIGNALLAMP_DETCFG</a>
COMM_ALARM_TPS_REAL_TIME	TPS 实时过车数据上传	<a href="#">NET_DVR_TPS_REAL_TIME_INFO</a>
COMM_ALARM_TPS_STATISTICS	TPS 统计过车数据上传	<a href="#">NET_DVR_TPS_STATISTICS_INFO</a>
COMM_ALARM_V30	设备普通报警信息	<a href="#">NET_DVR_ALARMINFO_V30</a>
COMM_ITS_ROAD_EXCEPTION	路口设备异常报警信息	<a href="#">NET_ITS_ROADINFO</a>

- COMM\_UPLOAD\_PLATE\_RESULT 和 COMM\_ITS\_PLATE\_RESULT 分别对应接口 [NET\\_DVR\\_SetupAlarmChan\\_V41](#) 中布防参数 byAlarmInfoType=0 和 byAlarmInfoType=1。设备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 & 0x80（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。
- COMM\_UPLOAD\_PLATE\_RESULT：兼容旧型号(976/986/966 等)抓拍机，一次回调只上传一张图片信息
- COMM\_ITS\_PLATE\_RESULT：应用于违章图片组一次性上传以及合成图片上传等新功能。
- COMM\_ITS\_TRAFFIC\_COLLECT：支持收集交通调度信息的设备版本，用于接收交通调度统计信息。

[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                    NET\_DVR\_Login\_V40 的返回值  
[in]lpSetupParam                报警布防参数，请参见结构体：[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](#)）才能获得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 = NULL)
- 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                报警信息回调函数  
[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 5.11
[out] pAlarmer	报警设备信息，详见 <a href="#">NET_DVR_ALARMER</a>
[out] pAlarmInfo	报警信息，详见表 5.12
[out] dwBufLen	报警信息缓存大小
[out] pUser	用户数据

表 5.11 监听报警信息类型

宏定义	宏定义值	含义
COMM_ITC_STATUS_DETECT_RESULT	0x2810	实时状态检测结果上传
COMM_ITS_PLATE_RESULT	0x3050	交通抓拍结果上传(新报警信息类型)
COMM_VEHICLE_CONTROL_LIST_DSALARM	0x3058	允许名单数据需要同步报警上传
COMM_VEHICLE_CONTROL_ALARM	0x3059	允许名单车辆报警上传
COMM_FIRE_ALARM	0x3060	消防报警上传
COMM_SIGNAL_LAMP_ABNORMAL	0x3080	信号灯异常检测上传
COMM_ALARM_TPS_REAL_TIME	0x3081	TPS 实时过车数据上传
COMM_ALARM_TPS_STATISTICS	0x3082	TPS 统计过车数据上传
COMM_ALARM_V30	0x4000	设备普通报警信息上传
COMM_ITS_ROAD_EXCEPTION	0x4500	路口设备异常报警上传

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 5.12 所示。

- SDK 最大能支持 [512](#) 路监听。
- 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报警主机地址(struAlarmHostIpAddr)设置成 PC 机的 IP 地址（与接口中的 sLocalIP 参数一致），远程报警主机端口号(wAlarmHostIpPort)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，其他回调函数将接收不到报警信息。**

表 5.12 报警信息结构

消息类型（ICommand）	上传内容	pAlarmInfo 对应的结构体
COMM_ITC_STATUS_DETECT_RESULT	实时状态检测结果	<a href="#">NET_ITC_STATUS_DETECT_RESULT</a>
COMM_ITS_PLATE_RESULT	交通抓拍结果(新报警信息类型)	<a href="#">NET_ITS_PLATE_RESULT</a>
COMM_VEHICLE_CONTROL_LIST_DSALARM	允许名单数据需要同步报警上传	<a href="#">NET_DVR_VEHICLE_CONTROL_LIST_DSALARM</a>
COMM_VEHICLE_CONTROL_ALARM	允许名单车辆报警上传	<a href="#">NET_DVR_VEHICLE_CONTROL_ALARM</a>
COMM_FIRE_ALARM	消防报警上传	<a href="#">NET_DVR_FIRE_ALARM</a>
COMM_SIGNAL_LAMP_ABNORMAL	信号灯异常检测上传	<a href="#">NET_DVR_SIGNALLAMP_DETCFG</a>
COMM_ALARM_TPS_REAL_TIME	TPS 实时过车数据上传	<a href="#">NET_DVR_TPS_REAL_TIME_INFO</a>
COMM_ALARM_TPS_STATISTICS	TPS 统计过车数据上传	<a href="#">NET_DVR_TPS_STATISTICS_INFO</a>

COMM_ALARM_V30	设备普通报警信息	<a href="#">NET_DVR_ALARMINFO_V30</a>
COMM_ITS_ROAD_EXCEPTION	路口设备异常报警信息	<a href="#">NET_ITS_ROADINFO</a>

[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

### 通用参数配置

#### 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\_V40 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

[out] lpOutBuffer 接收数据的缓冲指针，详见表 5.13

[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 5.13 所示。

对于智能交通摄像机，设备只有一个通道，通道号即为 1。

表 5.13 通用参数获取

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_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_RS232CFG_V30	获取 RS232 串口参数	无效	<a href="#">NET_DVR_RS232CFG_V30</a>	1036
NET_DVR_GET_TIMECFG	获取时间参数	无效	<a href="#">NET_DVR_TIME</a>	118
NET_DVR_GET_USERCFG_V30	获取用户参数	无效	<a href="#">NET_DVR_USER_V30</a>	1006
NET_DVR_GET_SECURITY_CFG	获取安全认证配置	无效	<a href="#">NET_DVR_SECURITY_CFG</a>	147

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_NETCFG_MULTI	获取多网卡配置参数	无效	<a href="#">NET_DVR_NETCFG_MULTI</a>	1161
NET_DVR_GET_NETWORK_BONDING	获取 BONDING 网卡参数	无效	<a href="#">NET_DVR_NETWORK_BONDING</a>	1254
NET_DVR_GET_NTPCFG	获取网络应用参数(NTP)	无效	<a href="#">NET_DVR_NTTPARA</a>	224
NET_DVR_GET_DDNSCFG_V30	获取网络应用参数(DDNS)	无效	<a href="#">NET_DVR_DDNSPARA_V30</a>	1010
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_V40	获取录像计划参数	通道号	<a href="#">NET_DVR_RECORD_V40</a>	1008
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_NFSCFG	获取 NFS (网络文件系统) 参数	无效	<a href="#">NET_DVR_NFSCFG</a>	230
NET_DVR_GET_ZONEANDDST	获取时区和夏时制参数	无效	<a href="#">NET_DVR_ZONEANDDST</a>	128
NET_DVR_GET_VIDEOOUTCFG_V30	获取视频输出参数	无效	<a href="#">NET_DVR_VIDEOOUT_V30</a>	1028
NET_DVR_GET_STREAM_ATTACHINFO_CFG	获取码流附加信息配置	通道号	<a href="#">NET_DVR_STREAM_ATTACHINFO_CFG</a>	5042
NET_DVR_GET_HDCFG	获取硬盘管理参数	无效	<a href="#">NET_DVR_HDCFG</a>	1054
NET_DVR_GET_DISK_QUOTA_CFG	获取磁盘配额信息	通道号	<a href="#">NET_DVR_DISK_QUOTA_CFG</a>	1278

[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\_V40 的返回值
- [in] dwCommand 设备配置命令, 详见表 5.14
- [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 5.14
- [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 5.14 所示。

表 5.14 通用参数设置

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_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_RS232CFG_V30	设置 232 串口参数	无效	<a href="#">NET_DVR_RS232CFG_V30</a>	1037
NET_DVR_SET_TIMECFG	设置时间参数	无效	<a href="#">NET_DVR_TIME</a>	119
NET_DVR_SET_USERCFG_V30	设置用户参数	无效	<a href="#">NET_DVR_USER_V30</a>	1007

NET_DVR_SET_SECURITY_CFG	设置安全认证配置	无效	<a href="#">NET_DVR_SECURITY_CFG</a>	148
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_NETCFG_MULTI	设置多网卡配置参数	无效	<a href="#">NET_DVR_NETCFG_MULTI</a>	1162
NET_DVR_SET_NETWORK_BONDING	设置 BONDING 网卡参数	无效	<a href="#">NET_DVR_NETWORK_BONDING</a>	1255
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_DDNSCFG_V30	设置网络应用参数(DDNS)	无效	<a href="#">NET_DVR_DDNSPARA_V30</a>	1011
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_V40	设置录像计划参数	通道号	<a href="#">NET_DVR_RECORD_V40</a>	1009
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_NFSCFG	设置 NFS(网络文件系统)参数	无效	<a href="#">NET_DVR_NFSCFG</a>	231
NET_DVR_SET_ZONEANDDST	设置时区和夏时制参数	无效	<a href="#">NET_DVR_ZONEANDDST</a>	129
NET_DVR_SET_VIDEOOUTCFG_V30	设置视频输出参数	无效	<a href="#">NET_DVR_VIDEOOUT_V30</a>	1029
NET_DVR_SET_STREAM_ATTACHINFO_CFG	设置码流附加信息配置	通道号	<a href="#">NET_DVR_STREAM_ATTACHINFO_CFG</a>	5043
NET_DVR_SET_HDCFG	设置硬盘管理参数	无效	<a href="#">NET_DVR_HDCFG</a>	1055
NET_DVR_SET_DISK_QUOTA_CFG	设置磁盘配额信息	通道号	<a href="#">NET_DVR_DISK_QUOTA_CFG</a>	1279

[返回目录](#)

### 5.11.3 批量获取配置信息 **NET\_DVR\_GetDeviceConfig**

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

**参 数：**

- [in] lUserID                    用户 ID 号，NET\_DVR\_Login\_V40 的返回值
- [in] dwCommand                设备配置命令，详见表 5.15
- [in] dwCount                  一次要获取配置参数的个数，0 和 1 都表示 1 个信息，2 表示 2 个信息，最大 64 个
- [in] lpInBuffer                配置条件缓冲区，详见表 5.16
- [in] dwInBufferSize          缓冲区长度
- [out] lpStatusList            错误信息列表，和要查询的配置一一对应，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- 成功，大于 0-失败
- [out] lpOutBuffer             设备返回的参数内容（详见表 5.16），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

**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



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 指定需要获取的信息，lpOutBuffer 保存获取得到的 dwCount 个配置信息。不同的 dwCommand 对应不同的结构体和命令号，如表 5.16 所示。

表 5.15 参数批量获取命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_MULTI_STREAM_COMPRESSIONCFG	远程获取多码流压缩参数	3216
NET_DVR_GET_ROI_DETECT_NUM	获取 ROI 检测区域编号数目	3349
NET_DVR_GET_ROI_DETECT	获取 ROI 检测区域配置	3350

表 5.16 批量获取设备参数

dwCommand	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_GET_MULTI_STREAM_COMPRESSIONCFG	dwCount 个 <a href="#">NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND</a>	dwCount 个 <a href="#">NET_DVR_MULTI_STREAM_COMPRESSIONCFG</a>
NET_DVR_GET_ROI_DETECT_NUM	dwCount 个 <a href="#">NET_DVR_MUL_STREAM_CHANNEL_GROUP</a>	dwCount 个 <a href="#">NET_DVR_ROI_DETECT_NUM</a>
NET_DVR_GET_ROI_DETECT	dwCount 个 <a href="#">NET_DVR_ROI_DETECT_COND</a>	dwCount 个 <a href="#">NET_DVR_ROI_DETECT_CFG</a>

[返回目录](#)

#### 5.11.4 批量设置配置信息 **NET\_DVR\_SetDeviceConfig**

**函数：** BOOL NET\_DVR\_SetDeviceConfig(LONG lUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpInParameter, DWORD dwInParameterSize)

**参数：**

- [in] lUserID 用户 ID 号，NET\_DVR\_Login\_V40 的返回值
- [in] dwCommand 设备配置命令，详见表 5.17 表 5.18
- [in] dwCount 一次要设置的配置参数个数，0 和 1 都表示 1 个，2 表示 2 个，最大 64 个
- [in] lpInBuffer 配置条件缓冲区，详见表 5.18
- [in] dwInBufferSize 缓冲区长度
- [out] lpStatusList 错误信息列表，和要设置的配置一一对应，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- 成功，大于 0-失败
- [in] lpInParameter 需要设置给设备的参数内容（详见表 5.18），和 lpInBuffer 一一对应。如果某个配置对应的 lpStatusList 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] dwInParameterSize 设置内容缓冲区大小

**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 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设置子设备配置信息的通用接口，`lpInBuffer` 指定需要设置哪 `dwCount` 个，`lpInParamBuffer` 是设置 `dwCount` 个配置的参数信息。不同的获取功能对应不同的结构体和命令号，如表 5.18 所示。

表 5.17 参数批量设置命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_SET_MULTI_STREAM_COMPRESSIONCFG	远程设置多码流压缩参数	3217
NET_DVR_SET_ROI_DETECT	设置 ROI 检测区域配置	3351

表 5.18 批量设置设备参数

dwCommand	lpInBuffer 对应结构体	lpInParamBuffer 对应结构体
NET_DVR_SET_MULTI_STREAM_COMPRESSIONCFG	dwCount 个 <a href="#">NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND</a>	dwCount 个 <a href="#">NET_DVR_MULTI_STREAM_COMPRESSIONCFG</a>
NET_DVR_SET_ROI_DETECT	dwCount 个 <a href="#">NET_DVR_ROI_DETECT_COND</a>	dwCount 个 <a href="#">NET_DVR_ROI_DETECT_CFG</a>

[返回目录](#)

## 抓拍参数配置

### 5.11.5 获取设备的配置信息 `NET_DVR_GetDVRConfig`

**函数：** `BOOL NET_DVR_GetDVRConfig(LONG lUserID, DWORD dwCommand, LONG lChannel, LPVOID lpOutBuffer, DWORD dwOutBufferSize, LPDWORD lpBytesReturned)`

**参数：**

- [in] lUserID 用户 ID 号，`NET_DVR_Login_V40` 的返回值
- [in] dwCommand 设备配置命令，详见表 5.19
- [in] lChannel 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 5.19
- [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 5.19 所示。

- 对于智能交通摄像机，设备只有一个通道，通道号即为 1。
- 当命令为 `NET_ITC_GET_TRIGGERCFG` 或 `NET_ITC_GET_TRIGGER_DEFAULTCFG` 时，`lChannel` 有效，0 表示获取当前启用的触发模式参数，其他值详见 [ITC\\_TRIGGERMODE\\_TYPE](#)。
- 设备支持的 IO 输入输出个数、单 IO 触发组、红绿灯模式组数、测速模式组数、支持的触发模式通过能力集获取 ([NET\\_DVR\\_GetDeviceAbility](#))。

表 5.19 ITC 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_FTPCFG	获取 FTP 上传参数	无效	<a href="#">NET_DVR_FTPCFG</a>	134
NET_DVR_GET_CCDPARAMCFG	获取前端 CCD 参数	通道号	<a href="#">NET_DVR_CAMERAPARAMCFG</a>	1067

NET_DVR_GET_IOINCFG	获取 IO 输入参数	IO 输入口号, 从 1 开始	<a href="#">NET_DVR_IO_INCFG</a>	1070
NET_DVR_GET_IOOUTCFG	获取 IO 输出参数	IO 输出口号, 从 1 开始	<a href="#">NET_DVR_IO_OUTCFG</a>	1072
NET_DVR_GET_FLASHCFG	获取闪光灯参数	IO 输出口号, 从 1 开始	<a href="#">NET_DVR_FLASH_OUTCFG</a>	1074
NET_DVR_GET_LIGHTSNAPCFG	获取红绿灯功能参数	红绿灯组号, 从 1 开始	<a href="#">NET_DVR_LIGHTSNAPCFG</a>	1076
NET_DVR_GET_MEASURESPEEDCFG	获取测速功能参数	测速模式组号, 从 1 开始	<a href="#">NET_DVR_MEASURESPEEDCFG</a>	1078
NET_DVR_GET_IMAGEOVERLAYCFG	获取图像叠加信息参数	无效	<a href="#">NET_DVR_IMAGEOVERLAYCFG</a>	1080
NET_DVR_GET_SNAPCFG	获取单 IO 触发功能参数	单 IO 触发组号, 从 1 开始	<a href="#">NET_DVR_SNAPCFG</a>	1082
NET_DVR_GET_VTPPARAM	获取虚拟线圈参数	无效	<a href="#">NET_DVR_VTPARAM</a>	1084
NET_DVR_GET_SNAPENABLECFG	获取使能参数	无效	<a href="#">NET_DVR_SNAPENABLECFG</a>	1086
NET_DVR_GET_POSTEPOLICECFG	获取卡口电警参数	无效	<a href="#">NET_DVR_POSTEPOLICECFG</a>	1088
NET_DVR_GET_JPEGCFG_V30	获取 JPEG 参数	无效	<a href="#">NET_DVR_JPEGCFG_V30</a>	1090
NET_DVR_GET_SPRCFG	获取车牌识别参数	无效	<a href="#">NET_DVR_SPRCFG</a>	1092
NET_DVR_GET_PLCCFG	获取车牌亮度补偿参数	无效	<a href="#">NET_DVR_PLCCFG</a>	1094
NET_DVR_GET_DEVICESTATECFG	获取设备状态参数	无效	<a href="#">NET_DVR_DEVICESTATECFG</a>	1096
NET_DVR_GET_CALIBRATE_TIME	获取时间校时 (扩展)	无效	<a href="#">NET_DVR_CALIBRATE_TIME</a>	1098
NET_ITC_GET_TRIGGERCFG	获取触发参数 (3.1 及之后版本)	触发模式	<a href="#">NET_ITC_TRIGGERCFG</a>	3003
NET_ITC_GET_IOOUT_PARAM_CFG	获取 IO 输出参数 (3.1 及之后版本)	IO 输出口号, 从 1 开始	<a href="#">NET_ITC_IOOUT_PARAM</a>	3005
NET_DVR_GET_CAMERA_SETUPCFG	获取相机架设参数	通道号	<a href="#">NET_DVR_CAMERA_SETUPCFG</a>	3007
NET_ITC_GET_TRIGGER_DEFAULTCFG	获取触发模式推荐参数 (3.1 及之后版本)	触发模式	<a href="#">NET_ITC_SINGLE_TRIGGERCFG</a>	3013
NET_DVR_GET_STATUS_DETECTCFG	获取状态检测使能参数	通道号	<a href="#">NET_DVR_STATUS_DETECTCFG</a>	3015
NET_DVR_GET_TPS_ALARMCFG	获取交通统计报警参数	通道号	<a href="#">NET_DVR_TPS_ALARMCFG</a>	3019
NET_DVR_GET_CABINETCFG	获取机柜参数	无效	<a href="#">NET_DVR_CABINET_CFG</a>	3104
NET_ITC_GET_ICRCFG	获取 ICR 切换参数	无效	<a href="#">NET_ITC_ICRCFG</a>	3115
NET_ITC_GET_EXCEPTIONCFG	获取异常参数	无效	<a href="#">NET_ITC_EXCEPTION</a>	3119
NET_DVR_GET_CURTRIGGERMODE	获取设备当前触发模式	无效	<a href="#">NET_DVR_CURTRIGGERMODE</a>	3130
NET_DVR_GET_GPSDATA CFG	获取 GPS 参数	无效	<a href="#">NET_DVR_GPS_DATA CFG</a>	3131
NET_DVR_GET_CCDPARAMCFG_EX	获取前端 CCD 参数(扩展)	无效	<a href="#">NET_DVR_CAMERA PARAMCFG_EX</a>	3368
NET_ITS_GET_IMGMERGE_CFG	获取图片合成配置参数	通道号	<a href="#">NET_ITS_IMGMERGE_CFG</a>	5063

[返回目录](#)

### 5.11.6 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\_V40 的返回值  
[in] dwCommand 设备配置命令, 详见表 5.20  
[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 5.20  
[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 5.20 所示。
- 对于智能交通摄像机, 设备只有一个通道, 通道号即为 1。
  - 限速值: 如果没有设置 IO 测速模式, 卡口电警 ([NET\\_DVR\\_POSTEPOLICECFG](#)) 中限速值有效; 如果设置 IO 测速模式, 则测速模式 ([NET\\_DVR\\_MEASURESPEEDCFG](#)) 中限速值有效。
  - 当命令为 NET\_ITC\_SET\_TRIGGERCFG 时, IChannel 有效, 取值详见 [ITC\\_TRIGGERMODE\\_TYPE](#)。
  - 设备支持的 IO 输入输出个数、单 IO 触发组、红绿灯模式组数、测速模式组数、支持的触发模式通过能力集获取 ([NET\\_DVR\\_GetDeviceAbility](#))。

表 5.20 ITC 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_FTPCFG	设置 FTP 上传参数	无效	<a href="#">NET_DVR_FTPCFG</a>	135
NET_DVR_SET_CCDPARAMCFG	设置前端 CCD 参数	通道号	<a href="#">NET_DVR_CAMERAPARAMCFG</a>	1068
NET_DVR_SET_IOINCFG	设置 IO 输入参数	IO 输入口号, 从 1 开始	<a href="#">NET_DVR_IO_INCFG</a>	1071
NET_DVR_SET_IOOUTCFG	设置 IO 输出参数	IO 输出口号, 从 1 开始	<a href="#">NET_DVR_IO_OUTCFG</a>	1073
NET_DVR_SET_FLASHCFG	设置闪光灯参数	IO 输出口号, 从 1 开始	<a href="#">NET_DVR_FLASH_OUTCFG</a>	1075
NET_DVR_SET_LIGHTSNAPCFG	设置红绿灯功能参数	红绿灯组号, 从 1 开始	<a href="#">NET_DVR_LIGHTSNAPCFG</a>	1077
NET_DVR_SET_MEASURESPEEDCFG	设置测速功能参数	测速模式组号, 从 1 开始	<a href="#">NET_DVR_MEASURESPEEDCFG</a>	1079
NET_DVR_SET_IMAGEOVERLAYCFG	设置图像叠加信息参数	无效	<a href="#">NET_DVR_IMAGEOVERLAYCFG</a>	1081
NET_DVR_SET_SNAPCFG	设置单 IO 触发功能参数	单 IO 触发组号, 从 1 开始	<a href="#">NET_DVR_SNAPCFG</a>	1083
NET_DVR_SET_VTPPARAM	设置虚拟线圈参数	无效	<a href="#">NET_DVR_VTPPARAM</a>	1085
NET_DVR_SET_SNAPENABLECFG	设置使能参数	无效	<a href="#">NET_DVR_SNAPENABLECFG</a>	1087
NET_DVR_SET_POSTEPOLICECFG	设置卡口电警参数	无效	<a href="#">NET_DVR_POSTEPOLICECFG</a>	1089
NET_DVR_SET_JPEGCFG_V30	设置 JPEG 参数	无效	<a href="#">NET_DVR_JPEGCFG_V30</a>	1091
NET_DVR_SET_SPRCFG	设置车牌识别参数	无效	<a href="#">NET_DVR_SPRCFG</a>	1093
NET_DVR_SET_PLCCFG	设置车牌亮度补偿参数	无效	<a href="#">NET_DVR_PLCCFG</a>	1095
NET_DVR_SET_CALIBRATE_TIME	设置时间校时 (扩展)	无效	<a href="#">NET_DVR_CALIBRATE_TIME</a>	1097
NET_ITC_SET_TRIGGERCFG	设置触发参数 (3.1 及之后版本)	触发模式	<a href="#">NET_ITC_TRIGGERCFG</a>	3004
NET_ITC_SET_IOOUT_PARAM_CFG	设置 IO 输出参数 (3.1 及之后版本)	IO 输出口号, 从 1 开始	<a href="#">NET_ITC_IOOUT_PARAM</a>	3006
NET_DVR_SET_CAMERA_SETUPCFG	设置相机架设参数	通道号	<a href="#">NET_DVR_CAMERA_SETUPCFG</a>	3008

NET_DVR_SET_STATUS_DETECTCFG	设置状态检测使能参数	无效	<a href="#">NET_DVR_STATUS_DETECTCFG</a>	3016
NET_DVR_SET_TPS_ALARMCFG	设置交通统计报警参数	通道号	<a href="#">NET_DVR_TPS_ALARMCFG</a>	3020
NET_DVR_SET_CABINETCFG	设置机柜参数	无效	<a href="#">NET_DVR_CABINET_CFG</a>	3105
NET_ITC_SET_ICRCFG	设置 ICR 切换参数	无效	<a href="#">NET_ITC_ICRCFG</a>	3116
NET_ITC_SET_EXCEPTIONCFG	设置异常参数	无效	<a href="#">NET_ITC_EXCEPTION</a>	3120
NET_DVR_SET_GPSDATA CFG	设置 GPS 参数	无效	<a href="#">NET_DVR_GPS_DATA CFG</a>	3132
NET_DVR_SET_CCDPARAMCFG_EX	设置前端 CCD 参数(扩展)	无效	<a href="#">NET_DVR_CAMERAPARAMCFG_EX</a>	3369
NET_ITS_SET_IMGMERGE_CFG	设置图片合成参数	通道号	<a href="#">NET_ITS_IMGMERGE_CFG</a>	5064

[返回目录](#)

### 5.11.7 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\_V40 的返回值
  - [in] dwCommand 设备配置命令, 详见表 5.21
  - [in] dwCount 一次要获取配置参数的个数, 0 和 1 都表示 1 个信息, 2 表示 2 个信息, 最大 64 个
  - [in] lpInBuffer 配置条件缓冲区, 详见表 5.22
  - [in] dwInBufferSize 缓冲区长度
  - [out] lpStatusList 错误信息列表, 和要查询的配置一一对应, 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- 成功, 大于 0-失败
  - [out] lpOutBuffer 设备返回的参数内容 (详见表 5.22), 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 5.21 ITC 参数批量获取命令

dwCommand 宏定义	含义	宏定义值
NET_ITC_GET_VIDEO_TRIGGERCFG	获取视频触发参数	3017
NET_ITC_GET_RS485_ACCESSINFO	获取 RS485 关联接入设备的信息	3117
NET_ITC_GET_FTPCFG	获取 FTP 信息	3121
NET_DVR_GET_ENTRANCE_PARAMCFG	获取出入口控制参数	3126
NET_DVR_GET_CLOUDSTORAGE_CFG	获取云存储配置参数	5058
NET_ITS_GET_OVERLAP_CFG_V50	获取字符叠加参数配置	5055
NET_DVR_GET_TRIGGEREX_CFG	获取触发模式配置	5074

**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。IpInBuffer 指定需要获取哪 dwCount 个信息，IpOutBuffer 保存获取得到的 dwCount 个相应配置信息。不同的获取功能对应不同的结构体和命令号，如表 5.22 所示。

表 5.22 批量获取 ITC 参数

dwCommand 宏定义	IpInBuffer 对应结构体	IpOutBuffer 对应结构体
NET_ITC_GET_VIDEO_TRIGGERCFG	dwCount 个 <a href="#">NET_ITC_VIDEO_TRIGGER_COND</a>	dwCount 个 <a href="#">NET_ITC_VIDEO_TRIGGER_PARAM</a>
NET_ITC_GET_RS485_ACCESSINFO	dwCount 个 <a href="#">NET_ITC_RS485_ACCESS_INFO_COND</a>	dwCount 个 <a href="#">NET_ITC_RS485_ACCESS_CFG</a>
NET_ITC_GET_FTPCFG	dwCount 个 <a href="#">NET_ITC_FTP_TYPE_COND</a>	dwCount 个 <a href="#">NET_ITC_FTP_CFG</a>
NET_DVR_GET_ENTRANCE_PARAMCFG	dwCount 个 <a href="#">NET_DVR_BARRIERGATE_COND</a>	dwCount 个 <a href="#">NET_DVR_ENTRANCE_CFG</a>
NET_DVR_GET_CLOUDSTORAGE_CFG	dwCount 个 <a href="#">NET_DVR_CLOUDSTORAGE_COND</a>	dwCount 个 <a href="#">NET_DVR_CLOUDSTORAGE_CFG</a>
NET_ITS_GET_OVERLAP_CFG_V50	dwCount 个 <a href="#">NET_ITS_OVERLAPCFG_COND</a>	dwCount 个 <a href="#">NET_ITS_OVERLAP_CFG_V50</a>
NET_DVR_GET_TRIGGEREX_CFG	dwCount 个 <a href="#">NET_DVR_TRIGGER_COND</a>	dwCount 个 <a href="#">NET_ITC_TRIGGERCFG</a>

[返回目录](#)

### 5.11.8 批量设置配置信息 **NET\_DVR\_SetDeviceConfig**

**函数：** BOOL NET\_DVR\_SetDeviceConfig(LONG IUserID, DWORD dwCommand, DWORD dwCount, LPVOID IpInBuffer, DWORD dwInBufferSize, LPVOID IpStatusList, LPVOID IpInParameter, DWORD dwInParameterSize)

**参数：**

- [in] IUserID 用户 ID 号，NET\_DVR\_Login\_V40 的返回值
- [in] dwCommand 设备配置命令，详见表 5.23
- [in] dwCount 一次要设置的配置参数个数，0 和 1 都表示 1 个，2 表示 2 个，最大 64 个
- [in] IpInBuffer 配置条件缓冲区，详见表 5.24
- [in] dwInBufferSize 缓冲区长度
- [out] IpStatusList 错误信息列表，和要设置的配置一一对应，例如 IpStatusList[2]就对应 IpInBuffer[2]，由用户分配内存，每个错误信息为 4 个字节，参数值：0- 成功，大于 0-失败
- [in] IpInParameter 需要设置给设备的参数内容（详见表 5.24），和 IpInBuffer 一一对应。如果某个配置对应的 IpStatusList 信息为大于 0 值，表示对应的 IpInBuffer 设置失败，为 0 则设置成功
- [in] dwInParameterSize 设置内容缓冲区大小

表 5.23 ITC 参数批量设置命令

dwCommand 宏定义	含义	宏定义值
NET_ITC_SET_VIDEO_TRIGGERCFG	设置视频触发参数	3018
NET_ITC_SET_RS485_ACCESSINFO	设置 RS485 关联接入设备的信息	3118
NET_ITC_SET_FTPCFG	设置 FTP 信息	3122
NET_DVR_SET_ENTRANCE_PARAMCFG	设置出入口控制参数	3127
NET_DVR_SET_CLOUDSTORAGE_CFG	设置云存储配置参数	5059

NET_ITS_SET_OVERLAP_CFG_V50	设置字符叠加参数配置	5056
NET_DVR_SET_TRIGGEREX_CFG	设置触发模式配置	5075

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设置子设备配置信息的通用接口, lpInBuffer 指定需要设置哪 dwCount 个, lpInParamBuffer 是设置 dwCount 个配置的参数信息。不同的获取功能对应不同的结构体和命令号, 如表 5.24 所示。

表 5.24 批量设置 ITC 参数

dwCommand 宏定义	lpInBuffer 对应结构体	lpInParamBuffer 对应结构体
NET_ITC_SET_VIDEO_TRIGGERCFG	dwCount 个 <a href="#">NET_ITC_VIDEO_TRIGGER_COND</a>	dwCount 个 <a href="#">NET_ITC_VIDEO_TRIGGER_PARAM</a>
NET_ITC_SET_RS485_ACCESSINFO	dwCount 个 <a href="#">NET_ITC_RS485_ACCESS_INFO_COND</a>	dwCount 个 <a href="#">NET_ITC_RS485_ACCESS_CFG</a>
NET_ITC_SET_FTPCFG	dwCount 个 <a href="#">NET_ITC_FTP_TYPE_COND</a>	dwCount 个 <a href="#">NET_ITC_FTP_CFG</a>
NET_DVR_SET_ENTRANCE_PARAMCFG	dwCount 个 <a href="#">NET_DVR_BARRIERGATE_COND</a>	dwCount 个 <a href="#">NET_DVR_ENTRANCE_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_ITS_SET_OVERLAP_CFG_V50	dwCount 个 <a href="#">NET_ITS_OVERLAPCFG_COND</a>	dwCount 个 NET_ITS_OVERLAP_CFG_V50
NET_DVR_SET_TRIGGEREX_CFG	dwCount 个 <a href="#">NET_DVR_TRIGGER_COND</a>	dwCount 个 <a href="#">NET_ITC_TRIGGERCFG</a>

[返回目录](#)

### 5.11.9 获取设备参数 [NET\\_DVR\\_GetSTDConfig](#)

函数: BOOL NET\_DVR\_GetSTDConfig(LONG IUserID, DWORD dwCommand, LPNET\_DVR\_STD\_CONFIG lpConfigParam)

参数: [in]IUserID 用户 ID 号, NET\_DVR\_Login\_V40 的返回值  
[in]dwCommand 设备配置命令, 详见表 5.25  
[in&out]lpConfigParam 配置输入输出参数, 不同的配置功能对应不同的输入输出参数, 详见表 5.25

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:** 获取配置参数时, lpConfigParam 结构体中的 lpInBuffer 无效, 设为 NULL。对于不同的配置功能 (dwCommand), lpConfigParam 中 lpCondBuffer、lpOutBuffer 分别对应不同的内容, 具体如表 5.25 所示。

表 5.25 获取设备参数

dwCommand 宏定义	dwCommand 含义	lpCondBuffer	lpOutBuffer	宏定义值
NET_DVR_GET_ALARMCTRL_CFG	获取报警控制	4 字节(DWORD)通道号	<a href="#">NET_DVR_ALARMCTRL_CFG</a>	3148

[返回目录](#)

### 5.11.10 设置设备参数 **NET\_DVR\_SetSTDConfig**

函数: BOOL NET\_DVR\_SetSTDConfig(LONG IUserID, DWORD dwCommand, LPNET\_DVR\_STD\_CONFIG lpConfigParam)

参数: [in]IUserID 用户 ID 号, NET\_DVR\_Login\_V40 的返回值  
[in]dwCommand 设备配置命令, 详见表 5.26  
[in&out]lpConfigParam 配置输入输出参数, 不同的配置功能对应不同的输入输出参数, 详见表 5.26

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: 设置配置参数时, lpConfigParam 结构体里面的 lpOutBuffer 无效, 设为 NULL。对于不同的配置功能 (dwCommand), lpConfigParam 中的 lpCondBuffer、lpInBuffer 分别对应不同的内容, 具体如表 5.26 所示。

表 5.26 设置设备参数

dwCommand 宏定义	dwCommand 含义	lpCondBuffer	lpInBuffer	宏定义值
NET_DVR_SET_ALARMCTRL_CFG	设置报警控制	4 字节(DWORD)通道号	<a href="#">NET_DVR_ALARMCTRL_CFG</a>	3147

[返回目录](#)

## 长连接参数配置

### 5.11.11 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\_V40 返回值  
[in] dwCommand 配置命令, 详见表 5.27  
[in] lpInBuffer 输入参数, 具体内容跟配置命令相关, 详见表 5.27  
[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, 具体取值参见表 5.29  
[out] lpBuffer 存放数据的缓冲区指针, 具体取值参见表 5.29  
[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: 不同的功能对应不同的命令号(dwCommand), 同时 lpInBuffer 对应不同的结构体, 如表 5.27 所示。调用该接口启动长连接远程配置后, 还需要调用其他接口获取或者设置相关参数, 如表 5.28 所示。



表 5.27 长连接配置

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer	回调函数
NET_DVR_VEHICLE_CONTROL_LIST_START	3123	设置车辆允许名单信息	NULL	返回状态
NET_DVR_GET_ALL_VEHICLE_CONTROL_LIST	3124	获取所有车辆允许名单信息	<a href="#">NET_DVR_VEHICLE_CONTROL_COND</a>	NULL
NET_DVR_VEHICLELIST_CTRL_START	3133	设置车辆允许名单信息 (批量)	NULL	返回状态

表 5.28 后续接口调用

dwCommand 宏定义	后续接口调用
NET_DVR_VEHICLE_CONTROL_LIST_START	<a href="#">NET_DVR_SendRemoteConfig</a>
NET_DVR_GET_ALL_VEHICLE_CONTROL_LIST	<a href="#">NET_DVR_GetNextRemoteConfig</a>
NET_DVR_VEHICLELIST_CTRL_START	<a href="#">NET_DVR_SendRemoteConfig</a>

表 5.29 长连接回调参数取值

dwCommand 宏定义	dwType	lpInBuffer
NET_DVR_VEHICLE_CONTROL_LIST_START	typedef enum{ NET_SDK_CALLBACK_STATUS_SUCCESS = 1000, //成功 NET_SDK_CALLBACK_STATUS_PROCESSING, //处理中 NET_SDK_CALLBACK_STATUS_FAILED //失败 }NET_SDK_CALLBACK_STATUS_NORMAL;	NULL
NET_DVR_VEHICLELIST_CTRL_START	typedef enum{ NET_SDK_CALLBACK_TYPE_STATUS = 0, //回调状态值 NET_SDK_CALLBACK_TYPE_PROGRESS, //回调进度值 NET_SDK_CALLBACK_TYPE_DATA //回调数据内容 }NET_SDK_CALLBACK_TYPE;	dwType 为 0 时, 前 4 个字节(DWORD)的值对应状态, 详见: NET_SDK_CALLBACK_STATUS_NORMAL 如果 lpBuffer 前 4 个字节数据值为 NET_SDK_CALLBACK_STATUS_FAILED 时, 后面每 4 个字节表示一个错误号

[返回目录](#)

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

函数: LONG NET\_DVR\_GetNextRemoteConfig(LONG lHandle, void \*lpOutBuff, DWORD dwOutBuffSize)

参数: [in] lHandle 查找句柄, NET\_DVR\_StartRemoteConfig 的返回值  
[in] lpOutBuff 输出数据缓冲区, 与 NET\_DVR\_StartRemoteConfig 的命令 (dwCommand) 有关, 详见表 5.31  
[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 5.30 所示。获取错误码调用 [NET\\_DVR\\_GetLastError](#)。

表 5.30 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_NEXT_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 5.31 长连接获取参数

dwCommand 宏定义	宏定义值	控制功能	lpOutBuff 对应结构体
NET_DVR_GET_ALL_VEHICLE_CONTROL_LIST	3124	获取所有车辆允许名单信息	<a href="#">NET_DVR_VEHICLE_CONTROL_LIST_INFO</a>

[返回目录](#)

### 5.11.13 发送长连接数据 [NET\\_DVR\\_SendRemoteConfig](#)

**函数：** BOOL NET\_DVR\_SendRemoteConfig(LONG IHandle, DWORD dwDataType, char \*pSendBuf, DWORD dwBufSize)

**参数：** [in] IHandle 长连接句柄，NET\_DVR\_StartRemoteConfig 的返回值  
[in] dwDataType 数据类型，详见表 5.32  
[in] pSendBuf 保存发送数据的缓冲区，与 dwDataType 有关，详见表 5.32  
[in] dwBufSize 发送数据的长度

**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：** 在调用该接口之前，必须先调用 [NET\\_DVR\\_StartRemoteConfig](#) 获取长连接句柄。不同的数据类型(dwDataType)，pSendBuf 对应不同的结构体，如表 5.32 所示。

表 5.32 长连接发送数据

dwCommand 宏定义	宏定义值	含义	
NET_DVR_VEHICLE_CONTROL_LIST_START	3123	设置车辆允许名单信息长连接	
dwDataType 宏定义	宏定义值	控制功能	lpOutBuff 对应结构体
DVR_VEHICLE_CONTROL_LIST	0x1	发送车辆允许名单数据	<a href="#">NET_DVR_VEHICLE_CONTROL_LIST_INFO</a>

dwCommand 宏定义	宏定义值	含义	
NET_DVR_VEHICLELIST_CTRL_START	3133	设置车辆允许名单信息(批量)	
dwDataType 宏定义	宏定义值	含义	pSendBuf 对应结构体
ENUM_SENDDATA	0x0	发送车辆允许名单数据	m 个(m 最大为 128) <a href="#">NET_DVR_VEHICLE_CONTROL_LIST_INFO</a>

[返回目录](#)

### 5.11.14 关闭长连接配置 **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:** 关闭长连接配置接口所创建的句柄, 释放资源。

[返回目录](#)

## 获取设备支持的云台协议

### 5.11.15 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\_V40 的返回值  
 [out]pPtzcfg 设备的云台协议, 请参见 [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](#)

## 5.12 证书管理

### 证书创建、删除

#### 5.12.1 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\_V40 的返回值  
 [in]dwCommand 控制命令, 详见表 5.33  
 [in]lpInBuffer 输入参数, 具体内容跟控制命令相关, 详见表 5.33  
 [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 5.33 所示。

表 5.33 远程配置命令

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer 对应结构体
NET_DVR_CREATE_CERT	6138	创建证书	<a href="#">NET_DVR_CERT_INFO</a>
NET_DVR_DELETE_CERT	6139	删除证书	<a href="#">NET_DVR_CERT_PARAM</a>

[返回目录](#)

## 证书信息获取

### 5.12.2 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\_V40 的返回值
  - [in] dwCommand 设备配置命令，详见表 5.34
  - [in] dwCount 一次要获取配置参数的个数，0 和 1 都表示 1 个信息，2 表示 2 个信息，最大 64 个
  - [in] lpInBuffer 配置条件缓冲区，详见表 5.34
  - [in] dwInBufferSize 缓冲区长度
  - [out] lpStatusList 错误信息列表，和要查询的配置一一对应，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 5.34），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
- 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 指定需要获取的信息，lpOutBuffer 保存获取得到的 dwCount 个配置信息。不同的 dwCommand 对应不同的结构体和命令号，如表 5.34 所示。

表 5.34 参数批量获取命令

dwCommand 宏定义	含义	lpInBuffer 对应结构体	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_CERT	获取证书信息	dwCount 个 NET_DVR_CERT_PARAM	dwCount 个 <a href="#">NET_DVR_CERT_INFO</a>	6142

[返回目录](#)

## 证书上传下载

### 5.12.3 上传文件 **NET\_DVR\_UploadFile\_V40**

- 函 数：** LONG NET\_DVR\_UploadFile\_V40(LONG IUserID, DWORD dwUploadType, LPVOID lpInBuffer, DWORD dwInBufferSize, char \*sFileName, LPVOID lpOutBuffer, DWORD dwOutBufferSize)
- 参 数：**
- [in] IUserID NET\_DVR\_Login\_V40 的返回值
  - [in] dwUploadType 上传文件类型，详见表 5.35
  - [in] lpInBuffer 不同的 dwUploadType，输入参数不同，详见表 5.35
  - [in] dwInBufferSize 输入缓冲区大小

[in] sFileName 上传文件的绝对路径（包括文件名）  
 [out] lpOutBuffer 输出参数，不同的 dwUploadType，输出参数不同，详见表 5.35  
 [in] dwInBufferSize 输出缓冲区大小  
 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：

表 5.35 文件上传类型

dwUploadType 宏定义	宏定义值	含义	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
UPLOAD_CERTIFICATE	1	上传证书(WIFI、HTTPS 等)	<a href="#">NET_DVR_CERT_PARAM</a>	NULL

[返回目录](#)

#### 5.12.4 获取文件上传的进度和状态 [NET\\_DVR\\_GetUploadState](#)

函 数： LONG NET\_DVR\_GetUploadState(LONG IUploadHandle,LPDWORD pProgress)

参 数： [in] IUploadHandle 文件上传的句柄，NET\_DVR\_UploadFile 的返回值  
 [out] pProgress 返回的进度值，取值范围：0~100

返回值： -1 表示函数调用失败，其他为上传的状态值：1- 上传成功；2- 正在上传；3- 上传失败；4- 网络断开，状态未知；6- 硬盘错误；7- 无审讯文件存放盘；8- 容量不足；9- 设备资源不足；10- 文件个数超过 40；19- 文件格式不正确；20- 文件内容不正确。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](#)

#### 5.12.5 停止文件上传 [NET\\_DVR\\_UploadClose](#)

函 数： BOOL NET\_DVR\_UploadClose(LONG IUploadHandle)

参 数： [in] IUploadHandle 文件上传的句柄，NET\_DVR\_UploadFile 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](#)

#### 5.12.6 开始下载文件 [NET\\_DVR\\_StartDownload](#)

函 数： LONG NET\_DVR\_StartDownload(LONG IUserID, DWORD dwDownloadType, LPVOID lpInBuffer, DWORD dwInBufferSize, char const \*sFileName)

参 数： [in] IUserID 用户 ID，NET\_DVR\_Login\_V40 的返回值  
 [in] dwDownloadType 下载文件类型，详见表 5.36  
 [in] lpInBuffer 输入参数。不同的 dwUploadType，输入参数不同，详见表 5.36  
 [in] dwInBufferSize 输入缓冲区大小  
 [in] sFileName 下载文件的保存路径（绝对路径，包括文件名）

返回值： -1 表示失败，其他值作为 NET\_DVR\_StopDownload 和 NET\_DVR\_GetDownloadState 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：**

表 5.36 文件下载类型

dwDownloadType	宏定义值	含义	lpInBuffer 对应结构体
NET_SDK_DOWNLOAD_CERT	0	下载证书	<a href="#">NET_DVR_CERT_PARAM</a>

[返回目录](#)

### 5.12.7 获取文件下载的进度和状态 [NET\\_DVR\\_GetDownloadState](#)

函 数： LONG NET\_DVR\_GetDownloadState(LONG IDownloadHandle, LPDWORD pProgress)

参 数： [in] IDownloadHandle 文件下载的句柄，NET\_DVR\_StartDownload 的返回值  
[out] pProgress 返回的进度值，取值范围：0~100

返回值： -1 表示函数调用失败，其他为下载的状态值：1- 下载成功；2- 正在下载；3- 下载失败；4- 网络断开，状态未知。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](#)

### 5.12.8 停止文件下载 [NET\\_DVR\\_StopDownload](#)

函 数： BOOL NET\_DVR\_StopDownload(LONG IHandle);

参 数： [in] IHandle 文件下载的句柄，NET\_DVR\_StartDownload 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](#)

## 5.13 远程控制

### 5.13.1 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\_V40 返回值  
[in] dwCommand 控制命令，详见表 5.37  
[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 5.37 所示。

表 5.37 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_VEHICLE_DELINFO_CTRL	3125	删除设备内禁止名单数据库信息	<a href="#">NET_DVR_VEHICLE_CONTROL_DELINFO</a>
NET_DVR_BARRIERGATE_CTRL	3128	远程控制道闸	<a href="#">NET_DVR_BARRIERGATE_CFG</a>
NET_DVR_GATELAMP_CTRL	3129	常亮灯功能	<a href="#">NET_DVR_GATELAMP_INFO</a>

[返回目录](#)

## 5.14 手动触发抓拍

### 5.14.1 网络触发抓拍 **NET\_DVR\_ContinuousShoot**

函数: BOOL NET\_DVR\_ContinuousShoot(LONG lUserID, LPNET\_DVR\_SNAPCFG lpInter)

参数: [in] lUserID NET\_DVR\_Login\_V40 的返回值

[in] lpInter 网络触发抓拍配置, 请参见 [NET\\_DVR\\_SNAPCFG](#)

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: 使用网络触发连拍接收图片, 需要先对设备进行布防 ([NET\\_DVR\\_SetupAlarmChan\\_V41](#)) 并设置回调函数 ([NET\\_DVR\\_SetDVRMessageCallBack\\_V30](#)), 通过回调获取抓拍的图片数据。

[返回目录](#)

### 5.14.2 手动抓拍 **NET\_DVR\_ManualSnap**

函数: BOOL NET\_DVR\_ManualSnap(LONG lUserID, LPNET\_DVR\_MANUALSNAP lpInter, LPNET\_DVR\_PLATE\_RESULT lpOuter);

参数: [in] lUserID NET\_DVR\_Login\_V40 的返回值

[in] lpInter 手动抓拍参数, 请参见 [NET\\_DVR\\_MANUALSNAP](#)

[out] lpOuter 抓拍结果, 请参见 [NET\\_DVR\\_PLATE\\_RESULT](#)

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: 对于抓拍机, 该接口一般用于调试, 不能频繁调用进行抓图。调用该接口前, 应用层需要自己先分配 lpOuter 输出结构体里面的 pBuffer1 等缓冲区的内存, 不能小于抓图图片的数据大小。

[返回目录](#)

## 5.15 雨刷控制

### 5.15.1 云台控制操作 **NET\_DVR\_PTZControl\_Other**

函数: BOOL NET\_DVR\_PTZControl\_Other(LONG lUserID, LONG lChannel, DWORD dwPTZCommand, DWORD dwStop)

参 数: [in] IUserID NET\_DVR\_Login\_V40 的返回值  
[in] IChannel 设备通道号  
[in] dwPTZCommand 云台控制命令, 详见表 5.38  
[in] dwStop 云台停止动作或开始动作: 0- 开始, 1- 停止

表 5.38 云台控制命令

dwPTZCommand 宏定义	宏定义值	含义
WIPER_PWRON	3	接通雨刷开关

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](#)

## 5.16 图片的查找、下载

### 查找图片

#### 5.16.1 根据类型和时间查找图片 **NET\_DVR\_FindPicture**

函 数: LONG NET\_DVR\_FindPicture(LONG IUserID, NET\_DVR\_FIND\_PICTURE\_PARAM\* pFindParam)  
参 数: [in]IUserID NET\_DVR\_Login\_V40 的返回值  
[in]pFindParam 图片查找条件信息, 请参见 [NET\\_DVR\\_FIND\\_PICTURE\\_PARAM](#)

返回值: -1 表示失败, 其他值作为 NET\_DVR\_CloseFindPicture 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: 该接口指定了要查找的图片的类型和查找时间范围, 调用成功后, 就可以调用 NET\_DVR\_FindNextPicture 接口来获取图片信息。注: 该接口查找的是设备本地的图片, 可通过 NET\_DVR\_SetDVRConfig 配置设备的抓图计划 (NET\_DVR\_SCHED\_CAPTURECFG)。

[返回目录](#)

#### 5.16.2 逐个获取查找到的图片 **NET\_DVR\_FindNextPicture**

函 数: LONG NET\_DVR\_FindNextPicture(LONG IFindHandle, LPNET\_DVR\_FIND\_PICTURE lpFindData)  
参 数: [in]IFindHandle 图片查找句柄, NET\_DVR\_FindPicture 的返回值  
[out]lpFindData 保存图片信息的指针, 请参见 [NET\\_DVR\\_FIND\\_PICTURE](#)

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 5.39 所示。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。

表 5.39 查找状态信息

宏定义	宏定义值	含义
NET_DVR_FILE_SUCCESS	1000	获取图片信息成功
NET_DVR_FILE_NOFIND	1001	未查找到图片



NET_DVR_ISFINDING	1002	正在查找请等待
NET_DVR_NOMOREFILE	1003	没有更多的图片，查找结束
NET_DVR_FILE_EXCEPTION	1004	查找图片时异常

**说明：** 在调用该接口获取查找图片之前，必须先调用 NET\_DVR\_FindPicture 得到当前的查找句柄。此接口用于获取一条已查找到的图片信息，若要获取全部的已查找到的图片信息，需要循环调用此接口。

[返回目录](#)

### 5.16.3 关闭图片查找，释放资源 NET\_DVR\_CloseFindPicture

**函数：** BOOL NET\_DVR\_CloseFindPicture(LONG IFindHandle)

**参数：** [in]IFindHandle 图片查找句柄，NET\_DVR\_FindPicture 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](#)

## 图片下载

### 5.16.4 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 IUserID, char \*sDVRFileName, char \*sSavedFileBuf, DWORD dwBufLen, DWORD \*lpdwRetLen)

**参数：** [in]IUserID NET\_DVR\_Login\_V40 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](#)

### 5.16.5 图片下载 NET\_DVR\_GetPicture

**函数：** BOOL NET\_DVR\_GetPicture(LONG IUserID, char \*sDVRFileName, const char \*sSavedFileName)

**参数：** [in]IUserID NET\_DVR\_Login\_V40 的返回值

[in] sDVRFileName 要下载的图片名称

[in] sSavedFileName 下载图片的保存路径（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：** 图片为 JPEG 格式，文件名后缀为.jpg。

[返回目录](#)

## 5.17 文件上传

### 5.17.1 上传文件 **NET\_DVR\_UploadFile\_V40**

**函数：** LONG NET\_DVR\_UploadFile\_V40(LONG IUserID, DWORD dwUploadType, LPVOID lpInBuffer, DWORD dwInBufferSize, char \*sFileName, LPVOID lpOutBuffer, DWORD dwOutBufferSize)

**参数：**

- [in] IUserID                      NET\_DVR\_Login\_V40 的返回值
- [in] dwUploadType                上传文件类型，详见表 5.40
- [in] lpInBuffer                  输入参数，不同的 dwUploadType，输入参数不同，详见表 5.40
- [in] dwInBufferSize              输入缓冲区大小
- [in] sFileName                    上传文件的路径（包括文件名）
- [out] lpOutBuffer                输出参数，不同的 dwUploadType，输出参数不同，详见表 5.40
- [in] dwOutBufferSize            输出缓冲区大小

**返回值：** -1 表示失败，其他值作为 NET\_DVR\_CloseFindPicture 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：** 上传文件类型不同，对应不同的命令号（dwUploadType），输入参数也对应不同的结构体，如表 5.40 所示。

表 5.40 上传文件类型

dwUploadType 宏定义	宏定义值	dwUploadType 含义	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
UPLOAD_VIOLATION_FILE	7	上传违法字典文件	NULL	NULL
UPLOAD_SUBBRAND_FILE	23	上传车辆子品牌文件	NULL	NULL

[返回目录](#)

### 5.17.2 获取文件上传的进度和状态 **NET\_DVR\_GetUploadState**

**函数：** LONG NET\_DVR\_GetUploadState(LONG IUploadHandle, LPDWORD pProgress)

**参数：**

- [in] IUploadHandle                文件上传的句柄，NET\_DVR\_UploadFile\_V40 的返回值
- [out] pProgress                    返回的进度值，取值范围：0~100

**返回值：** -1 表示函数调用失败，其他为上传的状态值：1- 上传成功；2- 正在上传；3- 上传失败；4- 网络断开，状态未知。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 停止文件上传，释放资源 **NET\_DVR\_UploadClose**

**函数：** BOOL NET\_DVR\_UploadClose(LONG IUploadHandle)

**参数：** [in] IUploadHandle                文件上传的句柄，NET\_DVR\_UploadFile\_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](#)

## 5.18 获取设备状态

### 5.18.1 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_V40 的返回值
[in] dwCommand	获取设备状态的命令值，详见表 5.41
[in] dwCount	要获取的状态个数，设为 1
[in] lpInBuffer	配置条件缓冲区，详见表 5.41
[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（详见表 5.41），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

**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和命令号，如表 5.41 所示。

表 5.41 设备状态

dwCommand 宏定义	dwCommand 含义	lpInBuffer	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_SIGNALLAMP_STATUS	信号灯检测	dwCount 个 <a href="#">NET_DVR_EXDEVDET_COND</a>	dwCount 个 <a href="#">NET_DVR_EXDEVDET_CFG</a>	5099

[返回目录](#)

## 5.19 设备维护管理

### 在线状态检测

#### 5.19.1 设备在线状态检测 **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\_V40 的返回值  
[in]dwCommand 控制命令, 详见表 5.42  
[in]lpInBuffer 输入参数, 具体内容跟控制命令相关, 详见表 5.42  
[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: 该控制命令用于手动检测设备是否在线, 接口返回 TRUE 表示在线, FALSE 表示与设备通信失败或者返回错误状态。设备在线状态自动巡检功能通过 [NET\\_DVR\\_SetSDKLocalCfg](#) (配置类型: NET\_SDK\_LOCAL\_CFG\_TYPE\_CHECK\_DEV) 进行配置。

表 5.42 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_CHECK_USER_STATUS	20005	检测设备是否在线	NULL

[返回目录](#)

## 远程升级

### 5.19.2 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](#)

### 5.19.3 远程升级 [NET\\_DVR\\_Upgrade](#)

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

参 数: [in] IUserID NET\_DVR\_Login\_V40 的返回值  
[in] sFileName 升级的文件路径 (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 (包括文件名在内)。

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](#)

### 5.19.4 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](#)

### 5.19.5 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](#)

### 5.19.6 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

### 5.19.7 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\_V40 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[设备日志主类型](#)

[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[设备日志次类型](#)

[in] lpStartTime 文件的开始时间, 请参见结构体 [NET\\_DVR\\_TIME](#)

[in] lpStopTime 文件结束时间, 请参见结构体 [NET\\_DVR\\_TIME](#)

[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](#)

### 5.19.8 逐条获取查找到的日志信息 [NET\\_DVR\\_FindNextLog\\_V30](#)

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

参数: [in]lLogHandle 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, 请参见结构体 [NET\\_DVR\\_LOG\\_V30](#)

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](#)

### 5.19.9 释放查找日志的资源 [NET\\_DVR\\_FindLogClose\\_V30](#)

函数: BOOL NET\_DVR\_FindLogClose\_V30(LONG lLogHandle)

参数: [in]lLogHandle 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

### 5.19.10 恢复设备默认参数 [NET\\_DVR\\_RestoreConfig](#)

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

参数: [in] lUserID 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](#)

### 5.19.11 完全恢复出厂默认参数 [NET\\_DVR\\_RemoteControl](#)

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

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

[in] dwCommand                    控制命令，详见表 5.43  
 [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 5.43 所示。

表 5.43 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_COMPLETE_RESTORE_CTRL	3420	设置完全恢复出厂值	<a href="#">NET_DVR_COMPLETE_RESTORE_INFO</a>

[返回目录](#)

## 导入/导出配置文件

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

函 数: BOOL NET\_DVR\_GetConfigFile\_V30(LONG lUserID, char \*sOutBuffer, DWORD dwOutSize, DWORD \*pReturnSize)

参 数: [in] lUserID                    用户 ID 号，NET\_DVR\_Login\_V40 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 时用于获取参数配置文件的所需的缓冲区长度；

当 sOutBuffer != NULL 且 dwOutSize != 0 时用于获取参数配置文件的所需的缓冲区内容。

[返回目录](#)

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

函 数: BOOL NET\_DVR\_GetConfigFile(LONG lUserID, char \*sFileName)

参 数: [in] lUserID                    用户 ID 号，NET\_DVR\_Login\_V40 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](#)

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

函 数: BOOL NET\_DVR\_SetConfigFile\_EX(LONG lUserID, char \*sInBuffer, DWORD dwInSize)

参 数: [in] lUserID                    用户 ID 号，NET\_DVR\_Login\_V40 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](#)

### 5.19.15 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\_V40 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](#)

## 远程重启

### 5.19.16 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\_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](#)



## 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
抓拍机错误码		
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。
NET_DVR_ERR_LACK_IOABLITY	1413	IO 口实际支持的能力不足
NET_DVR_ERR_FTP_PORT	1414	FTP 端口号非法（端口号重复或者异常）
NET_DVR_ERR_FTP_CATALOGUE	1415	FTP 目录名非法（启用多级目录，多级目录传值为空）
NET_DVR_ERR_FTP_UPLOAD_TYPE	1416	FTP 上传类型非法（单 ftp 只支持全部/双 ftp 只支持卡口和违章）
NET_DVR_ERR_FLASH_PARAM_WRITE	1417	配置参数时写 FLASH 失败
NET_DVR_ERR_FLASH_PARAM_READ	1418	配置参数时读 FLASH 失败
NET_DVR_ERR_PICNAME_DELIMITER	1419	FTP 图片命名分隔符非法
NET_DVR_ERR_PICNAME_ITEM	1420	FTP 图片命名项非法（例如 分隔符）
NET_DVR_ERR_PLATE_RECOGNIZE_TYPE	1421	牌识区域类型非法（矩形和多边形有效性校验）
NET_DVR_ERR_CAPTURE_TIMES	1422	抓拍次数非法（有效值是 0~5）
NET_DVR_ERR_LOOP_DISTANCE	1423	线圈距离非法（有效值是 0~2000ms）
NET_DVR_ERR_LOOP_INPUT_STATUS	1424	线圈输入状态非法（有效值）
NET_DVR_ERR_RELATE_IO_CONFLICT	1425	测速组 IO 关联冲突
NET_DVR_ERR_INTERVAL_TIME	1426	连拍间隔时间非法（0~6000ms）
NET_DVR_ERR_SIGN_SPEED	1427	标志限速值非法（大车标志限速不能大于小车标志限速）
NET_DVR_ERR_PIC_FLIP	1428	图像配置翻转（配置交互影响）
NET_DVR_ERR_RELATE_LANE_NUMBER	1429	关联车道数错误(重复 有效值校验 1~99)
NET_DVR_ERR_TRIGGER_MODE	1430	配置抓拍机触发模式非法
NET_DVR_ERR_DELAY_TIME	1431	触发延时时间错误(2000ms)
NET_DVR_ERR_EXCEED_RS485_COUNT	1432	超过最大 485 个数限制
NET_DVR_ERR_RADAR_TYPE	1433	雷达类型错误
NET_DVR_ERR_RADAR_ANGLE	1434	雷达角度错误
NET_DVR_ERR_RADAR_SPEED_VALID_TIME	1435	雷达有效时间错误
NET_DVR_ERR_RADAR_LINE_CORRECT	1436	雷达线性矫正参数错误
NET_DVR_ERR_RADAR_CONST_CORRECT	1437	雷达常量矫正参数错误

NET_DVR_ERR_RECORD_PARAM	1438	录像参数无效（预录时间不超过 10s）
NET_DVR_ERR_LIGHT_WITHOUT_COLOR_AND_DIRECTION	1439	视频检测信号灯配置信号灯个数，但是没有勾选信号灯方向和颜色的
NET_DVR_ERR_LIGHT_WITHOUT_DETECTION_REGION	1440	视频检测信号灯配置信号灯个数，但是没有画检测区域
NET_DVR_ERR_RECOGNIZE_PROVINCE_PARAM	1441	牌识参数省份参数的合法性
NET_DVR_ERR_SPEED_TIMEOUT	1442	IO 测速超时时间非法（有效值大于 0）
NET_DVR_ERR_NTP_TIMEZONE	1443	ntp 时区参数错误
NET_DVR_ERR_NTP_INTERVAL_TIME	1444	ntp 校时间间隔错误
NET_DVR_ERR_NETWORK_CARD_NUM	1445	可配置网卡数目错误
NET_DVR_ERR_DEFAULT_ROUTE	1446	默认路由错误
NET_DVR_ERR_BONDING_WORK_MODE	1447	bonding 网卡工作模式错误
NET_DVR_ERR_SLAVE_CARD	1448	slave 网卡错误
NET_DVR_ERR_PRIMARY_CARD	1449	Primary 网卡错误
NET_DVR_ERR_DHCP_PPOE_WORK	1450	dhcp 和 pppoe 不能同时启动
NET_DVR_ERR_NET_INTERFACE	1451	网络接口错误
NET_DVR_ERR_MTU	1452	MTU 错误
NET_DVR_ERR_NETMASK	1453	子网掩码错误
NET_DVR_ERR_IP_INVALID	1454	IP 地址不合法
NET_DVR_ERR_MULTICAST_IP_INVALID	1455	多播地址不合法
NET_DVR_ERR_GATEWAY_INVALID	1456	网关不合法
NET_DVR_ERR_DNS_INVALID	1457	DNS 不合法
NET_DVR_ERR_ALARMHOST_IP_INVALID	1458	告警主机地址不合法
NET_DVR_ERR_IP_CONFLICT	1459	IP 冲突
NET_DVR_ERR_NETWORK_SEGMENT	1460	IP 不支持同网段
NET_DVR_ERR_NETPORT	1461	端口错误
NET_DVR_ERR_PPPOE_NOSUPPORT	1462	PPPOE 不支持
NET_DVR_ERR_DOMAINNAME_NOSUPPORT	1463	域名不支持
NET_DVR_ERR_NO_SPEED	1464	未启用测速功能
NET_DVR_ERR_IOSTATUS_INVALID	1465	IO 状态错误
NET_DVR_ERR_BURST_INTERVAL_INVALID	1466	连拍间隔非法
NET_DVR_ERR_RESERVE_MODE	1467	备用模式错误
NET_DVR_ERR_LANE_NO	1468	叠加车道号错误
NET_DVR_ERR_COIL_AREA_TYPE	1469	线圈区域类型错误
NET_DVR_ERR_TRIGGER_AREA_PARAM	1470	触发区域参数错误
NET_DVR_ERR_SPEED_LIMIT_PARAM	1471	违章限速参数错误
NET_DVR_ERR_LANE_PROTOCOL_TYPE	1472	车道关联协议类型错误
NET_DVR_ERR_INTERVAL_TYPE	1473	连拍间隔类型非法
NET_DVR_ERR_INTERVAL_DISTANCE	1474	连拍间隔距离非法
NET_DVR_ERR_RS485_ASSOCIATE_DEVTYPE	1475	RS485 关联类型非法
NET_DVR_ERR_RS485_ASSOCIATE_LANENO	1476	RS485 关联车道号非法
NET_DVR_ERR_LANENO_ASSOCIATE_MULTIRS485	1477	车道号关联多个 RS485 口
NET_DVR_ERR_LIGHT_DETECTION_REGION	1478	视频检测信号灯配置信号灯个数，但是检测区域宽或高为 0

NET_DVR_ERR_DN2D_NOSUPPORT	1479	不支持抓拍帧 2D 降噪
NET_DVR_ERR_IRISMODE_NOSUPPORT	1480	不支持的镜头类型
NET_DVR_ERR_WB_NOSUPPORT	1481	不支持的白平衡模式
NET_DVR_ERR_IO_EFFECTIVENESS	1482	IO 口的有效性
NET_DVR_ERR_LIGHTNO_MAX	1483	信号灯检测器接入红/黄灯超限(16)
NET_DVR_ERR_LIGHTNO_CONFLICT	1484	信号灯检测器接入红/黄灯冲突
NET_DVR_ERR_CANCEL_LINE	1485	直行触发线
NET_DVR_ERR_STOP_LINE	1486	待行区停止线
NET_DVR_ERR_RUSH_REDLIGHT_LINE	1487	闯红灯触发线
NET_DVR_ERR_IOOUTNO_MAX	1488	IO 输出口编号越界
NET_DVR_ERR_IOOUTNO_AHEADTIME_MAX	1489	IO 输出口提前时间超限
NET_DVR_ERR_IOOUTNO_IOWORKTIME	1490	IO 输出口有效持续时间超限
NET_DVR_ERR_IOOUTNO_FREQMULTI	1491	IO 输出口脉冲模式下倍频出错
NET_DVR_ERR_IOOUTNO_DUTYRATE	1492	IO 输出口脉冲模式下占空比出错
NET_DVR_ERR_VIDEO_WITH_EXPOSURE	1493	以曝闪起效, 工作方式不支持视频
NET_DVR_ERR_PLATE_BRIGHTNESS_WITHOUT_FLASHDET	1494	车牌亮度自动使能闪光灯仅在车牌亮度补偿模式下起效
NET_DVR_ERR_RECOGNIZE_TYPE_PARAM	1495	识别类型非法 车牌识别参数(如大车、小车、背向、正向、车标识别等)
NET_DVR_ERR_PALTE_RECOGNIZE_AREA_PARAM	1496	牌识参数非法 牌识区域配置时判断出错
NET_DVR_ERR_PORT_CONFLICT	1497	端口有冲突
NET_DVR_ERR_LOOP_IP	1498	IP 不能设置为回环地址
NET_DVR_ERR_DRIVELINE_SENSITIVE	1499	压线灵敏度出错(视频电警模式下)
NET_DVR_ERR_EXCEED_MAX_CAPTURE_TIMES	1600	抓拍模式为频闪时最大抓拍张数为 2 张(IVT 模式下)
NET_DVR_ERR_REDAR_TYPE_CONFLICT	1601	相同 485 口关联雷达类型冲突
NET_DVR_ERR_LICENSE_PLATE_NULL	1602	车牌号为空
NET_DVR_ERR_WRITE_DATABASE	1603	写入数据库失败
NET_DVR_ERR_LICENSE_EFFECTIVE_TIME	1604	车牌有效时间错误
NET_DVR_ERR_PRERECORDED_STARTTIME_LONG	1605	预录开始时间大于违法抓拍张数
NET_DVR_ERR_TRIGGER_RULE_LINE	1606	触发规则线错误
NET_DVR_ERR_LEFTRIGHT_TRIGGERLINE_NOTVERTICAL	1607	左/右触发线不垂直
NET_DVR_ERR_FLASH_LAMP_MODE	1608	闪光灯闪烁模式错误
NET_DVR_ERR_ILLEGAL_SNAPSHOT_NUM	1609	违章抓拍张数错误
NET_DVR_ERR_ILLEGAL_DETECTION_TYPE	1610	违章检测类型错误
NET_DVR_ERR_POSITIVEBACK_TRIGGERLINE_HIGH	1611	正背向触发线高度错误
NET_DVR_ERR_MIXEDMODE_CAPTYPE_ALLTARGETS	1612	混合模式下只支持机非人抓拍类型
NET_DVR_ERR_CARSIGNSPEED_GREATERTHAN_LIMITSPEED	1613	小车标志限速大于限速值
NET_DVR_ERR_BIGCARSIGNSPEED_GREATERTHAN_LIMITSPEED	1614	大车标志限速大于限速值
NET_DVR_ERR_BIGCARSIGNSPEED_GREATERTHAN_CARSIGNSPEED	1615	大车标志限速大于小车标志限速值
NET_DVR_ERR_BIGCARLIMITSPEED_GREATERTHAN_CARLIMITSPEED	1616	大车限速值大于小车限速值
NET_DVR_ERR_BIGCARLOWSPEEDLIMIT_GREATERTHAN_CARLOWSPEEDLIMIT	1617	大车低速限速值大于小车低速限速值
NET_DVR_ERR_CARLIMITSPEED_GREATERTHAN_EXCEPHIGHSPEED	1618	小车限速大于异常高速值
NET_DVR_ERR_BIGCARLIMITSPEED_GREATERTHAN_EXCEPHIGHSPEED	1619	大车限速大于异常高速值

NET\_DVR\_ERR\_STOPLINE\_MORETHAN\_TRIGGERLINE

1620

停止线超过直行触发线

## 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_DESCRIBERECVDATALOST	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_SETUPRECVDATALOST	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_PLAYRECVDATALOST	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_TEARDOWNRECVDATALOST	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	支持的云台轨迹数
MAX_CRUISE_V30	256	支持的云台巡航数
MAX_SERIAL_PORT	8	支持 232 串口数
MAX_FORTIFY_NUM	10	最大布防个数
MAX_DRIVECHAN_NUM	16	最大车道数
MAX_COIL_NUM	3	最大线圈个数
MAX_ETHERNET	2	设备可配以太网网络
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	V3.0 以下版本支持的设备的最大遮挡区域数
MAX_STRINGNUM_V30	8	最大 OSD 字符行数
PATHNAME_LEN	128	路径长度
PICNAME_MAXITEM	15	最大命名项数目
MAX_TIMESEGMENT_V30	8	最大时间段数

MAX_VIDEOOUT_V30	4	视频输出数
MAX_VGA_V30	4	最大可接 VGA 数
MAX_MATRIXOUT	16	最大模拟矩阵输出个数
MAX_DAYS	7	每周的天数
PHONENUMBER_LEN	32	PPPoE 拨号号码最大长度
MAX_DISKNUM_V30	33	最大硬盘数
MAX_NFS_DISK	8	最大 NFS 硬盘数
MAX_LICENSE_LEN	16	车牌号最大长度
MAX_SIGNALLIGHT_NUM	6	最大信号灯个数
MAX_LANERECT_NUM	5	最大车牌识别区域数
MAX_INTERVAL_NUM	4	最大时间间隔个数
MAX_CHJC_NUM	3	最大车辆省份简称字符个数
MAX_VL_NUM	5	最大虚拟线圈个数
MAX_IOSPEED_GROUP_NUM	4	IO 测速组个数
MAX_IOOUT_NUM	4	最大 IO 输出口个数
MAX_IOIN_NUMEX	10	最大 IO 输入口个数(扩展)
MAX_IOIN_NUM	8	最大 IO 输入口个数
MAX_ITC_LANE_NUM	6	最大车道个数
MAX_LIGHT_NUM	6	最大交通灯数
MAX_LANEAREA_NUM	2	单车道最大区域个数
MAX_USERNUM_V30	32	设备的最大用户数
ITC_MAX_POLYGON_POINT_NUM	20	检测区域最多支持 20 个点的多边形
MAX_ITC_SERIALCHECK_NUM	8	串口校验类型个数
MAX_VIDEO_DETECT_LIGHT_NUM	12	视频检测交通信号灯检测区域最大个数
MAX_CABINET_COUNT	8	最大机柜个数
ITS_MAX_DEVICE_NUM	32	最大设备个数
MAX_CUSTOMDIR_LEN	32	自定义目录长度
MAX_ICR_NUM	8	抓拍机红外滤光片预置点数
MAX_ITC_EXCEPTIONOUT	32	抓拍机最大报警输出个数

## 7.2 COUNTRY\_INDEX:国家索引值

```
enum_COUNTRY_INDEX_{
    COUNTRY_NONSUPPORT = 0,    //算法库不支持牌识国家
    COUNTRY_CZE        = 1,    //Czech Republic 捷克共和国
    COUNTRY_FRA        = 2,    //France 法国
    COUNTRY_DEU        = 3,    //Germany 德国
    COUNTRY_ESP        = 4,    //Spain 西班牙
}
```

```

COUNTRY_ITA           = 5,    //Italy 意大利
COUNTRY_NLD           = 6,    //Netherlands 荷兰
COUNTRY_POL           = 7,    //Poland 波兰
COUNTRY_SVK           = 8,    //Slovakia 斯洛伐克
COUNTRY_BLR           = 9,    //Belorussia 白俄罗斯
COUNTRY_MDA           = 10,   //Moldova 摩尔多瓦
COUNTRY_RUS           = 11,   //Russia 俄罗斯
COUNTRY_UKR           = 12,   //Ukraine 乌克兰
COUNTRY_BEL           = 13,   //Belgium 比利时
COUNTRY_BGR           = 14,   //Bulgaria 保加利亚
COUNTRY_DNK           = 15,   //Denmark 丹麦
COUNTRY_FIN           = 16,   //Finland 芬兰
COUNTRY_GBR           = 17,   //Great Britain 英国
COUNTRY_GRC           = 18,   //Greece 希腊
COUNTRY_HRV           = 19,   //Croatia 克罗地亚
COUNTRY_HUN           = 20,   //Hungary 匈牙利
COUNTRY_ISR           = 21,   //Israel 以色列
COUNTRY_LUX           = 22,   //Luxembourg 卢森堡
COUNTRY_MKD           = 23,   //Macedonia 马其顿共和国
COUNTRY_NOR           = 24,   //Norway 挪威
COUNTRY_PRT           = 25,   //Portugal 葡萄牙
COUNTRY_ROU           = 26,   //Romania 多马尼亚
COUNTRY_SRB           = 27,   //Serbia 塞尔维亚
COUNTRY_AZE           = 28,   //Azerbaijan 阿塞拜疆共和国
COUNTRY_GEO           = 29,   //Georgia 即格鲁吉亚
COUNTRY_KAZ           = 30,   //Kazakhstan 哈萨克斯坦
COUNTRY_LTU           = 31,   //Lithuania 立陶宛共和国
COUNTRY_TKM           = 32,   //Turkmenistan 土库曼斯坦
COUNTRY_UZB           = 33,   //Uzbekistan 乌兹别克斯坦
COUNTRY_LVA           = 34,   //Latvia 拉脱维亚
COUNTRY_EST           = 35,   //Estonia 爱沙尼亚
COUNTRY_ALB           = 36,   //Albania 阿尔巴尼亚
COUNTRY_AUT           = 37,   //Austria 奥地利
COUNTRY_BIH           = 38,   //Bosnia and Herzegovina 波斯尼亚和黑塞哥维那
COUNTRY_IRL           = 39,   //Ireland 爱尔兰
COUNTRY_ISL           = 40,   //Iceland 冰岛
COUNTRY_VAT           = 41,   //Vatican 梵蒂冈
COUNTRY_MLT           = 42,   //Malta 马耳他
COUNTRY_SWE           = 43,   //Sweden 瑞典
COUNTRY_CHE           = 44,   //Switzerland 瑞士
COUNTRY_CYP           = 45,   //Cyprus 塞浦路斯
COUNTRY_TUR           = 46,   //Turkey 土耳其
COUNTRY_SVN           = 47,   //Slovenia 斯洛文尼亚
COUNTRY_UNRECOGNIZED = 0xfe, //Unrecognized 无法识别

```

```
COUNTRY_ALL          = 0xff //ALL 全部
}COUNTRY_INDEX
```

### 7.3 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++。

### 7.4 NET\_DVR\_ALARM\_RADARINFO:雷达报警报警信息

```
struct{
    DWORD    dwSize;
    DWORD    dwRadarTriggerTimeSecond;
    DWORD    dwRadarTriggerTimeMSecond;
    DWORD    dwVedioTriggerTimeSecond;
    DWORD    dwVedioTriggerTimeMSecond;
    DWORD    dwVedioRadarDiffTimeMSecond;
    DWORD    dwRadarSpeed;
    BYTE     byRes[16];
}NET_DVR_ALARM_RADARINFO,*LPNET_DVR_ALARM_RADARINFO;
```

#### Members

##### *dwSize*

结构体大小

`dwRadarTriggerTimeSecond`

雷达触发时间：秒

`dwRadarTriggerTimeMSecond`

雷达触发时间：毫秒

`dwVedioTriggerTimeSecond`

视频触发时间：秒

`dwVedioTriggerTimeMSecond`

视频触发时间：毫秒

`dwVedioRadarDiffTimeMSecond`

雷达视频触发时间差：毫秒

`dwRadarSpeed`

雷达速度

`byRes`

保留

#### Remarks

该报警在三级布防下才会产生，布防等级在布防（[NET\\_DVR\\_SetupAlarmChan\\_V41](#)）时选择。

## 7.5 NET\_DVR\_ALARMCTRL\_CFG:报警控制配置

```
struct{
    DWORD    dwSize;
    BYTE     byListenPicUploadEnabled;
    BYTE     byRes[259];
}NET_DVR_ALARMCTRL_CFG, *LPNET_DVR_ALARMCTRL_CFG;
```

#### Members

`dwSize`

结构体大小

`byListenPicUploadEnabled`

监听图片上传使能：0- 禁用，1- 启用

`byRes`

保留，置为 0

#### Remarks

- 设备是否支持报警控制配置或者支持的参数能力，可以通过设备能力集进行判断，对应报警控制能力集([AlarmCtrlParam](#))，相关接口：[NET\\_DVR\\_GetSTDAbility](#)，能力集类型：[NET\\_DVR\\_GET\\_ALARMCTRL\\_CAPABILITIES](#)。

## 7.6 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\_V40 的返回值，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

## 7.7 NET\_DVR\_ALARMINFO\_V30:上传的报警信息

```

struct{
    DWORD    dwAlarmType;
    DWORD    dwAlarmInputNumber;
    BYTE     byAlarmOutputNumber[MAX_ALARMOUT_V30];
    BYTE     byAlarmRelateChannel[MAX_CHANNUM_V30];
    BYTE     byChannel[MAX_CHANNUM_V30];
    BYTE     byDiskNumber[MAX_DISKNUM_V30];
}NET_DVR_ALARMINFO_V30,*LPNET_DVR_ALARMINFO_V30;

```

### Members

*dwAlarmType*

报警类型：0-信号量报警，1-硬盘满，2-信号丢失，3-移动侦测，4-硬盘未格式化，5-读写硬盘出错，6-遮挡报警，7-制式不匹配，8-非法访问，9-视频信号异常，10-录像/抓图异常，11-智能场景变化，12-阵列异常，13-前端/录像分辨率不匹配，17-闪光灯异常

*dwAlarmInputNumber*

报警输入端口

*byAlarmOutputNumber*

触发的报警输出端口，值为 1 表示该报警端口输出，如 `byAlarmOutputNumber[0]=1` 表示触发第 1 个报警输出口输出，`byAlarmOutputNumber[1]=1` 表示触发第 2 个报警输出口，依次类推。

*byAlarmRelateChannel*

触发的录像通道，值为 1 表示该通道录像，如 `byAlarmRelateChannel[0]=1` 表示触发第 1 个通道录像

*byChannel*

发生报警的通道。当报警类型为 2、3、6、9、10、11、13、17 时有效，如 `byChannel[0]=1` 表示第 1 个通道报警

*byDiskNumber*

发生报警的硬盘。当报警类型为 1、4、5 时有效，`byDiskNumber[0]=1` 表示 1 号硬盘异常

### Remarks

抓拍机支持类型：1-硬盘满、5-读写硬盘出错、8-非法访问，这里的硬盘即指 SD 卡。

## 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:0-off、1-UP、2-DOWN、3-LEFT、4-RIGHT、5-MIDDLE、6-自定义

#### *byBacklightLevel*

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\_BARRIERGATE\_CFG:道闸控制参数

```
struct{
    DWORD       dwSize;
    DWORD       dwChannel;
    BYTE        byLaneNo;
    BYTE        byBarrierGateCtrl;
    BYTE        byRes[14];
}NET_DVR_BARRIERGATE_CFG,*LPNET_DVR_BARRIERGATE_CFG;
```

### Members

#### *dwSize*

结构体大小

#### *dwChannel*



通道号

*byLaneNo*

道闸号：0- 表示无效值(设备需要做有效值判断)，1- 道闸 1

*byBarrierGateCtrl*

控制参数：0- 关闭道闸，1- 开启道闸，2- 停止道闸，3- 锁定道闸

*byRes*

保留，置为 0

## 7.10 NET\_DVR\_BARRIERGATE\_COND:出入口控制条件

```
struct{
    BYTE    byLaneNo;
    BYTE    byRes[3];
}NET_DVR_BARRIERGATE_COND,*LPNET_DVR_BARRIERGATE_COND;
```

### Members

*byLaneNo*

车道号：0- 表示无效值(设备需要做有效值判断)，1- 车道 1

*byRes*

保留，置为 0

## 7.11 NET\_DVR\_CABINET\_ALARM\_PARAMCFG:机柜报警参数

```
struct{
    char    sCabinetName[NAME_LEN];
    BYTE    byAssociateIO;
    BYTE    byCabinetState;
    BYTE    byAlarmIntervalTime;
    BYTE    byRes1[25];
}NET_DVR_CABINET_ALARM_PARAMCFG,*LPNET_DVR_CABINET_ALARM_PARAMCFG;
```

### Members

*sCabinetName*

机柜名称

*byAssociateIO*

关联的 IO 口号，取值范围：1~4 表示具体的 IO 通道号，0xff 表示不启用

*byCabinetState*

机柜触发报警状态：0- 低电平，1- 高电平

*byAlarmIntervalTime*

报警上传时间间隔，取值范围：1~60，单位：秒

*byRes1*

保留，置为 0

## 7.12 NET\_DVR\_CABINET\_CFG:机柜配置参数

```
struct{
    DWORD                dwSize;
    BYTE                 byEnable;
    BYTE                 byRes[3];
    NET\_DVR\_CABINET\_ALARM\_PARAMCFG struCabinetCfg[MAX_CABINET_COUNT];
    BYTE                 byRes1[84];
}NET_DVR_CABINET_CFG,*LPNET_DVR_CABINET_CFG;
```

### Members

#### *dwSize*

结构体大小

#### *byEnable*

使能：0- 禁用，1- 启用

#### *byRes*

保留，置为 0

#### *struCabinetCfg*

机柜报警参数配置

#### *byRes1*

保留，置为 0

## 7.13 NET\_DVR\_CALIBRATE\_TIME:时间校时参数

```
struct{
    DWORD                dwSize;
    NET\_DVR\_TIME        struTime;
    WORD                 wMilliSec;
    BYTE                 byRes[14];
}NET_DVR_CALIBRATE_TIME,*LPNET_DVR_CALIBRATE_TIME;
```

### Members

#### *dwSize*

结构体大小

#### *struTime*

时间参数，年月日时分秒

#### *wMilliSec*

毫秒

#### *byRes*

保留，置为 0

## 7.14 NET\_DVR\_CAMERAPARAMCFG:前端参数配置

```
struct{
```

DWORD	dwSize;
<a href="#">NET_DVR_VIDEOEFFECT</a>	struVideoEffect;
<a href="#">NET_DVR_GAIN</a>	struGain;
<a href="#">NET_DVR_WHITEBALANCE</a>	struWhiteBalance;
<a href="#">NET_DVR_EXPOSURE</a>	struExposure;
<a href="#">NET_DVR_GAMMACORRECT</a>	struGammaCorrect;
<a href="#">NET_DVR_WDR</a>	struWdr;
<a href="#">NET_DVR_DAYNIGHT</a>	struDayNight;
<a href="#">NET_DVR_BACKLIGHT</a>	struBackLight;
<a href="#">NET_DVR_NOISEREMOVE</a>	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;
<a href="#">NET_DVR_CMOSMODECFG</a>	struCmosModeCfg;
BYTE	byFilterSwitch;
BYTE	byFocusSpeed;
BYTE	byAutoCompensationInterval;
BYTE	bySceneMode;

}NET\_DVR\_CAMERAPARAMCFG, \*LPNET\_DVR\_CAMERAPARAMCFG;

### 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

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。对于 V3.5 或以上版本的智能交通摄像机，该参数无效，相关功能通过使能参数（[NET\\_DVR\\_SNAPENABLECFG](#)）的 `wJpegPicSize` 实现。

*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

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

```
struct{
    DWORD                dwSize;
    NET\_DVR\_VIDEOEFFECT  struVideoEffect;
    NET\_DVR\_GAIN         struGain;
    NET\_DVR\_WHITEBALANCE struWhiteBalance;
    NET\_DVR\_EXPOSURE    struExposure;
    NET\_DVR\_GAMMACORRECT struGammaCorrect;
    NET\_DVR\_WDR         struWdr;
    NET\_DVR\_DAYNIGHT    struDayNight;
    NET\_DVR\_BACKLIGHT   struBackLight;
    NET\_DVR\_NOISEREMOVE 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          byCaptureModeN;
BYTE          byCaptureModeP;
NET\_DVR\_SMARTIR\_PARAM      struSmartIRParam;
NET\_DVR\_PIRIS\_PARAM       struPIrisParam;
NET\_DVR\_LASER\_PARAM\_CFG    struLaserParam;
NET_DVR_FFC_PARAM    struFFCParam;
NET_DVR_DDE_PARAM    struDDEParam;
NET_DVR_AGC_PARAM    struAGCParam;
BYTE          byLensDistortionCorrection;
BYTE          byRes1[3];
NET\_DVR\_SNAP\_CAMERAPARAMCFG struSnapCCD;
BYTE          byRes2[188];
}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。对于 V3.5 或以上版本的智能交通摄像机，该参数无效，相关功能通过使能参数（[NET\\_DVR\\_SNAPENABLECFG](#)）的 `wJpegPicSize` 实现。

*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]

*byCaptureModeN*

视频输入模式（N 制）：

0-关闭，1-640\*480@25fps，2-640\*480@30ps，3-704\*576@25fps，4-704\*480@30fps，  
5-1280\*720@25fps，6-1280\*720@30fps，7-1280\*720@50fps，8-1280\*720@60fps，9-1280\*960@15fps，  
10-1280\*960@25fps，11-1280\*960@30fps，12-1280\*1024@25fps、13--1280\*1024@30fps，  
14-1600\*900@15fps，15-1600\*1200@15fps，16-1920\*1080@15fps，17-1920\*1080@25fps，  
18-1920\*1080@30fps，19-1920\*1080@50fps，20-1920\*1080@60fps，21-2048\*1536@15fps，  
22-2048\*1536@20fps，23-2048\*1536@24fps，24-2048\*1536@25fps，25-2048\*1536@30fps，  
26-2560\*2048@25fps，27-2560\*2048@30fps，28-2560\*1920@7.5fps，29-3072\*2048@25fps，  
30-3072\*2048@30fps，31-2048\*1536@12.5fps，32-2560\*1920@6.25fps，33-1600\*1200@25fps，  
34-1600\*1200@30fps，35-1600\*1200@12.5fps，36-1600\*900@12.5fps，37-1600@900@15fps，  
38-800\*600@25fps，39-800\*600@30fps，40-4000\*3000@12.5fps，41-4000\*3000@15fps，  
42-4096\*2160@20fps，43-3840\*2160@20fps，44-960\*576@25fps，45-960\*480@30fps，



46-752\*582@25fps, 47-768\*494@30fps, 48-2560\*1440@25fps, 49-2560\*1440@30fps,  
 50-720P@100fps, 51-720P@120fps, 52-2048\*1536@50fps, 53-2048\*1536@60fps,  
 54-3840\*2160@25fps, 55-3840\*2160@30fps, 56-4096\*2160@25fps, 57-4096\*2160@30fps ,  
 58-1280\*1024@50fps, 59-1280\*1024@60fps, 60-3072\*2048@50fps, 61-3072\*2048@60fps,  
 62-3072\*1728@25fps, 63-3072\*1728@30fps, 64-3072\*1728@50fps, 65-3072\*1728@60fps,  
 66-336\*256@50fps, 67-336\*256@60fps, 68-384\*288@50fps, 69-384\*288@60fps, 70-640\*512@50fps,  
 71-640\*512@60fps, 72-2592\*1944@25fps, 73-2592\*1944@30fps, 74-2688\*1536@25fps,  
 75-2688\*1536@30fps, 76-2592\*1944@20fps, 77-2592\*1944@15fps, 78-2688\*1520@20fps,  
 79-2688\*1520@15fps, 80-2688\*1520@25fps, 81-2688\*1520@30fps, 82-2720\*2048@25fps,  
 83-2720\*2048@30fps, 84-336\*256@25fps, 85-384\*288@25fps, 86-640\*512@25fps,  
 87-1280\*960@50fps, 88-1280\*960@60fps, 89-1280\*960@100fps, 90-1280\*960@120fps,  
 91-4000\*3000@20fps

*byCaptureModeP*

视频输入模式（P 制），取值同 *byCaptureModeN*

*struSmartIRParam*

红外过曝配置信息

*struIrisParam*

P-Iris 红外光圈大小等级配置信息

*struLaserParam*

激光参数

*struFFCParam*

FFC 参数，保留

*struDDEParam*

DDE 参数，保留

*struAGCParam*

AGC 参数，保留

*byLensDistortionCorrection*

镜头畸变校正：0- 关闭，1- 开启

*byRes1*

保留，置为 0

*struSnapCCD*

抓拍机 CCD 参数，只用于智能交通摄像机(抓拍机)

*byRes*

保留，置为 0

## 7.16 NET\_DVR\_CAMERA\_SETUPCFG:相机架设参数

struct{

```

  DWORD   dwSize;
  WORD    wSetupHeight;
  BYTE    byLensType;
  BYTE    bySetupHeightUnit;
  DWORD   dwSceneDis;
  float   fPitchAngle;
```

```

float      fInclineAngle;
float      fRotateAngle;
WORD       wVideoDetCoefficient;
BYTE       byErectMethod;
BYTE       byCameraViewAngle;
DWORD      dwHorizontalDistance;
BYTE       byDetailLensType;
BYTE       byRes[3];
float      fHorFieldAngle;
float      fVerFieldAngle;
BYTE       byRes1[24];
}NET_DVR_CAMERA_SETUPCFG,*LPNET_DVR_CAMERA_SETUPCFG;

```

### Members

#### *dwSize*

结构体大小

#### *wSetupHeight*

架设高度，取值范围：[0,20]

#### *byLensType*

镜头焦距类型：0-未知,1-8mm,2-12mm,3-16mm,4-25mm,5-35mm,6-50mm

#### *bySetupHeightUnit*

架设高度单位：0- 米(m)，1- 厘米(cm)

#### *dwSceneDis*

图像下边缘位置与相机之间的水平距离，单位：厘米

#### *fPitchAngle*

相机俯仰角度（暂不支持），取值范围：[-180, 180]，单位：度

#### *fInclineAngle*

相机倾斜角度（暂不支持），取值范围：[-180, 180]，单位：度

#### *fRotateAngle*

相机旋转角度（暂不支持），取值范围：[-180, 180]，单位：度

#### *wVideoDetCoefficient*

视频测试系数，取值范围：[0,300]

#### *byErectMethod*

架设方式：0-正装，1-侧装

#### *byCameraViewAngle*

摄像机安装视野角度：0- 垂直（默认），1- 倾斜

#### *dwHorizontalDistance*

摄像机与出入口水平距离，取值范围：1~1000，默认：30，单位：cm，摄像机角度为倾斜时有效

#### *byDetailLensType*

镜头类型，取值范围：1~100，默认：28，单位：0.1mm

#### *byRes*

保留

#### *fHorFieldAngle*

摄像机水平视场角角度，取值范围：[0,360]，单位：度

#### *fVerFieldAngle*

摄像机垂直视场角角度，取值范围：[0,360]，单位：度

*byRes1*

保留

## 7.17 NET\_DVR\_CERT\_INFO:证书信息

```
struct{
    DWORD                dwSize;
    NET_DVR_CERT_PARAM  struCertParam;
    DWORD                dwValidDays;
    BYTE                 byPasswd[NAME_LEN];
    NET_DVR_CERT_NAME   struCertName;
    NET_DVR_CERT_NAME   strulssuerName;
    NET_DVR_TIME_EX     struBeginTime;
    NET_DVR_TIME_EX     struEndTime;
    BYTE                 serialNumber[NAME_LEN];
    BYTE                 byVersion;
    BYTE                 byKeyAlgorithm;
    BYTE                 byKeyLen;
    BYTE                 bySignatureAlgorithm;
    BYTE                 byRes[128];
}NET_DVR_CERT_INFO,*LPNET_DVR_CERT_INFO;
```

### Members

*dwSize*

结构体大小

*struCertParam*

证书参数

*dwValidDays*

有效天数，类型为自签名时有效

*byPasswd*

私钥密码

*struCertName*

证书名称

*strulssuerName*

证书发行者名称（自签名证书信息获取时有效）

*struBeginTime*

证书创建时间（自签名证书信息获取时有效）

*struEndTime*

证书截止时间（自签名证书信息获取时有效）

*serialNumber*

证书标识码（自签名证书信息获取时有效）

*byVersion*

证书版本

*byKeyAlgorithm*

加密类型：0- RSA， 1- DSA

*byKeyLen*

加密长度：0- 512， 1- 1024， 2- 2048

*bySignatureAlgorithm*

签名算法类型（自签名证书信息获取时有效）

*byRes*

保留，置为 0

## 7.18 NET\_DVR\_CERT\_NAME:证书名称信息

```
struct{
    BYTE    byCountry[MAX_COUNTRY_NAME_LEN];
    BYTE    byState[MAX_DOMAIN_NAME];
    BYTE    byLocality[MAX_DOMAIN_NAME];
    BYTE    byOrganization[MAX_DOMAIN_NAME];
    BYTE    byUnit[MAX_DOMAIN_NAME];
    BYTE    byCommonName[MAX_DOMAIN_NAME];
    BYTE    byEmail[MAX_DOMAIN_NAME];
    BYTE    byRes[128];
}NET_DVR_CERT_NAME,*LPNET_DVR_CERT_NAME;
```

### Members

*byCountry*

国家简写，如：CN

*byState*

洲或省

*byLocality*

地区

*byOrganization*

组织

*byUnit*

单位

*byCommonName*

域名或者 IP 地址

*byEmail*

Email 地址

*byRes*

保留，置为 0

## 7.19 NET\_DVR\_CERT\_PARAM:证书参数

```
struct{
    DWORD    dwSize;
    WORD     wCertFunc;
```

```

WORD    wCertType;
BYTE    byFileType;
BYTE    byRes[35];
}NET_DVR_CERT_PARAM, *LPNET_DVR_CERT_PARAM;

```

### Members

#### *dwSize*

结构体大小

#### *wCertFunc*

证书种类：0- 802.1x， 1- HTTPS， 2- EAP-TLS

#### *wCertType*

证书类型：0- CA， 1- Certificate， 2- 私钥文件

#### *byFileType*

证书文件类型：0- PEM， 1- PFX

#### *byRes*

保留，置为 0

## 7.20 NET\_DVR\_CLIENTINFO:预览参数

```

struct{
    LONG    IChannel;
    LONG    ILinkMode;
    HWND    hPlayWnd;
    char    *sMultiCastIP;
}NET_DVR_CLIENTINFO, *LPNET_DVR_CLIENTINFO;

```

### Members

#### *IChannel*

通道号，对于摄像机，只有一个通道，通道号为 1

#### *ILinkMode*

最高位(31)为 0 表示主码流，为 1 表示子码流；0~30 位表示连接方式：0- TCP 方式，1- UDP 方式，2- 多播方式，3- RTP 方式，4- RTP over RTSP，5- RTSP over HTTP

例如子码流 TCP 连接，则 ILinkMode=0x80000000

#### *hPlayWnd*

播放窗口的句柄，为 NULL 表示不显示图像

#### *sMultiCastIP*

多播组地址

### Remarks:

Windows 下该结构中的 hPlayWnd 参数若设置为 NULL，则 SDK 仍取流，但不进行解码显示，所以仍可录像。Linux 下该结构中的 hPlayWnd = {0}则 SDK 仍取流，不进行解码显示，所以仍可录像，但是不能设置 hPlayWnd = 0(即 NULL)，否则非法结构地址会导致调用 hPlayWnd.x 等去判断的时候崩溃。

当客户选择普通协议传输时，若设备支持 RTSP 协议，SDK 将直接转成 RTSP 方式连接设备。

### Linux 下

HWND 表示播放窗口的句柄，定义为：

```
typedef unsigned int HWND;
```

如果使用 Qt 进行界面开发，示例如下：

```
NET_DVR_CLIENTINFO tmpclientinfo;
tmpclientinfo.hPlayWnd = (HWND)m_framePlayWnd->GetPlayWndId();
```

## 7.21 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

## 7.22 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

## 7.23 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

## 7.24 NET\_DVR\_COLOR:图像参数

```

struct{
    BYTE     byBrightness;
    BYTE     byContrast;

```

```

    BYTE    bySaturation;
    BYTE    byHue;
}NET_DVR_COLOR, *LPNET_DVR_COLOR;

```

### Members

#### *byBrightness*

亮度，取值范围[0,255]

#### *byContrast*

对比度，取值范围[0,255]

#### *bySaturation*

饱和度，取值范围[0,255]

#### *byHue*

色调，取值范围[0,255]

## 7.25 NET\_DVR\_COMPLETE\_RESTORE\_INFO:完全恢复出厂值控制参数

```

struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    BYTE     byRes[64];
}NET_DVR_COMPLETE_RESTORE_INFO,*LPNET_DVR_COMPLETE_RESTORE_INFO;

```

### Members

#### *dwSize*

结构体大小

#### *dwChannel*

设备通道号，目前该参数无效

#### *byRes*

保留，置为 0

### Remarks

远程控制设备完全恢复出厂默认值，包括设备 IP 地址会恢复成 192.0.0.64。只有 admin 管理员账号才支持该功能。

完全恢复之后设备会恢复为未激活状态，调用 [NET\\_DVR\\_ActivateDevice](#) 可以激活设备。

## 7.26 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）

## 7.27 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\*900，60-2752\*2208，0xff-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。

## 7.28 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。

## 7.29 NET\_DVR\_CURTRIGGERMODE:当前触发模式

```
struct{
    DWORD    dwSize;
    DWORD    dwTriggerType;
    BYTE     byRes[24];
}NET_DVR_CURTRIGGERMODE,*LPNET_DVR_CURTRIGGERMODE;
```

### Members

*dwSize*

结构体大小

*dwTriggerType*

触发类型，详见 [ITC\\_TRIGGERMODE\\_TYPE](#)

*byRes*

保留

## 7.30 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

### 7.31 NET\_DVR\_DDNSPARA\_V30:网络应用参数（DDNS）

```

struct{
    BYTE    byEnableDDNS;
    BYTE    byHostIndex;
    BYTE    byRes1[2];
    struct{
        BYTE    sUserName[NAME_LEN];
        BYTE    sPassword[PASSWD_LEN];
        BYTE    sDomainName[MAX_DOMAIN_NAME];
        BYTE    sServerName[MAX_DOMAIN_NAME];
        WORD    wDDNSPort;
        BYTE    byRes[10];
    }struDDNS[MAX_DDNS_NUMS];
    BYTE    byRes2[16];
}NET_DVR_DDNSPARA_V30,*LPNET_DVR_DDNSPARA_V30;

```

#### Members

*byEnableDDNS*

是否使能：0- 否，1- 是

*byHostIndex*

0- Private DNS，1- Dyndns，2- PeanutHull(花生壳)，3- NO-IP，4- hiDDNS

*byRes1*

保留，置为 0

*sUsername*

DDNS 账号用户名

*sPassword*

DDNS 账号密码

*sDomainName*

域名

*sServerName*

DDNS 对应的服务器地址，可以是 IP 地址或域名

*wDDNSPort*

DDNS 端口

*byRes*

保留，置为 0

*byRes2*

保留，置为 0

## 7.32 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*

轨迹是否设置：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。

**7.33 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

**7.34 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

*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

*byPCPlug*

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。

byDVRType 和 wDevType 取值定义如下所示:

宏定义	宏定义值	设备类型
ITCCAM	35	智能交通摄像机

## 7.35 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，详见下文列表

#### *byChanNum*

设备模拟通道个数，数字（IP）通道最大个数通过 NET\_DVR\_GetDVRConfig（配置命令 NET\_DVR\_GET\_IPPARACFG\_V40）获取（dwDChanNum）。

#### *byStartChan*

模拟通道的起始通道号，目前设备模拟通道号从 1 开始，数字通道的起始通道号通过 NET\_DVR\_GetDVRConfig（配置命令 NET\_DVR\_GET\_IPPARACFG\_V40）获取（dwStartDChan）。

#### *byAudioChanNum*

设备语音通道数

#### *byIPChanNum*

设备最大数字通道个数，低 8 位，高 8 位见 byHighDChanNum。可以根据 IP 通道个数来判断是否调用 NET\_DVR\_GetDVRConfig（配置命令 NET\_DVR\_GET\_IPPARACFG\_V40）获取模拟和数字通道相关参数（NET\_DVR\_IPPARACFG\_V40）。

#### *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, 详见下文列表

**bySupport3**

能力集扩展, 位与结果为 0 表示不支持, 1 表示支持  
 bySupport3 & 0x1, 表示是否支持多码流

**byMultiStreamProto**

多码流是否支持 rtsp 协议取流, 按位表示, 0-不支持, 1-支持: bit0- 码流 3, bit1- 码流 4, 依次类推

**byStartDChan**

起始数字通道号, 0 表示无效

**byStartDTalkChan**

起始数字对讲通道号, 区别于模拟对讲通道号, 0 表示无效

**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。

byDVRTYPE 和 wDevType 取值定义如下所示:

宏定义	宏定义值	设备类型
ITCCAM	35	智能交通摄像机

## 7.36 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++。

## 7.37 NET\_DVR\_DEVICESTATECFG:设备状态参数

```

struct{
    DWORD          dwSize;
    WORD           wPreviewNum;
    WORD           wFortifyLinkNum;
    NET_DVR_IPADDR struPreviewIP[MAX_LINK];
    NET_DVR_IPADDR struFortifyIP[MAX_FORTIFY_NUM];
    DWORD          dwVideoFrameRate;
    BYTE           byResolution;
    BYTE           bySnapResolution;
    BYTE           byStreamType;
    BYTE           byTriggerType;
    DWORD          dwSDVolume;
    DWORD          dwSDFreeSpace;
    BYTE           byDetectorState[MAX_DRIVECHAN_NUM][MAX_COIL_NUM];
    BYTE           byDetectorLinkState;
    BYTE           bySDStatus;
    BYTE           byFortifyLevel[MAX_FORTIFY_NUM];
    BYTE           byRes2[116];
}NET_DVR_DEVICESTATECFG, *LPNET_DVR_DEVICESTATECFG;

```

### Members

#### *dwSize*

大小

#### *wPreviewNum*

预览连接个数

#### *wFortifyLinkNum*

布防连接个数

#### *struPreviewIP*

预览的用户 IP 地址

#### *struFortifyIP*

布防连接的用户 IP 地址

#### *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；

#### *byResolution*

视频分辨率：0-DCIF 1-CIF, 2-QCIF, 3-4CIF, 4-2CIF, 16-VGA(640\*480), 17-UXGA(1600\*1200), 18-SVGA (800\*600),19-HD720p(1280\*720),20-XVGA, 21-HD900p, 27-HD1080i, 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, 38-2336\*1744, 39-1940\*1460, 0xff-Auto(使用当前码流分辨率)

#### *bySnapResolution*

抓拍分辨率：0-DCIF 1-CIF, 2-QCIF, 3-4CIF, 4-2CIF, 16-VGA(640\*480), 17-UXGA(1600\*1200), 18-SVGA

(800\*600),19-HD720p(1280\*720), 20-XVGA, 21-HD900p, 27-HD1080i, 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, 38-2336\*1744, 39-1940\*1460 0xff-Auto(使用当前码流分辨率)

#### *byStreamType*

码流类型: 0- 主码流, 1- 子码流

#### *byTriggerType*

触发模式: 0- 视频触发, 1- 普通触发

#### *dwSDVolume*

SD 卡容量

#### *dwSDFreeSpace*

SD 卡剩余空间

#### *byDetectorState*

车道的线圈状态: 0- 未使用, 1- 正常, 2- 异常

#### *byDetectorLinkState*

车检器连接状态: 0- 未连接, 1- 连接

#### *bySDStatus*

SD 卡状态: 0- 活动, 1- 休眠, 2- 异常, 3- 无 sd 卡

#### *byFortifyLevel*

布防等级: 0- 一等级 (高), 1- 二等级 (中), 2- 三等级 (低), 0xff- 无

#### *byRes2*

保留, 置为 0

#### Remarks

该中参数 *byFortifyLevel* 和 *struFortifyIP* 是一一对应的

## 7.38 NET\_DVR\_DISK\_QUOTA:磁盘配额信息

```
struct{
    BYTE        byQuotaType;
    BYTE        byRes1[5];
    WORD        wStoragePeriod;
    DWORD       dwHCapacity;
    DWORD       dwLCapacity;
    DWORD       dwHUsedSpace;
    DWORD       dwLUsedSpace;
    BYTE        byQuotaRatio;
    BYTE        byRes2[21];
}NET_DVR_DISK_QUOTA,*LPNET_DVR_DISK_QUOTA;
```

#### Members

##### *byQuotaType*

磁盘配额类型: 1- 按容量, 2- 按比例, 3- 按时间

##### *byRes1*

保留

##### *wStoragePeriod*

录像存储周期, 单位: 天, 配额类型为按时间时有效

***dwHCapacity***

分配的磁盘容量高 32 位，单位 MB

***dwLCapacity***

分配的磁盘容量低 32 位，单位 MB

***dwHUsedSpace***

已使用的磁盘空间高 32 位，单位 MB

***dwLUsedSpace***

已使用的磁盘空间低 32 位，单位 MB

***byQuotaRatio***

分配的磁盘比例，单位：%

***byRes2***

保留

**7.39 NET\_DVR\_DISK\_QUOTA\_CFG:磁盘配额配置**

```

struct{
    DWORD                dwSize;
    NET\_DVR\_DISK\_QUOTA  struPicQuota;
    NET\_DVR\_DISK\_QUOTA  struRecordQuota;
    BYTE                 res2[60];
}NET_DVR_DISK_QUOTA_CFG,*LPNET_DVR_DISK_QUOTA_CFG;

```

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

结构体大小

***struPicQuota***

图片配额

***struRecordQuota***

录像配额

***res2***

保留，置为 0

**7.40 NET\_DVR\_DOWNLOAD\_ALARM\_RECORD\_COND:报警录像下载条件**

```

struct{
    DWORD    dwSize;
    WORD     wAlarmRecordID;
    BYTE     byRes[130];
}NET_DVR_DOWNLOAD_ALARM_RECORD_COND,*LPNET_DVR_DOWNLOAD_ALARM_RECORD_COND;

```

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

结构体大小

*wAlarmRecordID*

报警录像 ID

*byRes*

保留，置为 0

## 7.41 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

## 7.42 NET\_DVR\_ENTRANCE\_CFG: 出入口控制参数

```
struct{
    DWORD    dwSize;
    BYTE    byLaneNo;
    BYTE    byBarrierGateCtrlMode;
    BYTE    byRes1[2];
    DWORD    dwRelateTriggerMode;
    DWORD    dwMatchContent;
    NET\_DVR\_RELAY\_PARAM    struRelayRelateInfo[MAX_RELAY_NUM];
    BYTE    byGateSingleIO[MAX_IOIN_NUM];
    NET\_DVR\_VEHICLE\_CONTROL    struVehicleCtrl[MAX_VEHICLE_TYPE_NUM];
    BYTE    byRes2[64];
}NET_DVR_ENTRANCE_CFG, *LPNET_DVR_ENTRANCE_CFG;
```

### Members

*dwSize*

结构体大小

*byLaneNo*

使能：0- 关闭，1- 打开

*byBarrierGateCtrlMode*

道闸控制模式：0- 相机自动控制，1- 平台外部控制

*byRes1*

保留，置为 0



***dwRelateTriggerMode***

触发模式，参见 [ITC\\_TRIGGERMODE\\_TYPE](#)

***dwMatchContent***

异常处理，不同异常处理方式的"或"结果：

0x00: 不做匹配

0x01: 车牌匹配（带车牌颜色）

0x02: 车牌匹配（不带车牌颜色）

0x04: 卡号匹配

***struRelayRelateInfo***

继电器关联配置信息(一个继电器只能控制一个接入设备)，数组 0 表示继电器 1，数组 1 表示继电器 2，依次类推

***byGateSingleIO***

单个 IO 触发参数，数组 0 表示 IO1，数组 1 表示 IO2，依次类推，具体数组值的含义：0- 无，1- 道闸开到到位，2- 道闸关到位，3- 消防报警

***struVehicleCtrl***

车辆信息管控：

数组 0- 临时车辆配置的模式

数组 1- 禁止名单车辆的模式

数组 2- 允许名单车辆的模式

***byRes2***

保留，置为 0

## 7.43 NET\_DVR\_ETHERNET\_MULTI:单个网卡配置信息

```
struct{
    NET\_DVR\_IPADDR    struDVRIP;
    NET\_DVR\_IPADDR    struDVRIPMask;
    DWORD             dwNetInterface;
    BYTE              byCardType;
    BYTE              byRes1;
    WORD              wMTU;
    BYTE              byMACAddr[MACADDR_LEN];
    BYTE              byRes2[2];
    BYTE              byUseDhcp;
    BYTE              byRes3[3];
    NET\_DVR\_IPADDR    struGatewayIpAddr;
    NET\_DVR\_IPADDR    struDnsServer1IpAddr;
    NET\_DVR\_IPADDR    struDnsServer2IpAddr;
}NET_DVR_ETHERNET_MULTI, *LPNET_DVR_ETHERNET_MULTI;
```

### 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

***byCardType***

网卡类型: 0-普通网卡, 1-内网网卡, 2-外网网卡

***byRes1***

保留

***wMTU***

MTU 设置, 默认 1500

***byMACAddr***

设备物理地址, 只读

***byRes2***

保留

***byUseDhcp***

是否启用 DHCP

***byRes3***

保留

***struGatewayIpAddr***

网关地址

***struDnsServer1IpAddr***

域名服务器 1 的 IP 地址

***struDnsServer2IpAddr***

域名服务器 2 的 IP 地址

**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。

## 7.44 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。

## 7.45 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

## 7.46 NET\_DVR\_EXDEVDET\_CFG:外设状态信息

```
struct{
    DWORD dwSize;
    BYTE byExternalDevStatus;
    BYTE byRes[63];
}NET_DVR_EXDEVDET_CFG,*LPNET_DVR_EXDEVDET_CFG;
```

### Members

*dwSize*

结构体大小

*byExternalDevStatus*

外设状态：0- 正常，1- 异常

*byRes*

保留，置为 0

## 7.47 NET\_DVR\_EXDEVDET\_COND: 外设状态检测条件

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    BYTE     byExternalDevType;
    BYTE     byRes[63];
}NET_DVR_EXDEVDET_COND,*LPNET_DVR_EXDEVDET_COND;
```

### Members

*dwSize*

结构体大小

*dwChannel*

通道号

*byExternalDevType*

外接设备类型：0- 无效值（保留），1- 信号灯，2- 闪光灯

*byRes*

保留，置为 0

## 7.48 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

## 7.49 NET\_DVR\_FIND\_PICTURE: 图片查找结果信息

```
struct{
    char                sFileName[PICTURE_NAME];
    NET\_DVR\_TIME      struTime;
    DWORD              dwFileSize;
    char                sCardNum[CARDNUM_LEN_V30];
    BYTE                byPlateColor;
    BYTE                byVehicleLogo;
    BYTE                byRes[30];
}NET_DVR_FIND_PICTURE,*LPNET_DVR_FIND_PICTURE;
```

### Members

#### *sFileName*

图片名称

#### *struTime*

图片的时间

#### *dwFileSize*

图片的大小

#### *sCardNum*

卡号

#### *byPlateColor*

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_GREEN_PLATE,
    VCA_BKAIR_PLATE,
    VCA_OTHER = 0xff
}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

*VCA\_GREEN\_PLATE*

绿色车牌

*VCA\_BKAIR\_PLATE*

民航黑色车牌

*VCA\_OTHER = 0xff*

其他

#### *byVehicleLogo*

汽车品牌，定义如下所示：

```
enum _VLR_VEHICLE_CLASS_{
    VLR_OTHER          = 0,    //其它
    VLR_VOLKSWAGEN    = 1,    //大众
    VLR_BUICK          = 2,    //别克
    VLR_BMW            = 3,    //宝马
    VLR_HONDA          = 4,    //本田
    VLR_PEUGEOT        = 5,    //标致
    VLR_TOYOTA         = 6,    //丰田
    VLR_FORD           = 7,    //福特
    VLR_NISSAN         = 8,    //日产
    VLR_AUDI           = 9,    //奥迪
    VLR_MAZDA          = 10,   //马自达
    VLR_CHEVROLET      = 11,   //雪佛兰
    VLR_CITROEN        = 12,   //雪铁龙
    VLR_HYUNDAI        = 13,   //现代
    VLR_CHERY          = 14,   //奇瑞
    VLR_KIA             = 15,   //起亚
    VLR_ROEWE          = 16,   //荣威
    VLR_MITSUBISHI     = 17,   //三菱
    VLR_SKODA          = 18,   //斯柯达
    VLR_GEELY          = 19,   //吉利
    VLR_ZHONGHUA       = 20,   //中华
    VLR_VOLVO          = 21,   //沃尔沃
    VLR_LEXUS          = 22,   //雷克萨斯
    VLR_FIAT           = 23,   //菲亚特
    VLR_EMGRAND        = 24,   //帝豪
    VLR_DONGFENG       = 25,   //东风
    VLR_BYD            = 26,   //比亚迪
    VLR_SUZUKI         = 27,   //铃木
    VLR_JINBEI         = 28,   //金杯
    VLR_HAIMA          = 29,   //海马
    VLR_SGMW           = 30,   //五菱
    VLR_JAC             = 31,   //江淮
    VLR_SUBARU         = 32,   //斯巴鲁
    VLR_ENGLON         = 33,   //英伦
    VLR_GREATWALL      = 34,   //长城
}
```

```

VLR_HAFEI      = 35, //哈飞
VLR_ISUZU     = 36, //五十铃
VLR_SOUEAST   = 37, //东南
VLR_CHANA     = 38, //长安
VLR_FOTON     = 39, //福田
VLR_XIALI     = 40, //夏利
VLR_BENZ      = 41, //奔驰
VLR_FAW       = 42, //一汽
VLR_NAVECO    = 43, //依维柯
VLR_LIFAN     = 44, //力帆
VLR_BESTURN   = 45, //一汽奔腾
VLR_CROWN     = 46, //皇冠
VLR_RENAULT   = 47, //雷诺
VLR_JMC       = 48, //JMC
VLR_MG        = 49, //MG 名爵
VLR_KAMA      = 50, //凯马
VLR_ZOTYE     = 51, //众泰
VLR_CHANGHE   = 52, //昌河
VLR_XMKINGLONG = 53, //厦门金龙
VLR_HUIZHONG  = 54, //上海汇众
VLR_SZKINGLONG = 55, //苏州金龙
VLR_HIGER     = 56, //海格
VLR_YUTONG    = 57, //宇通
VLR_CNHTC     = 58, //中国重汽
VLR_BEIBEN    = 59, //北奔重卡
VLR_XINGMA    = 60, //华菱星马
VLR_YUEJIN    = 61, //跃进
VLR_HUANGHAI = 62 //黄海
}VLR_VEHICLE_CLASS

```

byRes

保留

## 7.50 NET\_DVR\_FIND\_PICTURE\_PARAM: 图片查找条件

```

struct{
    DWORD      dwSize;
    LONG       lChannel;
    BYTE       byFileType;
    BYTE       byNeedCard;
    BYTE       byRes1[2];
    BYTE       sCardNum[CARDNUM_LEN_V30];
    NET_DVR_TIME struStartTime;
    NET_DVR_TIME struStopTime;
    DWORD      dwTrafficType;
}

```

```

DWORD      dwVehicleType;
DWORD      dwIllegalType;
BYTE       byLaneNo;
BYTE       byRes2[27];

```

```
}NET_DVR_FIND_PICTURE_PARAM,*LPNET_DVR_FIND_PICTURE_PARAM;
```

### Members

#### *dwSize*

结构体大小

#### *lChannel*

通道号

#### *byFileType*

查找的图片类型：0- 定时抓图，1- 移动侦测抓图，2- 报警抓图，3- 报警|移动侦测抓图，4- 报警&移动侦测抓图，6- 手动抓图，9-智能图片，10- PIR 报警，11- 无线报警，12- 呼救报警，0xa- 预览时截图，0xd- 人脸侦测，0xe- 越界侦测，0xf- 入侵区域侦测，0x10- 场景变更侦测，0x11- 设备本地回放时截图，0x12- 智能侦测，0xff- 全部类型

#### *byNeedCard*

是否需要卡号

#### *byRes1*

保留，置为 0

#### *sCardNum*

卡号

#### *struStartTime*

查找图片的开始时间

#### *struStopTime*

查找图片的结束时间

#### *dwTrafficType*

图片检索生效项，具体类型如下：

```

enum _VCA_OPERATE_TYPE_{
    VCA_LICENSE_TYPE      = 0x1,
    VCA_PLATECOLOR_TYPE   = 0x2,
    VCA_CARDNO_TYPE       = 0x4,
    VCA_PLATETYPE_TYPE    = 0x8,
    VCA_LISTTYPE_TYPE     = 0x10,
    VCA_INDEX_TYPE        = 0x20,
    VCA_OPERATE_INDEX_TYPE = 0x40
}VCA_OPERATE_TYPE

```

*VCA\_LICENSE\_TYPE*

车牌号码

*VCA\_PLATECOLOR\_TYPE*

车牌颜色

*VCA\_CARDNO\_TYPE*

卡号

*VCA\_PLATETYPE\_TYPE*

车牌类型



*VCA\_LISTTYPE\_TYPE*

车辆名单类型

*VCA\_INDEX\_TYPE*

数据流水号

*VCA\_OPERATE\_INDEX\_TYPE*

操作数

*dwVehicleType*

车辆类型，具体类型如下：

```
enum _VCA_VEHICLE_TYPE_{
    VCA_OTHER_TYPE           = 0x1,
    VCA_SMALLCAR_TYPE       = 0x2,
    VCA_BIGCAR_TYPE         = 0x4,
    VCA_BUS_TYPE            = 0x8,
    VCA_TRUCK_TYPE          = 0x10,
    VCA_CAR_TYPE            = 0x20,
    VCA_MINIBUS_TYPE        = 0x40,
    VCA_SMALL_TRUCK_TYPE    = 0x80
}VCA_VEHICLE_TYPE
```

*VCA\_OTHER\_TYPE*

其它车型

*VCA\_SMALLCAR\_TYPE*

小型车

*VCA\_BIGCAR\_TYPE*

大型车

*VCA\_BUS\_TYPE*

客车

*VCA\_TRUCK\_TYPE*

货车

*VCA\_CAR\_TYPE*

轿车

*VCA\_MINIBUS\_TYPE*

面包车

*VCA\_SMALL\_TRUCK\_TYPE*

小货车

*dwIllegalType*

违规检测类型，具体类型如下：

```
enum _VCA_ILLEGAL_TYPE_{
    VCA_POSTPIC_TYPE         = 0x1,
    VCA_LOWSPEED_TYPE       = 0x2,
    VCA_HIGHSPEED_TYPE      = 0x4,
    VCA_RETROGRADE_TYPE     = 0x8,
    VCA_RUSH_REDLIGHT_TYPE  = 0x10,
    VCA_PRESS_LANE_TYPE     = 0x20,
    VCA_VIOLATE_GUIDE_TYPE  = 0x40,
```

```

VCA_ROAD_STRAND_TYPE           = 0x80,
VCA_VEHICLE_ILLEGAL_LANE_TYPE = 0x100,
VCA_ILLEGAL_LANE_CHANGE_TYPE  = 0x200,
VCA_ILLEGAL_LANE_DRIVE_TYPE   = 0x400,
VCA_VIOLATE_BAN_TYPE          = 0x800,
VCA_CROSS_PARKING_TYPE        = 0x1000,
VCA_GREEN_PARKING_TYPE        = 0x2000

```

```
}VCA_ILLEGAL_TYPE
```

```
VCA_POSTPIC_TYPE
```

卡口图片

```
VCA_LOWSPEED_TYPE
```

低速

```
VCA_HIGHSPEED_TYPE
```

高速

```
VCA_RETROGRADE_TYPE
```

逆行

```
VCA_RUSH_REDLIGHT_TYPE
```

闯红灯

```
VCA_PRESS_LANE_TYPE
```

压车道线

```
VCA_VIOLATE_GUIDE_TYPE
```

不按导向

```
VCA_ROAD_STRAND_TYPE
```

路口滞留(设备不支持)

```
VCA_VEHICLE_ILLEGAL_LANE_TYPE
```

机占非

```
VCA_ILLEGAL_LANE_CHANGE_TYPE
```

违法变道

```
VCA_ILLEGAL_LANE_DRIVE_TYPE
```

不按车道

```
VCA_VIOLATE_BAN_TYPE
```

违法禁令

```
VCA_CROSS_PARKING_TYPE
```

路口停车

```
VCA_GREEN_PARKING_TYPE
```

绿灯停车

*byLaneNo*

车道号, 取值范围: 1~99

*byRes2*

保留, 置为 0

## 7.51 NET\_DVR\_FIRE\_ALARM:消防报警信息

```
struct{
```

```

    DWORD           dwSize;
    NET\_DVR\_TIME\_V30 struAlarmTime;
    BYTE            byRes[128];
}NET_DVR_FIRE_ALARM,*LPNET_DVR_FIRE_ALARM;

```

### Members

*dwSize*

结构体大小

*struAlarmTime*

报警时间

*byRes*

保留，置为 0

## 7.52 NET\_DVR\_FLASH\_OUTCFG:IO 闪光灯输出参数

```

struct{
    DWORD   dwSize;
    BYTE    byMode;
    BYTE    byRelatedIoIn;
    BYTE    byRecognizedLane;
    BYTE    byDetectBrightness;
    BYTE    byBrightnessThreld;
    BYTE    byStartHour;
    BYTE    byStartMinute;
    BYTE    byEndHour;
    BYTE    byEndMinute;
    BYTE    byFlashLightEnable;
    BYTE    byRes[2];
}NET_DVR_FLASH_OUTCFG, *LPNET_DVR_FLASH_OUTCFG;

```

### Members

*dwSize*

结构体大小

*byMode*

闪光灯闪烁模式，0-不闪，1-闪，2-关联闪，3-轮闪

*byRelatedIoIn*

闪光灯关联的输入 IO 号（关联闪时此参数有效，暂时保留）

*byRecognizedLane*

关联的 IO 号，按位表示，bit0 表示 IO1 是否关联，0-不关联，1-关联

*byDetectBrightness*

自动检测亮度使能闪光灯 0-不检测；1-检测

*byBrightnessThreld*

使能闪光灯亮度阈值，范围[0,100],高于阈值闪

*byStartHour*

开始时间，单位：小时，取值范围[0,23]

*byStartMinute*

开始时间，单位：分，取值范围[0,59]

*byEndHour*

结束时间，单位：小时，取值范围[0,23]

*byEndMinute*

结束时间，单位：分，取值范围[0,59]

*byFlashLightEnable*

闪光灯时间是否开启：0-关，1-开

*byRes*

保留，置为 0

## 7.53 NET\_DVR\_FTPCFG:FTP 上传参数

```
struct{
    DWORD      dwSize;
    DWORD      dwEnableFTP;
    char       sFTPIP[16];
    DWORD      dwFTPPort;
    BYTE       sUserName[NAME_LEN];
    BYTE       sPassword[PASSWD_LEN];
    DWORD      dwDirLevel;
    WORD       wTopDirMode;
    WORD       wSubDirMode;
    BYTE       byEnableAnony;
    BYTE       byRes[23];
}NET_DVR_FTPCFG, *LPNET_DVR_FTPCFG;
```

### Members

*dwSize*

结构体大小

*dwEnableFTP*

是否启用 FTP，0-不启用，1-启用

*sFTPIP*

FTP 服务器 IP 地址

*dwFTPPort*

FTP 端口号

*sUserName*

用户名

*sPassword*

密码

*dwDirLevel*

目录结构，0-不使用目录结构，直接保存在根目录，1-使用 1 级目录，2-使用 2 级目录

*wTopDirMode*

一级目录，0x1-使用设备名，0x2-使用设备号，0x3-使用设备 ip 地址，0x4-使用监测点，0x5-使用时间(年月)，0x6-自定义，0x7-违规类型，0x8-方向，0x9-地点

*wSubDirMode*

二级目录, 0x1-使用通道名, 0x2-使用通道号, 0x3-使用时间(年月日), 0x4-使用车道号, 0x5-自定义, 0x6-违规类型, 0x7-方向, 0x8-地点

*byEnableAnony*

启用匿名: 0- 否, 1- 是, 智能交通摄像机不支持该字段

*byRes*

保留

## 7.54 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

## 7.55 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

## 7.56 NET\_DVR\_GATELAMP\_INFO:常亮灯控制参数

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    BYTE     byLaneNo;
    BYTE     byBrightlampCtrl;
    BYTE     byRes[14];
}NET_DVR_GATELAMP_INFO,*LPNET_DVR_GATELAMP_INFO;
```

### Members

#### *dwSize*

结构体大小

#### *dwChannel*

通道号

#### *byLaneNo*

道闸号: 0- 表示无效值(设备需要做有效值判断), 1- 道闸 1

#### *byBrightlampCtrl*

控制参数: 0- 关闭常亮灯, 1- 开启常亮灯

#### *byRes*

保留, 置为 0

## 7.57 NET\_DVR\_GEOGLOCATION:地址位置信息

```
struct{
    int      iRes[2];
    DWORD    dwCity;
}NET_DVR_GEOGLOCATION,*LPNET_DVR_GEOGLOCATION;
```

### Members

#### *iRes*

保留, 置为 0

#### *dwCity*

城市

```
enum _PROVINCE_CITY_IDX_{
    ANHUI_PROVINCE           = 0,
    AOMEN_PROVINCE           = 1,
    BEIJING_PROVINCE         = 2,
    CHONGQING_PROVINCE       = 3,
    FUJIAN_PROVINCE          = 4,
    GANSU_PROVINCE           = 5,
    GUANGDONG_PROVINCE       = 6,
    GUANGXI_PROVINCE         = 7,
    GUIZHOU_PROVINCE         = 8,
    HAINAN_PROVINCE          = 9,
```

```
HEBEI_PROVINCE           = 10,  
HENAN_PROVINCE           = 11,  
HEILONGJIANG_PROVINCE   = 12,  
HUBEI_PROVINCE           = 13,  
HUNAN_PROVINCE           = 14,  
JILIN_PROVINCE           = 15,  
JIANGSU_PROVINCE        = 16,  
JIANGXI_PROVINCE        = 17,  
LIAONING_PROVINCE       = 18,  
NEIMENGGU_PROVINCE      = 19,  
NINGXIA_PROVINCE        = 20,  
QINGHAI_PROVINCE        = 21,  
SHANDONG_PROVINCE       = 22,  
SHANXI_JIN_PROVINCE     = 23,  
SHANXI_SHAN_PROVINCE    = 24,  
SHANGHAI_PROVINCE       = 25,  
SICHUAN_PROVINCE        = 26,  
TAIWAN_PROVINCE         = 27,  
TIANJIN_PROVINCE        = 28,  
XIZANG_PROVINCE         = 29,  
XIANGGANG_PROVINCE      = 30,  
XINJIANG_PROVINCE       = 31,  
YUNNAN_PROVINCE         = 32,  
ZHEJIANG_PROVINCE       = 33  
}PROVINCE_CITY_IDX
```

*ANHUI\_PROVINCE*

安徽

*AOMEN\_PROVINCE*

澳门

*BEIJING\_PROVINCE*

北京

*CHONGQING\_PROVINCE*

重庆

*FUJIAN\_PROVINCE*

福建

*GANSU\_PROVINCE*

甘肃

*GUANGDONG\_PROVINCE*

广东

*GUANGXI\_PROVINCE*

广西

*GUIZHOU\_PROVINCE*

贵州

*HAINAN\_PROVINCE*

海南

*HEBEI\_PROVINCE*

河北

*HENAN\_PROVINCE*

河南

*HEILONGJIANG\_PROVINCE*

黑龙江

*HUBEI\_PROVINCE*

湖北

*HUNAN\_PROVINCE*

湖南

*JILIN\_PROVINCE*

吉林

*JIANGSU\_PROVINCE*

江苏

*JIANGXI\_PROVINCE*

江西

*LIAONING\_PROVINCE*

辽宁

*NEIMENGGU\_PROVINCE*

内蒙古

*NINGXIA\_PROVINCE*

宁夏

*QINGHAI\_PROVINCE*

青海

*SHANDONG\_PROVINCE*

山东

*SHANXI\_JIN\_PROVINCE*

山西

*SHANXI\_SHAN\_PROVINCE*

陕西

*SHANGHAI\_PROVINCE*

上海

*SICHUAN\_PROVINCE*

四川

*TAIWAN\_PROVINCE*

台湾

*TIANJIN\_PROVINCE*

天津

*XIZANG\_PROVINCE*

西藏

*XIANGGANG\_PROVINCE*

香港

*XINJIANG\_PROVINCE*



新疆  
YUNNAN\_PROVINCE  
云南  
ZHEJIANG\_PROVINCE  
浙江

## 7.58 NET\_DVR\_GPS\_DATACFG:GPS 参数

```
struct{
    DWORD                dwSize;
    BYTE                 byGpsDataMode;
    BYTE                 byLongitudeType;
    BYTE                 byLatitudeType;
    BYTE                 byRes;
    NET_DVR_LATITUDE_PARAM struLatitude;
    NET_DVR_LONGITUDE_PARAM struLongitude;
    BYTE                 byRes1[128];
}NET_DVR_GPS_DATACFG, *LPNET_DVR_GPS_DATACFG;
```

### Members

#### *dwSize*

结构体大小

#### *byGpsDataMode*

GPS 数据获取方式: 0- 自动, 1- 手动

#### *byLongitudeType*

经度: 0- 东经, 1- 西经

#### *byLatitudeType*

纬度: 0- 南纬, 1- 北纬

#### *byRes*

保留, 置为 0

#### *struLatitude*

纬度

#### *struLongitude*

经度

#### *byRes1*

保留, 置为 0

## 7.59 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

**7.60 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。

**7.61 NET\_DVR\_IMAGEOVERLAYCFG:图像叠加参数**

```
struct{
    DWORD  dwSize;
    BYTE   byOverlayInfo;
    BYTE   byOverlayMonitorInfo;
    BYTE   byOverlayTime;
    BYTE   byOverlaySpeed;
    BYTE   byOverlaySpeeding;
    BYTE   byOverlayLimitFlag;
    BYTE   byOverlayPlate;
```

```

BYTE    byOverlayColor;
BYTE    byOverlayLength;
BYTE    byOverlayType;
BYTE    byOverlayColorDepth;
BYTE    byOverlayDriveChan;
BYTE    byOverlayMilliSec;
BYTE    byOverlayIllegalInfo;
BYTE    byOverlayRedOnTime;
BYTE    byRes1[5];
BYTE    byMonitorInfo1[32];
BYTE    byMonitorInfo2[44];
BYTE    byRes2[52];
}NET_DVR_IMAGEOVERLAYCFG, *LPNET_DVR_IMAGEOVERLAYCFG;

```

### Members

#### *dwSize*

大小

#### *byOverlayInfo*

叠加使能开关，0-不叠加，1-叠加

#### *byOverlayMonitorInfo*

是否叠加监测点信息，0-不叠加，1-叠加

#### *byOverlayTime*

是否叠加时间，0-不叠加，1-叠加

#### *byOverlaySpeed*

是否叠加速度，0-不叠加，1-叠加

#### *byOverlaySpeeding*

是否叠加超速比例，0-不叠加，1-叠加

#### *byOverlayLimitFlag*

是否叠加限速标志，0-不叠加，1-叠加

#### *byOverlayPlate*

是否叠加车牌，0-不叠加，1-叠加

#### *byOverlayColor*

是否叠加车身颜色，0-不叠加，1-叠加

#### *byOverlayLength*

是否叠加车长，0-不叠加，1-叠加

#### *byOverlayType*

是否叠加车型，0-不叠加，1-叠加

#### *byOverlayColorDepth*

是否叠加车身颜色深浅，保留

#### *byOverlayDriveChan*

是否叠加车道，0-不叠加，1-叠加

#### *byOverlayMilliSec*

是否叠加毫秒信息，0-不叠加，1-叠加

#### *byOverlayIllegalInfo*

是否叠加违章信息，0-不叠加，1-叠加

*byOverlayRedOnTime*

叠加红灯已亮时间 0-不叠加, 1-叠加

*byres1*

保留, 置为 0

*byMonitorInfo1*

监测点信息 1

*byMonitorInfo2*

监测点信息 2

*byres2*

保留, 置为 0

## 7.62 NET\_DVR\_IO\_INCFG:IO 输入配置参数

```
struct{
    DWORD    dwSize;
    BYTE     byIoInStatus;
    BYTE     byRes[3];
}NET_DVR_IO_INCFG, *LPNET_DVR_IO_INCFG;
```

### Members

*dwSize*

大小

*byIoInStatus*

输入的 IO 口状态, 0-下降沿, 1-上升沿, 2-上升沿和下降沿, 3-高电平, 4-低电平

*byRes*

保留, 置为 0

## 7.63 NET\_DVR\_IO\_OUTCFG:IO 输出配置参数

```
struct{
    DWORD    dwSize;
    BYTE     byDefaultStatus;
    BYTE     byIoOutStatus;
    WORD     wAheadTime;
    DWORD    dwTimePluse;
    DWORD    dwTimeDelay;
    BYTE     byFreqMulti;
    BYTE     byDutyRate;
    BYTE     byRes[2];
}NET_DVR_IO_OUTCFG, *LPNET_DVR_IO_OUTCFG;
```

### Members

*dwSize*

大小

*byDefaultStatus*



```

    DWORD dwAlarmInPicChanTriggered[MAX_ALARMIN_V30];
}NET_DVR_JPEGCFG_V30, *LPNET_DVR_JPEGCFG_V30;

```

### Members

#### *dwSize*

大小

#### *struJpegPara*

每个通道的图像参数

#### *wBurstMode*

抓图方式, 按位设置, 0x1=报警输入触发, 0x2=移动侦测触发, 0x4=232 触发, 0x8=485 触发, 0x10=网络触发

#### *wUploadInterval*

图片上传间隔(秒), 取值范围[0, 65535]

#### *struPicNameRule*

图片命名规则

#### *bySaveToHD*

是否保存到硬盘

#### *byRes1*

保留

#### *wCatchInterval*

抓图间隔 (秒), 取值范围[0, 65535]

#### *byRes2*

保留

#### *struRs232Cfg*

232 抓图参数

#### *struRs485Cfg*

485 串口抓图参数

#### *dwTriggerPicTimes*

每个通道一次触发拍照次数

#### *dwAlarmInPicChanTriggered*

报警触发抓拍通道, 按位设置, 从第 1 位开始

## 7.66 NET\_DVR\_JPEGPARA:JPEG 图像信息

```

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

```

### Members

#### *wPicSize*

图片尺寸: 0-CIF, 1-QCIF, 2-D1, 3-UXGA(1600x1200), 4-SVGA(800x600), 5-HD720p(1280x720), 6-VGA, 7-XVGA, 8-HD900p, 9-HD1080, 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

#### *wPicQuality*

图片质量系数：0-最好，1-较好，2-一般

## 7.67 NET\_DVR\_LATITUDE\_PARAM:纬度参数

```
struct{
    BYTE    byDegree;
    BYTE    byMinute;
    BYTE    bySec;
    BYTE    byRes;
}NET_DVR_LATITUDE_PARAM, *LPNET_DVR_LATITUDE_PARAM;
```

### Members

#### *byDegree*

度，取值范围：[0,179]

#### *byMinute*

分，取值范围：[0,59]

#### *byMinute*

秒，取值范围：[0,59]

#### *byRes*

保留，置为 0

## 7.68 NET\_DVR\_LASER\_PARAM\_CFG:激光参数配置

```
struct{
    BYTE    byControlMode;
    BYTE    bySensitivity;
    BYTE    byTriggerMode;
    BYTE    byBrightness;
    BYTE    byAngle;
    BYTE    byLimitBrightness;
    BYTE    byRes[10];
}NET_DVR_LASER_PARAM_CFG, *LPNET_DVR_LASER_PARAM_CFG;
```

### Members

#### *byControlMode*

控制模式：0-无效，1-自动，2-手动，默认：自动

#### *bySensitivity*

激光灯灵敏度，取值范围：0~100，默认：50

#### *byTriggerMode*

激光灯触发模式：0-无效，1-机芯触发，2-光敏触发，默认：机芯触发

#### *byBrightness*

激光灯亮度，控制模式为手动模式下有效，取值范围：0~255，默认：100

#### *byAngle*

激光灯角度，0 表示无效，取值范围：1~36，默认：12。激光灯照射范围为一个圆圈，调节激光角度是调节这个圆的半径的大小

*byLimitBrightness*

激光灯亮度限制，控制模式为自动模式下有效，取值范围：0~100

*byRes*

保留

## 7.69 NET\_DVR\_LIGHTSNAPCFG:红绿灯功能参数

```
struct{
    DWORD    dwSize;
    BYTE     byLightIoIn;
    BYTE     byTrigIoIn;
    BYTE     byRelatedDriveWay;
    BYTE     byTrafficLight;
    BYTE     bySnapTimes1;
    BYTE     bySnapTimes2;
    BYTE     byRes1[2];
    WORD     wIntervalTime1[MAX_INTERVAL_NUM];
    WORD     wIntervalTime2[MAX_INTERVAL_NUM];
    BYTE     byRecord;
    BYTE     bySessionTimeout;
    BYTE     byPreRecordTime;
    BYTE     byVideoDelay;
    BYTE     byRes2[32];
}NET_DVR_LIGHTSNAPCFG, *LPNET_DVR_LIGHTSNAPCFG;
```

### Members

*dwSize*

大小

*byLightIoIn*

红绿灯的 IO 号，从 1 开始

*byTrigIoIn*

触发的 IO 号，从 1 开始

*byRelatedDriveWay*

触发 IO 关联的车道号，取值范围[0,9]

*byTrafficLight*

0-高电平红灯，低电平绿灯；1-高电平绿灯，低电平红灯

*bySnapTimes1*

红灯抓拍次数，0-不抓拍，非 0-连拍次数，最大 5 次

*bySnapTimes2*

绿灯抓拍次数，0-不抓拍，非 0-连拍次数，最大 5 次

*byRes1*

保留，置为 0

*wIntervalTime1*

红灯连拍间隔时间，单位 ms，取值范围[67,60000]

*wIntervalTime2*



绿灯连拍间隔时间，单位 ms，取值范围[67,60000]

*byRecord*

闯红灯周期录像标志，0-不录像，1-录像

*bySessionTimeout*

闯红灯周期录像超时时间（单位秒），取值范围[0,100]

*byPreRecordTime*

闯红灯录像片段预录时间(单位秒)，取值范围[0,100]

*byVideoDelay*

闯红灯录像片段延时时间（单位秒），取值范围[0,100]

*byRes2*

保留，置为 0

## 7.70 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）。

## 7.71 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。

## 7.72 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。

## 7.73 NET\_DVR\_LOCAL\_LOG\_CFG:SDK 日志参数配置

```
struct{
    WORD    wSDKLogNum;
    BYTE    byRes[254];
}NET_DVR_LOCAL_LOG_CFG,*LPNET_DVR_LOCAL_LOG_CFG;
```

#### Members

##### wSDKLogNum

SDK 日志文件个数，SDK 写日志模式为覆盖模式([NET\\_DVR\\_SetLogToFile](#)接口中参数 bAutoDel 为 TRUE) 时日志生成的个数，0 表示默认值（即 10 个）

##### byRes

保留，置为 0

## 7.74 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

## 7.75 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

## 7.76 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

## 7.77 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。

## 7.78 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。

## 7.79 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。

## 7.80 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_FRAMEYPE	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

## 7.81 NET\_DVR\_LONGITUDE\_PARAM:经度参数

```
struct{
    BYTE    byDegree;
    BYTE    byMinute;
    BYTE    bySec;
    BYTE    byRes;
}NET_DVR_LONGITUDE_PARAM, *LPNET_DVR_LONGITUDE_PARAM;
```

### Members

*byDegree*

度，取值范围：[0,179]

*byMinute*

分，取值范围：[0,59]

*byMinute*

秒，取值范围：[0,59]

*byRes*

保留，置为 0

## 7.82 NET\_DVR\_MANUALSNAP:手动抓拍请求参数

```
struct{
```

```

BYTE    byOSDEnable;
BYTE    byLaneNo;
BYTE    byRes[22];

```

```
}NET_DVR_MANUALSNAP, *LPNET_DVR_MANUALSNAP;
```

### Members

#### *byOSDEnable*

抓拍图片上是否关闭 OSD 信息叠加: 0- 不关闭(默认), 1- 关闭

#### *byLaneNo*

车道号, 取值范围: 1~6, 默认为 1

#### *byRes*

保留, 置为 0

## 7.83 NET\_DVR\_MEASURESPEEDCFG:测速功能参数

```

struct{
    DWORD    dwSize;
    BYTE    byTriglo1;
    BYTE    byTriglo2;
    BYTE    byRelatedDriveWay;
    BYTE    byTestSpeedTimeOut;
    DWORD    dwDistance;
    BYTE    byCapSpeed;
    BYTE    bySpeedLimit;
    BYTE    bySnapTimes1;
    BYTE    bySnapTimes2;
    WORD    wIntervalTime1[MAX_INTERVAL_NUM];
    WORD    wIntervalTime2[MAX_INTERVAL_NUM];
    BYTE    byRes[32];
}NET_DVR_MEASURESPEEDCFG, *LPNET_DVR_MEASURESPEEDCFG;

```

### Members

#### *dwSize*

结构体大小

#### *byTriglo1*

测速第 1 线圈, 从 1 开始

#### *byTriglo2*

测速第 2 线圈, 从 1 开始

#### *byRelatedDriveWay*

触发 IO 关联的车道号, 取值范围[0,9]

#### *byTestSpeedTimeOut*

测速模式超时时间, 单位 s, 取值范围[0,255]

#### *dwDistance*

线圈距离, 单位 cm, 取值范围[0,20000]

#### *byCapSpeed*

测速模式起拍速度, 单位 km/h, 取值范围[0,255]



**bySpeedLimit**

限速值，单位 km/h，取值范围[0,255]

**bySnapTimes1**

线圈 1 抓拍次数，0-不抓拍，非 0-连拍次数，最大 5 次

**bySnapTimes2**

线圈 2 抓拍次数，0-不抓拍，非 0-连拍次数，最大 5 次

**wIntervalTime1**

线圈 1 连拍间隔时间，单位 ms，取值范围[67,20000]

**wIntervalTime2**

线圈 2 连拍间隔时间，单位 ms，取值范围[67,20000]

**byRes**

保留，置为 0

## 7.84 NET\_DVR\_MUL\_STREAM\_CHANNEL\_GROUP:单组码流配置参数

```
struct{
    DWORD dwSize;
    DWORD dwGroup;
    NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND struStreamPara;
    BYTE byRes[64];
}NET_DVR_MUL_STREAM_CHANNEL_GROUP,*LPNET_DVR_MUL_STREAM_CHANNEL_GROUP;
```

### Members

**dwSize**

结构体大小

**dwGroup**

组号

**struStreamPara**

多码流配置参数

**byRes**

保留

## 7.85 NET\_DVR\_MULTI\_STREAM\_COMPRESSIONCFG:多码流压缩参数配

### 置

```
struct{
    DWORD dwSize;
    DWORD dwStreamType;
    NET_DVR_COMPRESSION_INFO_V30 struStreamPara;
    BYTE byRes[80];
}NET_DVR_MULTI_STREAM_COMPRESSIONCFG,*LPNET_DVR_MULTI_STREAM_COMPRESSIONCFG;
```

### Members

**dwSize**

结构体大小

*dwStreamType*

码流类型：0-主码流，1-子码流，2-事件类型，3-码流 3，4-虚拟码流，.....

*struStreamPara*

码流压缩参数

*byRes*

保留

## 7.86 NET\_DVR\_MULTI\_STREAM\_COMPRESSIONCFG\_COND:多码流压缩参数配置条件

```
struct{
    DWORD                dwSize;
    NET\_DVR\_STREAM\_INFO  struStreamInfo;
    DWORD                dwStreamType;
    BYTE                 byRes[32];
}NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND,*LPNET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND;
```

### Members

*dwSize*

结构体大小

*struStreamInfo*

流信息，流 ID 或者通道号

*dwStreamType*

码流类型：0-主码流，1-子码流，2-事件类型，3-码流 3，4-虚拟码流，.....

*byRes*

保留

## 7.87 NET\_DVR\_NFSCFG:网络文件系统参数

```
struct{
    DWORD                dwSize;
    NET\_DVR\_SINGLE\_NFS  struNfsDiskParam[MAX_NFS_DISK];
}NET_DVR_NFSCFG,*LPNET_DVR_NFSCFG;
```

### Members

*dwSize*

结构体大小

*struNfsDiskParam*

NFS 配置子结构

### Remarks

修改 NFS 各参数时需要重启设备才能生效。

## 7.88 NET\_DVR\_NETCFG\_MULTI:多网卡网络配置

```

struct{
    DWORD                dwSize;
    BYTE                 byDefaultRoute;
    BYTE                 byNetworkCardNum;
    BYTE                 byRes[2];
    NET_DVR_ETHERNET_MULTI struEtherNet[MAX_NETWORK_CARD];
    NET_DVR_IPADDR       struManageHost1IpAddr;
    NET_DVR_IPADDR       struManageHost2IpAddr;
    NET_DVR_IPADDR       struAlarmHostIpAddr;
    WORD                 wManageHost1Port;
    WORD                 wManageHost2Port;
    WORD                 wAlarmHostIpPort;
    BYTE                 byIpResolver[MAX_DOMAIN_NAME];
    WORD                 wIpResolverPort;
    WORD                 wDvrPort;
    WORD                 wHttpPortNo;
    BYTE                 byRes2[6];
    NET_DVR_IPADDR       struMulticastIpAddr;
    NET_DVR_PPPOECFG     struPPPoE;
    BYTE                 byRes3[24];
}NET_DVR_NETCFG_MULTI,*LPNET_DVR_NETCFG_MULTI;

```

### Members

#### *dwSize*

结构体大小

#### *byDefaultRoute*

默认路由，0 表示 struEtherNet[0]，1 表示 struEtherNet[1]

#### *byNetworkCardNum*

设备实际可配置的网卡数目

#### *byRes*

保留，置为 0

#### *struEtherNet*

以太网口

#### *struManageHost1IpAddr*

主管理主机 IP 地址

#### *struManageHost2IpAddr*

辅管理主机 IP 地址

#### *struAlarmHostIpAddr*

报警主机 IP 地址

#### *wManageHost1Port*

主管理主机端口号

#### *wManageHost2Port*

辅管理主机端口号

*wAlarmHostIpPort*

报警主机端口号

*byIpResolver*

IP 解析服务器域名或 IP 地址

*wIpResolverPort*

IP 解析服务器端口号

*wDvrPort*

通讯端口，默认 8000

*wHttpPortNo*

HTTP 端口号

*byRes2*

保留，置为 0

*struMulticastIpAddr*

多播组地址

*struPPPoE*

PPPoE 参数

*byRes3*

保留，置为 0

## 7.89 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*

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

### Remarks

3.0 协议以下的设备，参数 *byUseDhcp* 为 0xff-无效，将其 IP 地址填成空，设备会自动去获取 DHCP。

## 7.90 NET\_DVR\_NETWORK\_BONDING:BONDING 网卡配置

```
struct{
    DWORD                dwSize;
    BYTE                 byEnable;
    BYTE                 byNum;
    BYTE                 byRes1[2];
    NET\_DVR\_ONE\_BONDING struOneBond[MAX_BOND_NUM];
    BYTE                 byRes2[40];
}
```

```
}NET_DVR_NETWORK_BONDING,*LPNET_DVR_NETWORK_BONDING;
```

### Members

*dwSize*

结构体大小

*byEnable*

是否启用 bonding 功能

*byNum*

Bonding 网卡的个数

*byRes1*

保留

*struOneBond*

单 BONDING 网卡配置参数

*byRes2*

保留

## 7.91 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

## 7.92 NET\_DVR\_NTPPARA: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，以分钟或小时为单位（通过能力集 [NET\\_DVR\\_GetDeviceAbility](#) 获取，对应网络应用参数能力集：DEVICE\_NETAPP\_ABILITY）

#### *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

## 7.93 NET\_DVR\_ONE\_BONDING:单 BONDING 网卡配置

```

struct{
    BYTE          byMode;
    BYTE          byUseDhcp;
    BYTE          byMasterCard;
    BYTE          byStatus;
    BYTE          byBond[MAX_NETWORK_CARD];
    NET\_DVR\_ETHERNET\_V30 struEtherNet;
    NET\_DVR\_IPADDR    struGatewayIpAddr;
    BYTE          byRes[20];
}NET_DVR_ONE_BONDING,*LPNET_DVR_ONE_BONDING;

```

### Members

#### *byMode*

工作模式：0- 网络容错，1- 负载均衡

#### *byUseDhcp*

是否使能 dhcp

#### *byMasterCard*

指定哪张网卡为主网卡

#### *byStatus*

BONDING 的状态: 0 - 异常, 1-正常, 只能获取不能设置

*byBond*

*byBond*[0]== 1 表示使用 eh0, 0 表示不使用 eh0

*struEtherNet*

网卡参数

*struGatewayIpAddr*

网关地址

*byRes*

保留

## 7.94 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                 byOSDColorType;
    BYTE                 byAlignment;
    BYTE                 byOSDMilliSecondEnable;
    NET_DVR_RGB_COLOR   struOsdColor;
    BYTE                 byRes[56];
}NET_DVR_PICCFG_V30,*LPNET_DVR_PICCFG_V30;
```

### Members

*dwSize*

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，智能交通摄像机不支持，保留

*struRes*

保留，置为 0

*struMotion*

移动侦测报警参数，智能交通摄像机不支持，保留

*struHideAlarm*

遮挡报警参数，智能交通摄像机不支持，保留

*dwEnableHide*

是否启动遮挡：0-否，1-是，智能交通摄像机不支持，保留

*struShelter*

遮挡区域参数，智能交通摄像机不支持，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

*byOSDAttrib*

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，智能交通摄像机不支持，保留

*byOSDColorType*

OSD 颜色模式：0- 默认（黑白），1-自定义(颜色见 *struOsdColor*)

*byAlignment*

对齐方式：0- 自适应，1- 右对齐

*byOSDMilliSecondEnable*

视频叠加时间支持毫秒（智能交通摄像机 V4.1 及以后版本支持）：0- 不叠加，1- 叠加

*struOsdColor*

OSD 自定义颜色

*byRes*

保留，置为 0

### Remarks

设备通道名称、OSD、移动侦测、视频遮挡等图像参数配置能力，对应设备图像参数能力集（*VideoPicAbility*），相关接口：[NET\\_DVR\\_GetDeviceAbility](#)，能力集类型：DEVICE\_VIDEOPIC\_ABILITY。

智能交通摄像机，视频叠加是否支持时间毫秒（*byOSDMilliSecondEnable*）也可以通过能力集 [NET\\_DVR\\_SNAP\\_ABILITY](#) 中的 *bySupport2&0x02* 进行判断。

## 7.95 NET\_DVR\_PICTURE\_NAME:图片命名规则

```
struct{
    BYTE        byItemOrder[PICNAME_MAXITEM];
    BYTE        byDelimiter;
}NET_DVR_PICTURE_NAME, *LPNET_DVR_PICTURE_NAME;
```

### Members

*byItemOrder*

按数组定义文件命名的规则，每个元素取值见上传的相关元素的定义：

宏定义	宏定义值	含义
PICNAME_ITEM_DEV_NAME	1	设备名
PICNAME_ITEM_DEV_NO	2	设备号
PICNAME_ITEM_DEV_IP	3	设备 IP
PICNAME_ITEM_CHAN_NAME	4	通道名
PICNAME_ITEM_CHAN_NO	5	通道号
PICNAME_ITEM_TIME	6	时间
PICNAME_ITEM_CARDNO	7	卡号
PICNAME_ITEM_PLATE_NO	8	车牌号码
PICNAME_ITEM_PLATE_COLOR	9	车牌颜色
PICNAME_ITEM_CAR_CHAN	10	车道号

PICNAME_ITEM_CAR_SPEED	11	车辆速度
PICNAME_ITEM_CARCHAN	12	监测点
PICNAME_ITEM_SPEED_LIMIT_VALUES	15	限速值
PICNAME_ITEM_ILLEGAL_CODE	16	国标违法代码
PICNAME_ITEM_CROSS_NUMBER	17	路口编号
PICNAME_ITEM_DIRECTION_NUMBER	18	方向编号
PICNAME_ITEM_CAR_COLOR	19	车身颜色
PICNAME_ITEM_PLATE_COORDINATE	20	车牌坐标
PICNAME_ITEM_CAR_TYPE	21	车辆类型
PICNAME_ITEM_VIOLATION_TYPE	22	违规类型
PICNAME_ITEM_CUSTOM	255	自定义

*byDelimiter*

分隔符, “\_”“-”“+”“=”“.”, 一般为'\_'

## 7.96 NET\_DVR\_PIRIS\_PARAM:P-Iris 红外光圈大小等级配置

```
struct{
    BYTE    byMode;
    BYTE    byPirisAperture;
    BYTE    byRes[6];
}NET_DVR_PIRIS_PARAM, *LPNET_DVR_PIRIS_PARAM;
```

### Members

*byMode*

P-Iris 模式: 0- 自动, 1- 手动

*byPirisAperture*

红外光圈大小等级(等级越高, 光圈越大): 1~100, 默认:50, 手动模式下可修改

*byRes*

保留, 置为 0

## 7.97 NET\_DVR\_PLATE\_INFO:车牌信息

```
struct{
    BYTE    byPlateType;
    BYTE    byColor;
    BYTE    byBright;
    BYTE    byLicenseLen;
    BYTE    byEntireBelieve;
    BYTE    byRegion;
    BYTE    byCountry;
    BYTE    byRes[33];
}
```

```

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_GREEN_PLATE,
    VCA_BKAIR_PLATE,
    VCA_OTHER = 0xff
}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

*VCA\_GREEN\_PLATE*

绿色车牌

*VCA\_BKAIR\_PLATE*

民航黑色车牌

*VCA\_OTHER*

其他

*byBright*

车牌亮度

*byLicenseLen*

车牌字符个数

*byEntireBelieve*

整个车牌的置信度，取值范围：0~100

*byRegion*

区域索引值：0- 保留，1- 欧洲(Europe Region)，2- 俄罗斯(Russian Region)，3- 欧洲&amp;俄罗斯(EU&amp;CIS) ，0xff- 所有

*byCountry*国家索引值，取值定义详见 [COUNTRY\\_INDEX](#)，不支持 0xff(全部)*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%

## 7.98 NET\_DVR\_PLATE\_RESULT:车牌识别结果

```

struct{
    DWORD                dwSize;
    BYTE                 byResultType;
    BYTE                 byChanIndex;
    WORD                 wAlarmRecordID;
    DWORD                dwRelativeTime;
    BYTE                 byAbsTime[32];

```

```

DWORD          dwPicLen;
DWORD          dwPicPlateLen;
DWORD          dwVideoLen;
BYTE          byTrafficLight;
BYTE          byPicNum;
BYTE          byDriveChan;
BYTE          byVehicleType;
DWORD          dwBinPicLen;
DWORD          dwCarPicLen;
DWORD          dwFarCarPicLen;
BYTE          *pBuffer3;
BYTE          *pBuffer4;
BYTE          *pBuffer5;
BYTE          byRelaLaneDirectionType;
BYTE          byRes3[7];
NET\_DVR\_PLATE\_INFO struPlateInfo;
NET\_DVR\_VEHICLE\_INFO struVehicleInfo;
BYTE          *pBuffer1;
BYTE          *pBuffer2;
}NET_DVR_PLATE_RESULT, *LPNET_DVR_PLATE_RESULT;

```

### Members

#### *dwSize*

结构体大小

#### *byResultType*

识别结果类型，0-通过视频识别，1-通过图像识别，2-长录像(支持查询)

#### *byChanIndex*

车道号

#### *wAlarmRecordID*

报警录像 ID(用于查询录像，仅当 *byResultType* 为 2 时有效)

#### *dwRelativeTime*

相对时间（保留）

#### *byAbsTime*

绝对时间，精确到毫秒，yyyymmddhhmmssxxx，例如 20090810235959999

#### *dwPicLen*

图片长度（近景图）

#### *dwPicPlateLen*

车牌小图片长度（车牌彩图）

#### *dwVideoLen*

录像内容长度

#### *byTrafficLight*

0-非红绿灯抓拍，1-绿灯时抓拍；2-红灯时抓拍

#### *byPicNum*

连拍的图片序号

#### *byDriveChan*

触发的车道号

*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

*dwBinPicLen*

二值图长度（仅 iDS-65xx 支持）

*dwCarPicLen*

车辆原图长度（仅 iDS-65xx 支持）

*dwFarCarPicLen*

远景图长度（仅 iDS-65xx 支持）

*pBuffer3*

车牌二值图（仅 iDS-65xx 支持）

*pBuffer4*

车辆原图（仅 iDS-65xx 支持）

*pBuffer5*

远景图（仅 iDS-65xx 支持）

*byRelaLaneDirectionType*

关联车道方向类型，定义详见 [ITC\\_RELA\\_LANE\\_DIRECTION\\_TYPE](#)，与关联车道号对应，确保车道唯一性

*byRes3*

保留

*struPlateInfo*

车牌信息参数

*struVehicleInfo*

车辆信息参数

*\*pBuffer1*

当上传的是图片(近景图)信息时，指针指向图片信息，图片长度为 *dwPicLen*；当上传的是录像时，指针指向录像信息，录像长度为 *dwVideolen*

*\*pBuffer2*

当上传的是图片(车牌图)信息时，指针指向车牌小图片信息，车牌小图片的长度为 *dwPicPlateLen*

### Remarks

- 智能交通摄像机报警上传的信息是图片或者录像中的一种，可以通过判断图片和录像的长度是否为 0 以确定上传的是图片信息还是录像信息。图片数据为场景图片+车牌小图片。视频长度为 0xffffffff 时，表示视频内容异常，此时只上传报警信息，后面无视频内容，指向视频内容的指针为 NULL。
- iDS-65xx 手动抓拍只支持上传 *pBuffer1* 和 *pBuffer2*。
- 当 *byResultType* 为 2 时，可以用 *wAlarmRecordID* 作为查找条件，搜索报警录像。

## 7.99 NET\_DVR\_PLCCFG:车牌亮度补偿参数

```
struct{
    DWORD    dwSize;
    BYTE     byPlcEnable;
    BYTE     byPlateExpectedBright;
    BYTE     byRes1[2];
    BYTE     byTradeoffFlash;
```

```

BYTE    byCorrectFactor;
WORD    wLoopStatsEn;
WORD    wPlcBrightOffSet;
BYTE    byRes[18];
}NET_DVR_PLCCFG,*LPNET_DVR_PLCCFG;

```

### Members

*dwSize*

结构体大小

*byPlcEnable*

是否启用车牌亮度补偿（默认启用）：0-关闭，1-启用

*byPlateExpectedBright*

车牌的预期亮度（默认值 50），范围[0,100]

*byRes1*

保留

*byTradeoffFlash*

是否考虑闪光灯的影响(默认启用): 0-关闭，1-启用。（使用闪光灯补光时，如果考虑减弱闪光灯的亮度增强效应，则需要设为 1，否则为 0）

*byCorrectFactor*

纠正系数（默认值 50），范围[0,100]。默认值 50(在 *byTradeoffFlash* 切换时，恢复默认值)

*wLoopStatsEn*

触发线圈是否参与车牌亮度统计，按位表示，0-不统计，1-统计

*wPlcBrightOffSet*

车牌亮度补偿灵敏度(虚拟线圈模式起效)，取值范围：1~100

*byRes*

保留

## 7.100 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

## 7.101 NET\_DVR\_POSTEPOLICECFG:卡口电警参数

```

struct{
    DWORD    dwSize;
    DWORD    dwDistance;

```



```

DWORD   dwLightChan[MAX_SIGNALLIGHT_NUM];
BYTE    byCapSpeed;
BYTE    bySpeedLimit;
BYTE    byTrafficDirection;
BYTE    byRes[129];
}NET_DVR_POSTEPOLICECFG,*LPNET_DVR_POSTEPOLICECFG;

```

### Members

*dwSize*

结构体大小

*dwDistance*

线圈距离，单位：cm，取值范围：[0,20000]

*dwLightChan*

信号灯通道号

*byCapSpeed*

标志限速，单位 km/h，取值范围[0,255]

*bySpeedLimit*

限速值，单位 km/h，取值范围[0,255]

*byTrafficDirection*

车流方向，0-由东向西，1-由西向东，2-由南向北，3-由北向南

*byRes*

保留

## 7.102 NET\_DVR\_PPPCFG\_V30:PPP 参数结构体

```

struct{
    NET_DVR_IPADDR   struRemoteIP;
    NET_DVR_IPADDR   struLocalIP;
    char              sLocalIPMask[16];
    BYTE              sUsername[NAME_LEN];
    BYTE              sPassword[PASSWD_LEN];
    BYTE              byPPPMODE;
    BYTE              byRedial;
    BYTE              byRedialMode;
    BYTE              byDataEncrypt;
    DWORD             dwMTU;
    char              sTelephoneNumber[PHONENUMBER_LEN];
}NET_DVR_PPPCFG_V30,*LPNET_DVR_PPPCFG_V30;

```

### Members

*struRemoteIP*

远端 IP 地址

*struLocalIP*

本地 IP 地址

*sLocalIPMask*

本地 IP 地址掩码

*sUsername*

用户名

*sPassword*

密码

*byPPPMoDe*

PPP 模式：0—主动，1—被动

*byRedial*

是否回拨：0-否，1-是

*byRedialMoDe*

回拨模式：0-由拨入者指定，1-预置回拨号码

*byDataEncrypt*

数据加密：0-否，1-是

*dwMTU*

MTU

*sTelephoneNuMber*

电话号码

### 7.103 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

### 7.104 NET\_DVR\_PREVIEWINFO:预览参数

```
struct{
    LONG          lChannel;
    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，设备模拟通道号从 1 开始。

#### *dwStreamType*

码流类型：0-主码流，1-子码流，2-码流 3，3-虚拟码流，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。ANR 断网补录功能，客户端和设备之间网络异常恢复之后自动将前端数据同步过来，需要设备支持。

#### *byPreviewMode*

预览模式：0- 正常预览，1- 延迟预览

#### *byStreamID*

流 ID，为字母、数字和"\_"的组合，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。主子码流支持的取流协议通过登录返回结构参数 NET\_DVR\_DEVICEINFO\_V30 的 byMainProto、bySubProto 值得知。设备同时支持私协议和 RTSP 协议时，该参数才有效，默认使用私有协议，可选 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。
- Linux 64 位系统不支持软解码功能，因此需要将窗口句柄传 NULL，设置回调函数，只取流不解码显示。

## 7.105 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

## 7.106 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

## 7.107 NET\_DVR\_RECORDDAY\_V40:全天录像参数

```

struct{
    BYTE    byAllDayRecord;
    BYTE    byRecordType;
    char    byRes[62];
}NET_DVR_RECORDDAY_V40, *NET_DVR_RECORDDAY_V40;

```

### Members

#### *byAllDayRecord*

是否全天录像：0-否，1-是

#### *byRecordType*

录像类型：0-定时录像，1-移动侦测录像，2-报警录像，3-移动侦测或报警录像，4-移动侦测和报警录像，5-命令触发录像，6-智能报警录像，10-PIR 报警，11-无线报警，12-呼救报警，13-全部事件(移动侦测、PIR、无线、呼救等所有报警类型的"或")，14-智能交通事件，15-越界侦测，16-区域入侵侦测，17-音频异常侦测，18-场景变更侦测，19-智能侦测（越界侦测|区域入侵侦测|进入区域侦测|离开区域侦测|人脸侦测），20-人脸侦测，22-进入区域侦测，23-离开区域侦测，24-徘徊侦测，25-

人员聚集侦测, 26-快速运动侦测, 27-停车侦测, 28-物品遗留侦测, 29-物品拿取侦测, 30-火点检测

*byRes*

保留, 置为 0

#### Remarks

配置录像计划时, 如果录像类型设置为 13 (全部事件) 时, 按文件类型、时间方式查找时查找到的文件类型标志也为 13; 按事件方式查找时, 可以得到具体的录像事件类型。

## 7.108 NET\_DVR\_RECORDSCHED\_V40:时间段录像参数

```
struct{
    NET_DVR_SCHEDTIME    struRecordTime;
    BYTE                 byRecordType;
    BYTE                 byRes[31];
}NET_DVR_RECORDSCHED_V40, *LPNET_DVR_RECORDSCHED_V40;
```

#### Members

*struRecordTime*

录像时间段, 开始时间-结束时间

*byRecordType*

录像类型: 0-定时录像, 1-移动侦测录像, 2-报警录像, 3-移动侦测或报警录像, 4-移动侦测和报警录像, 5-命令触发录像, 6-智能报警录像, 10-PIR 报警, 11-无线报警, 12-呼救报警, 13-移动侦测、PIR、无线、呼救等所有报警类型的"或", 14-智能交通事件, 15-越界侦测, 16-区域入侵侦测, 17-音频异常侦测, 18-场景变更侦测, 19-智能侦测 (越界侦测|区域入侵侦测|进入区域侦测|离开区域侦测|人脸侦测), 20-人脸侦测, 22-进入区域侦测, 23-离开区域侦测, 24-徘徊侦测, 25-人员聚集侦测, 26-快速运动侦测, 27-停车侦测, 28-物品遗留侦测, 29-物品拿取侦测, 30-火点检测

*byRes*

保留, 置为 0

## 7.109 NET\_DVR\_RECORD\_V40:通道录像参数

```
struct{
    DWORD                dwSize;
    DWORD                dwRecord;
    NET_DVR_RECORDDAY_V40 struRecAllDay[MAX_DAYS];
    NET_DVR_RECORDSCHED_V40 struRecordSched[MAX_DAYS][MAX_TIMESEGMENT_V30];
    DWORD                dwRecordTime;
    DWORD                dwPreRecordTime;
    DWORD                dwRecorderDuration;
    BYTE                 byRedundancyRec;
    BYTE                 byAudioRec;
    BYTE                 byStreamType;
    BYTE                 byPassbackRecord;
    WORD                 wLockDuration;
```

```

BYTE          byRecordBackup;
BYTE          bySVCLLevel;
BYTE          byRecordManage;
BYTE          byExtraSaveAudio;
BYTE          byRes[126];

```

```
}NET_DVR_RECORD_V40, *LPNET_DVR_RECORD_V40;
```

### Members

#### *dwSize*

结构体大小

#### *dwRecord*

是否启用计划录像配置：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。超过这个时间，该录像文件将被强制删除；若设置为 0 天则不强制删除，除非文件被循环覆盖。获取参数时若为 0xffffffff，表示设备不支持设置该字段。

#### *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

#### *byStreamType*

码流类型：0- 主码流，1- 子码流，2- 主码流&子码流，3- 第三码流

#### *byPassbackRecord*

0- 不回传录像，1- 回传录像

#### *wLockDuration*

录像锁定时长，单位：小时，0 表示不锁定，0xffff 表示永久锁定。录像段的时长大于锁定的持续时长的录像，将不会锁定

#### *byRecordBackup*

0- 录像不存档，1- 录像存档，目前定时录像不存档

#### *bySVCLLevel*

SVC 抽帧类型：0- 不抽，1- 抽二分之一，2- 抽四分之三

#### *byRecordManage*

录像调度：0- 启用，1- 不启用。启用时进行定时录像；不启用时不进行定时录像，但是录像计划仍在使用的，比如移动侦测、回传都还在按这条录像计划进行

#### *byExtraSaveAudio*

是否单独存储音频录像：0- 不存储，1- 存储

#### *byRes*

保留，置为 0

## 7.110 NET\_DVR\_RELAY\_PARAM:继电器关联配置

```
struct{
    BYTE    byExternalDevInfo;
    BYTE    byRes[3];
}NET_DVR_RELAY_PARAM,*LPNET_DVR_RELAY_PARAM;
```

### Members

#### *byExternalDevInfo*

外接设备信息：0- 不接入设备，1- 开道闸，2- 关道闸，3- 停道闸，4- 报警信号，5- 常亮灯

#### *byRes*

保留，置为 0

## 7.111 NET\_DVR\_ROI\_DETECT\_CFG:ROI 检测区域配置

```
struct{
    DWORD                dwSize;
    DWORD                dwStreamType;
    BYTE                 byRoiDetectType;
    BYTE                 byRes[3];
    NET_DVR_ROI_DETECT_UNION  uRoiDetectInfo;
    BYTE                 byRes1[36];
}NET_DVR_ROI_DETECT_CFG,*LPNET_DVR_ROI_DETECT_CFG;
```

### Members

#### *dwSize*

结构体大小

#### *dwStreamType*

码流类型：0-主码流，1-子码流，2-事件类型，3-码流 3，.....

#### *byRoiDetectType*

区域检测类型具体定义如下：

```
enum _DETECT_ROI_TYPE_{
    DETECT_FIX_ROI    = 0,
    DETECT_TRACK_ROI = 1
}DETECT_ROI_TYPE
```

*DETECT\_FIX\_ROI*

检测 ROI 固定区域

*DETECT\_TRACK\_ROI*

检测 ROI 动态区域

#### *byRes*

保留

#### *uRoiDetectInfo*

ROI 检测区域

#### *byRes1*

保留

**Remarks**

- ROI: Region Of Interest, 感兴趣区域编码。用户可以通过启用 ROI 功能设置视频画面中最关心、最感兴趣的区域, 摄像机在进行视频编码时会提高相应区域的视频图像质量。
- `dwStreamType` (码流类型) 用于选择设置 ROI 的“主码流”、“子码流”、“第三码流”或者更多码流, 分别对不同码流设置不同的感兴趣区域, 不同码流相互独立。
- “固定区域”和“动态区域”为 ROI 的两种不同工作模式, 用户可根据需要设置固定的提升区域或允许摄像机动态提升侦测到人脸经过区域的画质。

**7.112 NET\_DVR\_ROI\_DETECT\_COND:ROI 检测区域信息**

```
struct{
    DWORD                dwSize;
    DWORD                dwRoiID;
    NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND struMultiStreamCfg;
    BYTE                 byRoiDetectType;
    BYTE                 byRoiDetectTrackType;
    BYTE                 byRes[30];
}NET_DVR_ROI_DETECT_COND,*LPNET_DVR_ROI_DETECT_COND;
```

**Members***dwSize*

结构体大小

*dwRoiID*

区域检测 ID 号

*struMultiStreamCfg*

多码流配置

*byRoiDetectType*

区域检测类型具体定义如下:

```
enum _DETECT_ROI_TYPE_{
    DETECT_FIX_ROI    = 0,
    DETECT_TRACK_ROI = 1
}DETECT_ROI_TYPE
```

*DETECT\_FIX\_ROI*

检测 ROI 固定区域

*DETECT\_TRACK\_ROI*

检测 ROI 动态区域

*byRoiDetectTrackType*

固定区域时, 该字段无效; 动态区域时, 类型: 1- 人脸侦测 (IPC), 2- 移动物体

*byRes*

保留

**7.113 NET\_DVR\_ROI\_DETECT\_NUM:ROI 检测区域编号数目信息**

```
struct{
```



```

DWORD    dwSize;
DWORD    dwGroup;
DWORD    dwStreamType;
DWORD    dwRoiFixNum;
DWORD    dwRoiFixID[MAX_ROIDETECT_NUM];
BYTE     szFixRoiName[MAX_ROIDETECT_NUM][NAME_LEN];
DWORD    dwRoiTrackNum;
DWORD    dwRoiTrackID[MAX_ROIDETECT_NUM];
BYTE     byRes[320];
}NET_DVR_ROI_DETECT_NUM,*LPNET_DVR_ROI_DETECT_NUM;

```

### Members

#### *dwSize*

结构体大小

#### *dwGroup*

组号

#### *dwStreamType*

码流类型：0-主码流，1-子码流，2-事件类型，3-码流 3，.....

#### *dwRoiFixNum*

固定区域总数，确定后续再次获取的组号

#### *dwRoiFixID*

ROI 固定区域 ID，IPC 现在支持 4 个，球机现在支持 24 个(根据能力集可以扩展支持数量)

#### *szFixRoiName*

ROI 固定区域名称

#### *dwRoiTrackNum*

ROI 动态区域总数，确定后续再次获取的组号

#### *dwRoiTrackID*

ROI 动态区域 ID，IPC 现在支持 4 个，球机现在支持 24 个(根据能力集可以扩展支持数量)

#### *byRes*

保留

### Remarks

- ROI: Region Of Interest, 感兴趣区域编码。用户可以通过启用 ROI 功能设置视频画面中最关心、最感兴趣的区域，摄像机在进行视频编码时会提高相应区域的视频图像质量。
- *dwGroup* 是组号的概念，如果后续扩展超出每组 *MAX\_ROIDETECT\_NUM* 个区域的时候，需要分组去获取。第一次获取的时候，组号为 1 或 0；当 *dwRoiFixNum* 或者 *dwRoiTrackNum* 数目超过 *MAX\_ROIDETECT\_NUM* 的时候，进行分组获取，组号依次递增。

## 7.114 NET\_DVR\_ROI\_DETECT\_UNION:ROI 检测区域配置联合体

```

struct{
    DWORD                uLen[128];
    NET\_DVR\_ROI\_FIX\_RECT\_CFG    strRoiFixRectCfg;
    NET\_DVR\_ROI\_TRACK\_RECT\_CFG  strRoiTrackRectCfg;
}NET_DVR_ROI_DETECT_UNION,*LPNET_DVR_ROI_DETECT_UNION;

```

### Members

*uLen*

联合体大小，512 字节

*strRoiFixRectCfg*

固定区域模式

*strRoiTrackRectCfg*

自动区域模式

#### Remarks

- ROI: Region Of Interest, 感兴趣区域编码。
- 当前设备在单场景下支持 4 个绘画区域 (区域检测号); 球机多场景下一共支持 24 个绘画区域 (区域检测号)。

## 7.115 NET\_DVR\_ROI\_FIX\_RECT\_CFG:ROI 固定区域模式配置

```
struct{
    BYTE            byEnableFixRoi;
    BYTE            byImageQualityLevel;
    BYTE            byRes[2];
    BYTE            byFixRoiName[NAME_LEN];
    NET_VCA_RECT    struRoiRect;
    BYTE            byRes1[468];
}NET_DVR_ROI_FIX_RECT_CFG,*LPNET_DVR_ROI_FIX_RECT_CFG;
```

#### Members

*byEnableFixRoi*

ROI 固定区域模式是否启用: 0- 不启用, 1- 启用

*byImageQualityLevel*

图像质量提升等级, 取值范围: 1~6

*byRes*

保留

*byFixRoiName*

ROI 固定区域名称

*struRoiRect*

ROI 区域

*byRes1*

保留

## 7.116 NET\_DVR\_ROI\_TRACK\_RECT\_CFG:ROI 动态模式配置

```
struct{
    BYTE            byEnableTrackRoi;
    BYTE            byImageQualityLevel;
    BYTE            byModeType;
    BYTE            byRes1[509];
}NET_DVR_ROI_TRACK_RECT_CFG,*LPNET_DVR_ROI_TRACK_RECT_CFG;
```

**Members***byEnableTrackRoi*

ROI 动态模式是否启用：0- 不启用，1- 启用

*byImageQualityLevel*

图像质量提升等级，取值范围：1~6

*byModeType*

动态模式：0- 保留，1- 人脸模式，2- 移动物体

*byRes1*

保留

**7.117 NET\_DVR\_RS232CFG\_V30:RS232 串口参数结构体**

```
struct{
    DWORD                dwSize;
    NET_DVR_SINGLE_RS232 struRs232[MAX_SERIAL_PORT];
    NET_DVR_PPPCFG_V30  struPPPConfig;
}NET_DVR_RS232CFG_V30, *LPNET_DVR_RS232CFG_V30;
```

**Members***dwSize*

结构体大小

*struRs232*

RS232 串口参数，设备只有一个 232 串口，采用第一个取 struRs232[0]即可

*struPPPConfig*

PPP 参数

**7.118 NET\_DVR\_SCHEDTIME:起止时间段参数**

```
struct{
    BYTE    byStartHour;
    BYTE    byStartMin;
    BYTE    byStopHour;
    BYTE    byStopMin;
}NET_DVR_SCHEDTIME, *LPNET_DVR_SCHEDTIME;
```

**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

## 7.119 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

*dwMaxLoginNum*

最大注册用户数

*dwMaxRealPlayNum*

最大实时预览的路数

*dwMaxPlayBackNum*

最大回放或下载的路数

*dwMaxAlarmChanNum*

最大建立报警通道的路数

*dwMaxFormatNum*

最大硬盘格式化的路数

*dwMaxFileSearchNum*

最大文件搜索的路数

*dwMaxLogSearchNum*

最大日志搜索的路数

*dwMaxSerialNum*

最大建立透明通道的路数

*dwMaxUpgradeNum*

最大升级的路数

*dwMaxVoiceComNum*

最大语音转发的路数

*dwMaxBroadCastNum*

最大语音广播的路数

*dwRes*

保留，置为 0

## 7.120 NET\_DVR\_SDKLOCAL\_CFG:模拟能力集参数

```
struct{
    BYTE    byEnableAbilityParse;
    BYTE    byVoiceComMode;
    BYTE    byRes[510];
}NET_DVR_SDKLOCAL_CFG,*LPNET_DVR_SDKLOCAL_CFG;
```

### Members

#### *byEnableAbilityParse*

使用能力集解析库：0- 不使用，1- 使用，默认不使用

#### *byVoiceComMode*

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

#### *byRes*

保留

## 7.121 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

## 7.122 NET\_DVR\_SECURITY\_CFG:安全认证配置

```
struct{
    DWORD    dwSize;
    BYTE     byRes1[3];
    BYTE     byWebAuthentication;
    BYTE     byRtspAuthentication;
    BYTE     byTelnetServer;
    BYTE     bySSHServer;
    BYTE     byIllegalLoginLock;
    BYTE     byRes[28];
}NET_DVR_SECURITY_CFG,*LPNET_DVR_SECURITY_CFG;
```

### Members

*dwSize*

结构体大小

*byRes1*

保留，置为 0

*byWebAuthentication*

web 认证配置：0- Digest，1- basic(默认)

*byRtspAuthentication*

rtsp 认证配置：0- 禁用，1-basic(默认)

*byTelnetServer*

telnet 设置：0- 禁用，1- 启用

*bySSHServer*

SSH 设置：0- 禁用（默认），1- 启用

*byIllegalLoginLock*

开启登录锁定：0- 启用（默认），1- 禁用

*byRes*

保留，置为 0

### Remarks

- 安全认证配置，只有管理员用户（admin）才有权限。
- 安全认证参数配置能力，对应安全认证配置能力集(SecurityAbility)，相关接口：[NET\\_DVR\\_GetDeviceAbility](#)(能力集类型：DEVICE\_ABILITY\_INFO)。

### 7.123 NET\_DVR\_SERIAL\_CATCHPIC\_PARA:串口抓图参数

```
struct{
    BYTE        byStrFlag;
    BYTE        byEndFlag;
    WORD        wCardIdx;
    DWORD       dwCardLen;
    DWORD       dwTriggerPicChans;
}NET_DVR_SERIAL_CATCHPIC_PARA, *LPNET_DVR_SERIAL_CATCHPIC_PARA;
```

#### Members

##### *byStrFlag*

串口数据开始符

##### *byEndFlag*

串口数据结束符

##### *wCardIdx*

卡号相对起始位置

##### *dwCardLen*

卡号长度

##### *dwTriggerPicChans*

所触发的通道号，按位，从第 1 位开始计，即 0x2 表示第一通道

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

```
struct{
    DWORD       dwSize;
    BYTE        byLevel;
    BYTE        byAlarmInfoType;
    BYTE        byRetAlarmTypeV40;
    BYTE        byRetDevInfoVersion;
    BYTE        byRetVQDAlarmType;
    BYTE        byFaceAlarmDetection;
    BYTE        bySupport;
    BYTE        byRes[9];
}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）

**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**

保留，置为 0

**byRetDevInfoVersion**

保留，置为 0

**byRetVQDAlarmType**

保留，置为 0

**byFaceAlarmDetection**

保留，置为 0

**bySupport**

按位表示，值：0-上传，1-不上传

bit0- 表示二级布防是否上传图片

**byRes**

保留，置为 0

**Remarks**

- 一级布防最大连接数为 1 个，二级最大连接数为 3 个，三级最大连接数为 5 个，设备支持一级、二级和三级布防同时进行，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。布防时设置 **byAlarmInfoType** 值为 0 或者 1，则 [NET\\_DVR\\_SetDVRMessageCallBack\\_V31](#) 设置的回调或者函数中上传的报警信息类型（ICommand）对应 COMM\_UPLOAD\_PLATE\_RESULT 或者 COMM\_ITS\_PLATE\_RESULT。

**7.125 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

## 7.126 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]

## 7.127 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)

#### *byHDAAttr*

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。

## 7.128 NET\_DVR\_SINGLE\_NFS:NFS 配置子结构

```
struct{
    char    sNfsHostIPAddr[16];
    BYTE    sNfsDirectory[PATHNAME_LEN];
}NET_DVR_SINGLE_NFS, *LPNET_DVR_SINGLE_NFS;
```

#### Members

##### *sNfsHostIPAddr*

网络文件系统主机 IP 地址

##### *sNfsDirectory*

NFS 路径

## 7.129 NET\_DVR\_SINGLE\_RS232:RS232 串口参数子结构

```
struct{
    DWORD        dwBaudRate;
    BYTE         byDataBit;
    BYTE         byStopBit;
    BYTE         byParity;
    BYTE         byFlowcontrol;
    DWORD        dwWorkMode;
}NET_DVR_SINGLE_RS232, *LPNET_DVR_SINGLE_RS232;
```

### Members

#### *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

#### *dwWorkMode*

工作模式, 0—窄带传输(232 串口用于 PPP 拨号), 1—控制台(232 串口用于参数控制), 2—透明通道, 3-ptz 模式 (审讯温湿度传感器), 4-报警盒模式

## 7.130 NET\_DVR\_SIGNALLAMP\_DETCFG:信号灯异常检测信息

```
struct{
    DWORD        dwSize;
    BYTE         byAbsTime[32];
    NET\_DVR\_IPADDR struAlarmCamIP;
    DWORD        dwPic1Len;
    DWORD        dwPic2Len;
    char         *pPic1Buffer;
    char         *pPic2Buffer;
    BYTE         byRes[128];
}NET_DVR_SIGNALLAMP_DETCFG, *LPNET_DVR_SIGNALLAMP_DETCFG;
```

### Members

#### *dwSize*

结构体大小

#### *byAbsTime*

绝对时间点, yyyyymmddhhmmssxxx, e.g.20090810235959999, 最后三位为毫秒数

*struAlarmCamIP*

报警摄像机 IP 地址

*dwPic1Len*

异常图片 1 数据长度

*dwPic2Len*

异常图片 2 数据长度

*pPic1Buffer*

数据缓冲区指针 1, 为叠加车道线的图片

*pPic2Buffer*

数据缓冲区指针 2, 为叠加信号灯状态的图片

*byRes*

保留

### 7.131 NET\_DVR\_SMARTIR\_PARAM:SMART IR(防过曝)配置参数

```
struct{
    BYTE    byMode;
    BYTE    byIRDistance;
    BYTE    byShortIRDistance;
    BYTE    byLongIRDistance;
}NET_DVR_SMARTIR_PARAM, *LPNET_DVR_SMARTIR_PARAM;
```

#### Members

*byMode*

SMART IR 模式: 0- 自动, 1- 手动

*byIRDistance*

红外距离等级(等级越高, 红外距离越远): 1~100, 默认:50, 手动模式下可修改

*byShortIRDistance*

近光灯距离等级, 取值范围: 1~100

*byLongIRDistance*

远光灯距离等级, 取值范围: 1~100

### 7.132 NET\_DVR\_SNAPCFG:触发参数

```
struct{
    DWORD   dwSize;
    BYTE    byRelatedDriveWay;
    BYTE    bySnapTimes;
    WORD    wSnapWaitTime;
    WORD    wIntervalTime[MAX_INTERVAL_NUM];
    DWORD   dwSnapVehicleNum;
    BYTE    byRes2[20];
}NET_DVR_SNAPCFG, *LPNET_DVR_SNAPCFG;
```

#### Members

*dwSize*

结构体大小

*byRelatedDriveWay*

触发 IO 关联的车道号，取值范围[0,9]

*bySnapTimes*

线圈抓拍次数，0-不抓拍，非 0-连拍次数，目前最大 5 次

*wSnapWaitTime*

抓拍等待时间，单位 ms，取值范围[0,60000]

*wIntervalTime*

连拍间隔时间，单位 ms，取值范围[67,60000]

*dwSnapVehicleNum*

抓拍匹配序号，与车牌报警上传 [NET ITS PLATE RESULT](#) 中 dwMatchNo 字段对应，网络触发连拍时有效

*byRes2*

保留，置为 0

### 7.133 NET\_DVR\_SNAP\_CAMERAPARAMCFG:智能交通摄像机 CCD 参数

struct{

```

    BYTE                byWDRMode;
    BYTE                byWDRType;
    BYTE                byWDRLevel;
    BYTE                byRes1;
    NET\_DVR\_TIME\_EX    struStartTime;
    NET\_DVR\_TIME\_EX    struEndTime;
    BYTE                byDayNightBrightness;
    BYTE                byMCEEnabled;
    BYTE                byMCELevel;
    BYTE                byAutoContrastEnabled;
    BYTE                byAutoContrastLevel;
    BYTE                byLSEDetailEnabled;
    BYTE                byLSEDetailLevel;
    BYTE                byLPDEEnabled;
    BYTE                byLPDELevel;
    BYTE                byRes[35];

```

}NET\_DVR\_SNAP\_CAMERAPARAMCFG, \*LPNET\_DVR\_SNAP\_CAMERAPARAMCFG;

#### Members

*byWDRMode*

宽动态模式：0- 关闭，1- 数字宽动态，2- 宽动态态

*byWDRType*

宽动态切换模式：0- 强制启用，1- 按时间启用，2- 按亮度启用

*byWDRLevel*

宽动态等级，索引 0~6 分别对应等级 1~7，默认索引 2（即 3 级）

*byRes1*

保留

*struStartTime*

开始宽动态时间

*struEndTime*

结束宽动态时间

*byDayNightBrightness*

日夜转换亮度阈值，取值范围：0~100，默认：50

*byMCEEnabled*

记忆色增强使能：0- 关闭，1- 开启

*byMCELevel*

记忆色增强强度，取值范围：0~100，默认值：50

*byAutoContrastEnabled*

自动对比度使能：0- 关闭，1- 开启

*byAutoContrastLevel*

自动对比等级，取值范围：0~100，默认值：50

*byLSEDetailEnabled*

细节增强使能：0- 关闭，1- 开启

*byLSEDetailLevel*

细节增强等级，取值范围：0~100，默认值：50

*byLPDEEnabled*

车牌增强使能：0- 关闭，1- 开启

*byLPDELevel*

车牌增强等级，取值范围：0~100，默认值：50

*byRes*

保留

## 7.134 NET\_DVR\_SNAP\_ABILITY:智能交通 IPC 能力集参数

```
struct{
    DWORD    dwSize;
    BYTE     byIoInNum;
    BYTE     byIoOutNum;
    BYTE     bySingleSnapNum;
    BYTE     byLightModeArrayNum;
    BYTE     byMeasureModeArrayNum;
    BYTE     byPlateEnable;
    BYTE     byLensMode;
    BYTE     byPreTriggerSupport;
    DWORD    dwAbilityType;
    BYTE     byIoSpeedGroup;
    BYTE     byIoLightGroup;
    BYTE     byRecogRegionType;
    BYTE     bySupport;
    WORD     wSupportMultiRadar;
```

```

BYTE    byICRPresetNum;
BYTE    byICRTIMEslot;
BYTE    bySupportRS485Num;
BYTE    byExpandRs485SupportSensor;
BYTE    byExpandRs485SupportSignalLampDet;
BYTE    byRelayNum;
BYTE    bySupport1;
BYTE    bySupport2;
BYTE    bySupportWhiteBalance;
BYTE    byRes[10];

```

```
}NET_DVR_SNAP_ABILITY, *LPNET_DVR_SNAP_ABILITY;
```

### Members

*dwSize*

结构体大小

*byIoInNum*

IO 输入口数

*byIoOutNum*

IO 输出口数

*bySingleSnapNum*

单 IO 触发组数

*byLightModeArrayNum*

红绿灯模式组数

*byMeasureModeArrayNum*

测速模式组数

*byPlateEnable*

车牌识别能力, 0-没有, 1-有

*byLensMode*

前端镜头模式: 0-CCD, 1-CMOS

*byPreTriggerSupport*

是否支持原触发模式: 0- 支持, 1- 不支持

*dwAbilityType*

支持的触发模式能力, 按位表示, 定义如下所示:

```

enum _ITC_MAINMODE_ABILITY_{
    ITC_MODE_UNKNOW    = 0x0,
    ITC_POST_MODE      = 0x1,
    ITC_EPOLICE_MODE    = 0x2,
    ITC_POSTEPOLICE_MODE = 0x4
}ITC_MAINMODE_ABILITY

```

*ITC\_MODE\_UNKNOW*

无

*ITC\_POST\_MODE*

卡口模式

*ITC\_EPOLICE\_MODE*

电警模式

*ITC\_POSTEPOLICE\_MODE*

自卡式电警模式动

*byIoSpeedGroup*

支持的 IO 测速组数

*byIoLightGroup*

支持的 IO 红绿灯组数

*byRecogRegionType*

牌识区域支持的类型，定义如下所示：

```
enum _ITC_RECOG_REGION_TYPE_{
    ITC_REGION_RECT          = 0x0,
    ITC_REGION_POLYGON      = 0x1
}ITC_RECOG_REGION_TYPE
```

*ITC\_REGION\_RECT*

矩形

*ITC\_REGION\_POLYGON*

多边形

*bySupport*

设备能力，按位表示，值：0-不支持，1-支持

*bySupport&0x1*，表示是否支持扩展的字符叠加配置([NET ITS OVERLAP\\_CFG](#))

*bySupport&0x2*，表示是否支持扩展的校时配置结构([NET DVR CALIBRATE\\_TIME](#))

*bySupport&0x4*，表示是否支持多网卡(多网隔离)([NET DVR NETCFG\\_MULTI](#))

*bySupport&0x8*，表示是否支持网卡的 bonding 功能(网络容错)([NET DVR NETWORK\\_BONDING](#))

*bySupport&0x10*，表示是否支持语音对讲

*bySupport&0x20*，表示是否支持单 IO 触发界面 IO 使能配置([NET ITC SINGLEIO\\_PARAM](#) 中参数 *byRelatedIOOutEx*)

*bySupport&0x40*，表示是否支持扩展的 FTP 接口([NET ITC FTP\\_CFG](#))

*bySupport&0x80*，表示是否支持云存储接口([NET DVR CLOUDSTORAGE\\_CFG](#))

*wSupportMultiRadar*

设备能力（RS485 口关联雷达能力），按位表示，值：0-不支持，1-支持

*wSupportMultiRadar&0x1*，表示卡口 RS485 雷达是否支持车道关联雷达处理

*wSupportMultiRadar&0x2*，表示卡口虚拟线圈是否支持车道关联雷达处理

*wSupportMultiRadar&0x8*，表示视频检测是否支持车道关联雷达处理

*byICRPresetNum*

表示支持的 ICR 预置点（滤光片偏移点）数

*byICRTimeSlot*

表示支持的 ICR 的时间段数（1~8）

*bySupportRS485Num*

表示支持的 RS485 口的数量

*byExpandRs485SupportSensor*

设备能力（RS485 口关联车检器能力），按位表示，值：0-不支持，1-支持

*byExpandRs485SupportSensor&0x1*，表示电警车检器是否支持车检器

*byExpandRs485SupportSensor&0x2*，表示卡式电警车检器是否支持车检器

*byExpandRs485SupportSignalLampDet*

设备能力（RS485 口关联信号灯检测器能力），按位表示，值：0-不支持，1-支持



byExpandRs485SupportSignalLampDet&0x1, 表示电警车检器是否支持外接信号灯检测器  
 byExpandRs485SupportSignalLampDet&0x2, 表示卡式电警车检器是否支持外接信号灯检测器  
 byExpandRs485SupportSignalLampDet&0x4, 表示视频电警是否支持外接信号灯检测器

#### bySupport1

设备能力（扩展），按位表示，值：0-不支持，1-支持

bySupport1&0x1, 表示是否支持自定义校时信息配置（[NET\\_DVR\\_CALIBRATE\\_TIME](#)）

bySupport1&0x2, 表示是否支持视频触发优化方案（[NET\\_ITC\\_POST\\_VTCOIL\\_PARAM](#)）

bySupport1&0x4, 表示是否支持 DDNS 配置（[NET\\_DVR\\_DDNSPARA\\_V30](#)）

bySupport1&0x8, 表示是否支持获取设备当前触发模式接口命令（[NET\\_DVR\\_CURTRIGGERMODE](#)）

bySupport1&0x10, 表示是否支持出入口配置（[NET\\_DVR\\_ENTRANCE\\_CFG](#)）

bySupport1&0x20, 表示是否支持 GPS 模块参数配置（[NET\\_DVR\\_GPS\\_DATA\\_CFG](#)）

bySupport1&0x40, 表示是否支持 NTP 校时间隔支持分钟设置（[NET\\_DVR\\_NTPPARA](#)）

bySupport1&0x80, 表示是否支持 TPS 报警数据上传（COMM\_ALARM\_TPS\_REAL\_TIME 和 COMM\_ALARM\_TPS\_STATISTICS）

#### bySupport2

设备能力（扩展），按位表示，值：0-不支持，1-支持

bySupport2&0x01, 表示是否支持码流附加信息配置（[NET\\_DVR\\_STREAM\\_ATTACHINFO\\_CFG](#)）

bySupport2&0x02, 表示是否支持通道图像参数（[NET\\_DVR\\_PICCFG\\_V30](#)）中视频叠加时间毫秒使能参数 (byOSDMilliSecondEnable)

#### bySupportWhiteBalance

白平衡能力（默认都是支持手动白平衡、自动白平衡、自动白平衡 2 这 3 种模式），按位表示，值：0-不支持，1-支持

bySupportWhiteBalance &0x01, 表示是否支持日光灯模式

bySupportWhiteBalance &0x02, 表示是否支持自然光模式

bySupportWhiteBalance &0x04, 表示是否支持暖光灯模式

bySupportWhiteBalance &0x08, 表示是否支持白炽灯模式

#### byRes

保留，置为 0

#### Remarks

V3.0(含)之前版本仅支持老的触发模式 (byPreTriggerSupport=0)，不支持新的触发模式 (dwAbilityType = 0)。

## 7.135 NET\_DVR\_SNAPENABLECFG:使能参数

```
struct{
  DWORD    dwSize;
  BYTE     byPlateEnable;
  BYTE     byRes1[2];
  BYTE     byFrameFlip;
  WORD     wFlipAngle;
  WORD     wLightPhase;
  BYTE     byLightSyncPower;
  BYTE     byFrequency;
  BYTE     byUploadSDEnable;
```

```

BYTE    byPlateMode;
BYTE    byUploadInfoFTP;
BYTE    byAutoFormatSD;
WORD    wJpegPicSize;
BYTE    bySnapPicResolution;
BYTE    byRes[55];
}NET_DVR_SNAPENABLECFG, *LPNET_DVR_SNAPENABLECFG;

```

### Members

#### *dwSize*

结构体大小

#### *byPlateEnable*

是否支持车牌识别：0- 不支持，1- 支持

#### *byRes1*

保留，置为 0

#### *byFrameFlip*

图像是否翻转：0- 不翻转，1- 翻转

#### *wFlipAngle*

图像翻转角度：0, 90, 180, 270

#### *wLightPhase*

相位，取值范围[0, 360]

#### *byLightSyncPower*

是否信号灯电源同步：0- 不同步，1- 同步

#### *byFrequency*

信号频率

#### *byUploadSDEnable*

是否自动上传 SD 图片：0- 否，1- 是

#### *byPlateMode*

识别模式参数（仅 iDS-65xx 支持）：0- 视频触发，1- 外部触发。

#### *byUploadInfoFTP*

是否上传抓拍附加信息到 FTP：0- 否，1- 是

#### *byAutoFormatSD*

是否自动格式化 SD 卡：0- 否，1- 是

#### *wJpegPicSize*

Jpeg 图片大小，取值范围：64~8196，单位：KB，用于 V3.5 或以上版本的智能交通摄像机的图像质量配置

#### *bySnapPicResolution*

抓拍图片分辨率(与 NET\_DVR\_COMPRESSION\_INFO\_V30 中分辨率保持一致):

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), 22-1360\*1024, 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, 60-2752\*2208, 61-384\*288, 62-4000\*3000, 63-4096\*2160, 64-3840\*2160, 65-4000\*2250, 66-3072\*1728, 67-2592\*1944, 68-2464\*1520, 69-1280\*1920, 70-2560\*1440, 71-1024\*1024, 72-160\*128, 73-324\*240, 74-324\*256, 75-336\*256, 76-640\*512, 77-2720\*2048, 78-384\*256, 79-384\*216, 80-320\*256, 81-320\*180, 82-320\*192, 83-512\*384, 84-325\*256, 85-256\*192, 86-640\*360, 0xff-Auto(使用当前码流分辨率)

*byRes*

保留, 置为 0

### Remarks

使能参数配置能力: [NET\\_DVR\\_GetDeviceAbility](#)(能力集类型: DEVICE\_ABILITY\_INFO), 获取抓拍机和智能终端能力(ITDeviceAbility), 对应节点: <SnapEnableCfg>。

## 7.136 NET\_DVR\_SPRCFG:车牌识别参数

```
struct{
    DWORD          dwSize;
    BYTE           byDefaultCHN[MAX_CHJC_NUM];
    BYTE           byPlateOSD;
    BYTE           bySendJPEG1;
    BYTE           bySendJPEG2;
    WORD           wDesignedPlateWidth;
    BYTE           byTotalLaneNum;
    BYTE           byRes1;
    WORD           wRecognizedLane;
    NET\_VCA\_RECT struLaneRect[MAX_LANERECT_NUM];
    DWORD          dwRecogMode;
    BYTE           bySendPRRaw;
    BYTE           bySendBinImage;
    BYTE           byDelayCapture;
    BYTE           byUseLED;
    BYTE           byRes2[68];
}NET_DVR_SPRCFG, *LPNET_DVR_SPRCFG;
```

### Members

*dwSize*

结构体大小

*byDefaultCHN*

设备运行省份的汉字简写

*byPlateOSD*

0-不叠加车牌彩色图, 1-叠加车牌彩色图

*bySendJPEG1*

0-不传送近景 JPEG 图,1-传送近景 JPEG 图

*bySendJPEG2*

0-不传送远景 JPEG 图,1-传送远景 JPEG 图

*wDesignedPlateWidth*

车牌设计宽度

*byTotalLaneNum*

识别的车道数

*byRes1*

保留，置为 0

*wRecognizedLane*

识别的车道号，按位表示，bit0 表示车道 1 是否识别，0-不识别，1-识别

*struLaneRect*

车道识别区域

*dwRecogMode*

识别的类型，

bit0-背向识别：0-正向车牌识别，1-背向识别(尾牌识别)；

bit1-大牌识别或小车牌识别：0-小车牌识别，1-大牌识别；

bit2-车身颜色识别：0-不采用车身颜色识别，在背向识别或小车牌识别时禁止启用，1-车身颜色识别；

bit3-农用车识别：0-不采用农用车识别，1-农用车识别；

bit4-模糊识别：0-不采用模糊识别，1-模糊识别；

bit5-帧定位或场定位：0-帧定位，1-场定位；

bit6-帧识别或场识别：0-帧识别，1-场识别；

bit7-晚上或白天：0-白天，1-晚上

*bySendPRRaw*

是否发送车辆原图（仅 iDS-65xx 支持）：0-不发送，1-发送

*bySendBinImage*

是否发送车牌二值图（仅 iDS-65xx 支持）：0-不发送，1-发送

*byDelayCapture*

延时抓拍控制（仅 iDS-65xx 支持），单位：帧

*byUseLED*

是否使用 LED 控制（仅 iDS-65xx 支持）：0-否，1-是

*byRes2*

保留，置为 0

## 7.137 NET\_DVR\_STATUS\_DETECTCFG:状态检测使能参数

```
struct{
    DWORD    dwSize;
    BYTE     byEnableTrigIODetect;
    BYTE     byEnableFlashOutDetect;
    BYTE     byEnableRS485Detect;
    BYTE     byEnableTrafficLightDetect;
    BYTE     byRes[28];
}NET_DVR_STATUS_DETECTCFG,*LPNET_DVR_STATUS_DETECTCFG;
```

**Members**

*dwSize*

结构体大小

*byEnableTrigIODetect*

开启触发 IO 状态检测：1- 是，0- 否

*byEnableFlashOutDetect*

开启同步输出状态检测：1- 是，0- 否

*byEnableRS485Detect*

开启 RS485 接收状态检测：1- 是，0- 否

*byEnableTrafficLightDetect*

开启交通灯状态检测，1- 是，0- 否

*byRes*

保留

## 7.138 NET\_DVR\_STREAM\_ATTACHINFO\_CFG:码流附加信息配置

```
struct{
    DWORD    dwSize;
    BYTE     byStreamWithVca;
    BYTE     byRes[127];
}NET_DVR_STREAM_ATTACHINFO_CFG, *LPNET_DVR_STREAM_ATTACHINFO_CFG;
```

### Members

*dwSize*

结构体大小

*byStreamWithVca*

码流中叠加智能信息：0- 不叠加，1- 叠加

*byRes*

保留

## 7.139 NET\_DVR\_STREAM\_INFO:流信息

```
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，为字母、数字和"\_"的组合。智能交通摄像机不支持流 ID 标识功能，设为 0

*dwChannel*

设备通道号

*byRes*

保留，置为 0

## 7.140 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

## 7.141 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

**7.142 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

## 7.143 NET\_DVR\_TIME\_V30:时间参数(扩展)

```

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

## 7.144 NET\_DVR\_TIMEPOINT:时间点参数结构体

```

struct{
    DWORD    dwMonth;
    DWORD    dwWeekNo;
    DWORD    dwWeekDate;
    DWORD    dwHour;
    DWORD    dwMin;
}NET_DVR_TIMEPOINT, *LPNET_DVR_TIMEPOINT;

```

### Members



**dwMonth**

月：[0,11]取值分别表示第 1 个月到第 12 个月

**dwWeekNo**

周：0—第 1 周，1—第 2 周，2—第 3 周，3—第 4 周，4—最后一周

**dwWeekDate**

星期：0—星期日，1—星期一，2—星期二，3—星期三，4—星期四，5—星期五，6—星期六

**dwHour**

小时：开始时间 0~23，结束时间 1~23

**dwMin**

分：0~59

## 7.145 NET\_DVR\_TPS\_ALARMCFG:交通统计报警参数结构体

```
struct{
    DWORD    dwSize;
    BYTE     byEnable;
    BYTE     byRes1[3];
    DWORD    dwInterval;
    BYTE     byRes[248];
}NET_DVR_TPS_ALARMCFG,*LPNET_DVR_TPS_ALARMCFG;
```

### Members

**dwSize**

结构体大小

**byEnable**

是否启用交通统计报警上传

**byRes1**

保留

**dwInterval**

统计间隔时间，单位：分钟

**byRes**

保留

## 7.146 NET\_DVR\_TPS\_LANE\_PARAM:单车道 TPS 交通统计信息

```
struct{
    BYTE     byLane;
    BYTE     bySpeed;
    BYTE     byRes[2];
    DWORD    dwLightVehicle;
    DWORD    dwMidVehicle;
    DWORD    dwHeavyVehicle;
    DWORD    dwTimeHeadway;
    DWORD    dwSpaceHeadway;
```

```

float      fSpaceOccupyRation;
float      fTimeOccupyRation;
BYTE      byRes1[16];
}NET_DVR_TPS_LANE_PARAM, *LPNET_DVR_TPS_LANE_PARAM;

```

### Members

#### *byLane*

对应车道号

#### *bySpeed*

车道过车平均速度

#### *byRes*

保留

#### *dwLightVehicle*

小型车数量

#### *dwMidVehicle*

中型车数量

#### *dwHeavyVehicle*

重型车数量

#### *dwTimeHeadway*

车头时距，以秒计算

#### *dwSpaceHeadway*

车头间距，以米来计算

#### *fSpaceOccupyRation*

空间占有率，百分比计算

#### *fTimeOccupyRation*

时间占有率，百分比计算

#### *byRes1*

保留

## 7.147 NET\_DVR\_TPS\_PARAM:TPS 交通参数信息

```

struct{
    BYTE    byStart;
    BYTE    byCMD;
    BYTE    byRes[2];
    WORD    wDeviceID;
    WORD    wDataLen;
    BYTE    byLane;
    BYTE    bySpeed;
    BYTE    byLaneState;
    BYTE    byQueueLen;
    BYTE    byRes1[24];
}NET_DVR_TPS_PARAM, *LPNET_DVR_TPS_PARAM;

```

### Members

#### *byStart*

开始码(串口命令)

*byCMD*

命令号: 01- 进入指令, 02- 离开指令, 03- 拥堵状态指令(为 03 时, 只有 *byLaneState* 和 *byQueueLen* 有效)

*byRes*

保留

*wDeviceID*

设备 ID

*wDataLen*

数据长度(串口信息)

*byLane*

对应车道号

*bySpeed*

对应车速, 单位: Km/h

*byLaneState*

车道状态: 0- 无状态, 1- 畅通, 2- 拥挤, 3- 堵塞

*byQueueLen*

堵塞状态下排队长度, 单位: 米, 比如 50 米

*byRes1*

保留

## 7.148 NET\_DVR\_TPS\_REAL\_TIME\_INFO:TPS 实时过车数据信息

```
struct{
    DWORD          dwSize;
    DWORD          dwChan;
    NET\_DVR\_TIME\_V30   struTime;
    NET\_DVR\_TPS\_PARAM struTPSRealTimeInfo;
    BYTE          byRes[24];
}NET_DVR_TPS_REAL_TIME_INFO, *LPNET_DVR_TPS_REAL_TIME_INFO;
```

### Members

*dwSize*

结构体大小

*dwChan*

通道号

*struTime*

检测时间

*struTPSRealTimeInfo*

交通参数实时信息

*byRes*

保留

## 7.149 NET\_DVR\_TPS\_STATISTICS\_INFO:TPS 统计过车数据信息

```

struct{
    DWORD                dwSize;
    DWORD                dwChan;
    NET\_DVR\_TPS\_STATISTICS\_PARAM    struTPSStatisticsInfo;
    BYTE                byRes[128];
}NET_DVR_TPS_STATISTICS_INFO, *LPNET_DVR_TPS_STATISTICS_INFO;

```

### Members

*dwSize*

结构体大小

*dwChan*

通道号

*struTPSStatisticsInfo*

交通参数统计信息

*byRes*

保留

## 7.150 NET\_DVR\_TPS\_STATISTICS\_PARAM:TPS 交通统计信息

```

struct{
    BYTE                byStart;
    BYTE                byCMD;
    BYTE                byRes[2];
    WORD                wDeviceID;
    WORD                wDataLen;
    BYTE                byTotalLaneNum;
    BYTE                byRes1[15];
    NET\_DVR\_TIME\_V30        struStartTime;
    DWORD                dwSamplePeriod;
    NET\_DVR\_TPS\_LANE\_PARAM    struLaneParam[MAX_TPS_RULE];
}NET_DVR_TPS_STATISTICS_PARAM, *LPNET_DVR_TPS_STATISTICS_PARAM;

```

### Members

*byStart*

开始码(串口命令)

*byCMD*

命令号: 01- 进入指令, 02- 离开指令, 03- 拥堵状态指令(为 03 时, 只有 *byLaneState* 和 *byQueueLen* 有效)

*byRes*

保留

*wDeviceID*

设备 ID

*wDataLen*

数据长度(串口信息)

*byTotalLaneNum*

有效车道总数

*byRes1*

保留

*struStartTime*

统计开始时间

*dwSamplePeriod*

统计时间, 单位: 秒

*struLaneParam*

每个车道 TPS 统计信息

## 7.151 NET\_DVR\_TRIGCOORDINATE:线圈位置配置参数

```
struct{
    WORD    wTopLeftX;
    WORD    wTopLeftY;
    WORD    wWdith;
    WORD    wHeight;
}NET_DVR_TRIGCOORDINATE, *LPNET_DVR_TRIGCOORDINATE;
```

### Members

*wTopLeftX*

线圈左上角横坐标 (2 个字节), 默认为 160

*wTopLeftY*

线圈左上角纵坐标 (2 个字节), 默认为 210

*wWdith*

线圈宽度 (2 个字节), 默认为 1000

*wHeight*

线圈高度 (2 个字节), 默认为 45

## 7.152 NET\_DVR\_TRIGGER\_COND:触发模式配置条件

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    DWORD    dwtriggerMode;
    BYTE     byRes[64];
}NET_DVR_TRIGGER_COND,*LPNET_DVR_TRIGGER_COND;
```

### Members

*dwSize*

结构体大小

*dwChannel*

通道号

*dwTriggerMode*

触发模式，按位表示，定义详见 [ITC\\_TRIGGERMODE\\_TYPE](#)

*byRes*

保留，置为 0

**7.153 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—远程手动录像

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

## 7.154 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

## 7.155 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

## 7.156 NET\_DVR\_VEHICLE\_CONTROL:车辆信息管控参数

```
struct{
    BYTE    byGateOperateType;
    BYTE    byRes1;
    WORD    wAlarmOperateType;
    BYTE    byRes[8];
}NET_DVR_VEHICLE_CONTROL,*LPNET_DVR_VEHICLE_CONTROL;
```

### Members

*byGateOperateType*

操作类型: 0- 无操作, 1- 开道闸

*byRes1*

保留, 置为 0

*wAlarmOperateType*

报警处理类型: 0- 无操作, bit0- 继电器输出报警, bit1- 布防上传报警, bit3- 告警主机上传, 值: 0-表示关, 1-表示开, 可复选

*byRes*

保留，置为 0

## 7.157 NET\_DVR\_VEHICLE\_CONTROL\_ALARM: 车辆报警（允许名单）报警信息

```
struct{
    DWORD                dwSize;
    BYTE                 byListType;
    BYTE                 byPlateType;
    BYTE                 byPlateColor;
    BYTE                 byRes1;
    char                 sLicense[MAX_LICENSE_LEN];
    char                 sCardNo[MAX_CARDNO_LEN];
    NET_DVR_TIME_V30    struAlarmTime;
    BYTE                 byRes2[64];
}NET_DVR_VEHICLE_CONTROL_ALARM,*LPNET_DVR_VEHICLE_CONTROL_ALARM;
```

### Members

*dwSize*

结构体大小

*byListType*

名单属性（允许名单）：0- 允许名单，1- 禁止名单，2- 临时名单

*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

#### *byPlateColor*

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_GREEN_PLATE,
    VCA_BKAIR_PLATE,
    VCA_OTHER = 0xff
}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

*VCA\_GREEN\_PLATE*

绿色车牌

*VCA\_BKAIR\_PLATE*

民航黑色车牌

*VCA\_OTHER = 0xff*

其他

#### *byRes1*

保留，置为 0

#### *sLicense*

车牌号码

#### *sCardNo*

卡号

#### *struAlarmTime*

报警时间

#### *byRes2*

保留，置为 0

## 7.158 NET\_DVR\_VEHICLE\_CONTROL\_COND: 车辆允许名单信息获取条件

```

struct{
    DWORD    dwChannel;
    DWORD    dwOperateType;
    char     sLicense[MAX_LICENSE_LEN];
    char     sCardNo[MAX_CARDNO_LEN];
    BYTE     byListType;
    BYTE     byRes1[3];
    DWORD    dwDataIndex;
    BYTE     byRes[116];
}NET_DVR_VEHICLE_CONTROL_COND,*LPNET_DVR_VEHICLE_CONTROL_COND;

```

### Members

#### *dwChannel*

通道号，0xffffffff 表示全部通道（ITC 默认是 1）

#### *dwOperateType*

操作类型（可复选），具体类型如下：

```

enum _VCA_OPERATE_TYPE_{
    VCA_LICENSE_TYPE          = 0x1,
    VCA_PLATECOLOR_TYPE       = 0x2,
    VCA_CARDNO_TYPE           = 0x4,
    VCA_PLATETYPE_TYPE        = 0x8,
    VCA_LISTTYPE_TYPE         = 0x10,
    VCA_INDEX_TYPE            = 0x20,
    VCA_OPERATE_INDEX_TYPE    = 0x40
}VCA_OPERATE_TYPE

```

*VCA\_LICENSE\_TYPE*

车牌号码

*VCA\_PLATECOLOR\_TYPE*

车牌颜色

*VCA\_CARDNO\_TYPE*

卡号

*VCA\_PLATETYPE\_TYPE*

车牌类型

*VCA\_LISTTYPE\_TYPE*

车辆名单类型

*VCA\_INDEX\_TYPE*

数据流水号

*VCA\_OPERATE\_INDEX\_TYPE*

操作数

#### *sLicense*

车牌号码

*sCardNo*

卡号

*byListType*

名单属性（允许名单）：0- 允许名单，1- 禁止名单，0xff- 全部

*byRes1*

保留，置为 0

*dwDataIndex*

数据流水号

*byRes*

保留，置为 0

#### Remarks

同步操作数用来标记平台更新到哪个操作，为同步记录表的流水号，主要是平台进行增量数据同步时使用。数据流水号是具体描述：操作的是平台数据表哪一条记录，因为平台中允许名单是分开两个表，前端相机需要根据数据流水号+类型来唯一定位一条数据记录。同步操作数和数据流水号均由平台进行控制，通过这两个字段可以对外提供允许名单增量更新的功能。

## 7.159 NET\_DVR\_VEHICLE\_CONTROL\_DELINFO:禁止名单数据库删除信息

```
struct{
    DWORD    dwSize;
    DWORD    dwDelType;
    char     sLicense[MAX_LICENSE_LEN];
    char     sCardNo[MAX_CARDNO_LEN];
    BYTE     byPlateType;
    BYTE     byPlateColor;
    BYTE     byOperateType;
    BYTE     byListType;
    DWORD    dwDataIndex;
    char     sOperateIndex[MAX_OPERATE_INDEX_LEN];
    BYTE     byRes[24];
}NET_DVR_VEHICLE_CONTROL_DELINFO,*LPNET_DVR_VEHICLE_CONTROL_DELINFO;
```

#### Members

*dwSize*

结构体大小

*dwDelType*

删除条件类型（可复选，*byOperateType* 为 0 时有效），具体定义如下：

```
enum _VCA_OPERATE_TYPE_{
    VCA_LICENSE_TYPE      = 0x1,
    VCA_PLATECOLOR_TYPE  = 0x2,
    VCA_CARDNO_TYPE      = 0x4,
    VCA_PLATETYPE_TYPE   = 0x8,
```

```
VCA_LISTTYPE_TYPE      = 0x10,
VCA_INDEX_TYPE         = 0x20,
VCA_OPERATE_INDEX_TYPE = 0x40
}VCA_OPERATE_TYPE
```

*VCA\_LICENSE\_TYPE*

车牌号码

*VCA\_PLATECOLOR\_TYPE*

车牌颜色

*VCA\_CARDNO\_TYPE*

卡号

*VCA\_PLATETYPE\_TYPE*

车牌类型

*VCA\_LISTTYPE\_TYPE*

车辆名单类型

*VCA\_INDEX\_TYPE*

数据流水号

*VCA\_OPERATE\_INDEX\_TYPE*

操作数

*sLicense*

车牌号码

*sCardNo*

卡号

*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

#### *byPlateColor*

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_GREEN_PLATE,
    VCA_BKAIR_PLATE,
    VCA_OTHER = 0xff
}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

*VCA\_GREEN\_PLATE*

绿色车牌

*VCA\_BKAIR\_PLATE*

民航黑色车牌

*VCA\_OTHER = 0xff*

其他

#### *byOperateType*

删除操作类型：0- 条件删除，0xff- 删除全部

#### *byListType*

名单属性（允许名单）：0- 允许名单，1- 禁止名单

#### *dwDataIndex*

数据流水号（平台禁止名单表和允许名单表的流水号，在这里有可能会重复，要与 *byListType* 配合才能唯一确定一条记录）

#### *sOperateIndex*

操作数

#### *byRes*

保留，置为 0

## 7.160 NET\_DVR\_VEHICLE\_CONTROL\_LIST\_DSALARM:允许名单数据需要同步报警信息

```
struct{
    DWORD    dwSize;
    DWORD    dwDataIndex;
    char     sOperateIndex[MAX_OPERATE_INDEX_LEN];
    BYTE     byRes[32];
}NET_DVR_VEHICLE_CONTROL_LIST_DSALARM,*LPNET_DVR_VEHICLE_CONTROL_LIST_DSALARM;
```

### Members

#### *dwSize*

结构体大小

#### *dwDataIndex*

数据流水号（这个值返回设备当前接收到的数据的最后一个数据流水号，平台同步的时候从下一个流水号开始同步）

#### *sOperateIndex*

操作数（平台同步表流水号不会重复，用于增量更新，代表同步到同步表的某一条记录了，存在相机内存，重启后会清 0）

#### *byRes*

保留，置为 0

## 7.161 NET\_DVR\_VEHICLE\_CONTROL\_LIST\_INFO:出入口允许名单的数据同步信息

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    DWORD    dwDataIndex;
    char     sLicense[MAX_LICENSE_LEN];
    BYTE     byListType;
    BYTE     byPlateType;
    BYTE     byPlateColor;
    BYTE     byRes[21];
    char     sCardNo[MAX_CARDNO_LEN];
    NET\_DVR\_TIME\_V30 struStartTime;
    NET\_DVR\_TIME\_V30 struStopTime;
    char     sOperateIndex[MAX_OPERATE_INDEX_LEN];
    BYTE     byRes1[224];
}NET_DVR_VEHICLE_CONTROL_LIST_INFO,*LPNET_DVR_VEHICLE_CONTROL_LIST_INFO;
```

### Members

#### *dwSize*



结构体大小

#### *dwChannel*

通道号，0xffffffff 表示全部通道（ITC 默认是 1）

#### *dwDataIndex*

数据流水号（平台维护的数据唯一值，客户端操作的时候，该值不会起效。该值主要用于数据增量同步）

#### *sLicense*

车牌号码

#### *byListType*

名单属性（允许名单）：0- 允许名单，1- 禁止名单

#### *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

#### *byPlateColor*

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_GREEN_PLATE,
VCA_BKAIR_PLATE,
VCA_OTHER = 0xff
}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
VCA_GREEN_PLATE
绿色车牌
VCA_BKAIR_PLATE
民航黑色车牌
VCA_OTHER = 0xff
其他

```

*byRes*

保留，置为 0

*sCardNo*

卡号

*struStartTime*

有效开始时间

*struStopTime*

有效结束时间

*sOperateIndex*

同步操作数（平台同步表流水号不会重复，用于增量更新，代表同步到同步表的某一条记录了，存在相机内存，重启后会清 0）

*byRes2*

保留，置为 0

### Remarks

同步操作数用来标记平台更新到哪个操作，为同步记录表的流水号，主要是平台进行增量数据同步时使用。数据流水号是具体描述：操作的是平台数据表哪一条记录，因为平台中允许名单是分开两个表，前端相机需要根据数据流水号+类型来唯一定位一条数据记录。同步操作数和数据流水号均由平台进行控制，通过这两个字段可以对外提供允许名单增量更新的功能。

## 7.162 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，0xff-未进行车身颜色识别

#### *byRes1*

保留

#### *wSpeed*

车辆速度，单位 km/h

#### *wLength*

车身长度

#### *byIllegalType*

违法类型：0-正常：1-低速，2-超速，3-逆行，4-闯红灯，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，15-违章停车，16-违章掉头，17-占用应急车道，18-禁右，19-禁左，20-压黄线，21-未系安全带。V4.0.0 及以上版本的智能交通摄像机的违法类型也支持通过

[NET ITS PLATE RESULT](#) 中的 *wIllegalType* 进行判断

#### *byVehicleLogoRecog*

汽车品牌，定义详见 [VLR\\_VEHICLE\\_CLASS](#)

#### *byVehicleSubLogoRecog*

车辆子品牌，根据不同的主类型，子品牌取值定义不同

#### *byVehicleModel*

车辆子品牌年款，根据不同的主类型，子品牌年款取值定义不同

#### *byCustomInfo*

自定义信息

*wVehicleLogoRecog*

车辆主品牌(该字段兼容 *byVehicleLogoRecog*), 定义详见 [VLR\\_VEHICLE\\_CLASS](#)

*byRes3*

保留

**Remarks**

车辆主品牌对应的子品牌、子品牌年款取值, 请参考相应设备提供的品牌对应关系列表

**7.163 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]

**7.164 NET\_DVR\_VIDEOOUT\_V30:视频输出参数**

```
struct{
    DWORD                dwSize;
    NET\_DVR\_VOOUT        struVOOut[MAX_VIDEOOUT_V30];
}
```

```

NET_DVR_VGAPARA          struVGAPara[MAX_VGA_V30];
NET_DVR_MATRIXPARA_V30  struMatrixPara[MAX_MATRIXOUT];
BYTE                     byRes[16];
}NET_DVR_VIDEOOUT_V30,*LPNET_DVR_VIDEOOUT_V30;

```

### Members

*dwSize*

结构体大小

*struVOOut*

模拟视频 BNC 输出口参数

*struVGAPara*

VGA 参数，智能交通摄像机不支持，保留

*struMatrixPara*

MATRIX 参数，智能交通摄像机不支持，保留

*byRes*

保留，置为 0

## 7.165 NET\_DVR\_VOOUT:模拟视频输出参数

```

struct{
    BYTE      byVideoFormat;
    BYTE      byMenuAlphaValue;
    WORD      wScreenSaveTime;
    WORD      wVOffset;
    WORD      wBrightness;
    BYTE      byStartMode;
    BYTE      byEnableScaler;
}NET_DVR_VOOUT,*LPNET_DVR_VOOUT;

```

### Members

*byVideoFormat*

输出制式：0-PAL，1-NTSC

*byMenuAlphaValue*

菜单与背景图象对比度

*wScreenSaveTime*

屏幕保护时间：0-从不，1-1 分钟，2-2 分钟，3-5 分钟，4-10 分钟，5-20 分钟，6-30 分钟

*wVOffset*

视频输出偏移

*wBrightness*

视频输出亮度

*byStartMode*

启动后视频输出模式：0-菜单，1-预览

*byEnableScaler*

是否启动缩放：0-不启动，1-启动

### Remarks

该结构体中各参数配置需要设备支持，例如：智能交通摄像机不支持配置是否启动缩放，默认启动缩

放成 D1(720\*576)输出。

## 7.166 NET\_DVR\_VTPARAM: 虚拟线圈参数

```

struct{
    DWORD                dwSize;
    BYTE                 byEnable;
    BYTE                 byIsDisplay;
    BYTE                 byLoopPos;
    BYTE                 bySnapGain;
    BYTE                 dwSnapShutter;
    NET_DVR_TRIGCOORDINATE struTrigCoordinate;
    NET_DVR_TRIGCOORDINATE struRes[MAX_VL_NUM];
    BYTE                 byTotalLaneNum;
    BYTE                 byPolarLenType;
    BYTE                 byDayAuxLightMode;
    BYTE                 byLoopToCalRoadBright;
    BYTE                 byRoadGrayLowTh;
    BYTE                 byRoadGrayHighTh;
    WORD                 wLoopPosBias;
    DWORD                dwHfrShtterInitValue;
    DWORD                dwSnapShtterInitValue;
    DWORD                dwHfrShtterMaxValue;
    DWORD                dwSnapShtterMaxValue;
    DWORD                dwHfrShtterNightValue;
    DWORD                dwSnapShtterNightMinValue;
    DWORD                dwSnapShtterNightMaxValue;
    DWORD                dwInitAfe;
    DWORD                dwMaxAfe;
    WORD                 wResolutionX;
    WORD                 wResolutionY;
    WORD                 dwGainNightValue;
    WORD                 dwSceneMode;
    WORD                 dwRecordMode;
    NET_DVR_GEOGLOCATION struGeogLocation;
    BYTE                 byTrigFlag[MAX_VL_NUM];
    BYTE                 byTrigSensitive[MAX_VL_NUM];
    BYTE                 byRes2[62];
}NET_DVR_VTPARAM, *LPNET_DVR_VTPARAM;

```

### Members

*dwSize*

结构体大小

*byEnable*

是否使能虚拟线圈，0-不使用，1-使用

***byIsDisplay***

是否显示虚拟线圈，0-不显示，1-显示

***byLoopPos***

晚间触发线圈的偏向：0-正向，1-负向（保留）

***bySnapGain***

抓拍增益

***dwSnapShutter***

抓拍快门速度

***struTrigCoordinate***

虚拟线圈参数

***struRes***

保留

***byTotalLaneNum***

视频触发的车道数

***byPolarLenType***

偏振镜类型，0：不加偏振镜；1：加施耐德偏振镜

***byDayAuxLightMode***

白天辅助照明模式，0：无辅助照明；1：LED 灯照明；2：闪光灯照明

***byLoopToCalRoadBright***

用以计算路面亮度的车道(虚拟线圈)

***byRoadGrayLowTh***

路面亮度低阈值初始化值，默认值为 1

***byRoadGrayHighTh***

路面亮度高阈值初始化值，默认值为 140

***wLoopPosBias***

晚间触发线圈位移，默认值为 30

***dwHfrShtterInitValue***

连续图像曝光时间的初始值，单位 us，默认值为 2000

***dwSnapShtterInitValue***

抓拍图像曝光时间的初始值，默认值为 500

***dwHfrShtterMaxValue***

连续图像曝光时间的最大值，默认值为 20000

***dwSnapShtterMaxValue***

抓拍图像曝光时间的最大值，默认值为 1500

***dwHfrShtterNightValue***

晚间连续图像曝光时间的设置值，默认值为 3000

***dwSnapShtterNightMinValue***

晚间抓拍图像曝光时间的最小值，默认值为 3000

***dwSnapShtterNightMaxValue***

晚间抓拍图像曝光时间的最大值，默认值为 5000

***dwInitAfe***

增益的初始值，默认值为 200

***dwMaxAfe***

增益的最大值，默认值为 400

***wResolutionX***

设备当前分辨率宽

***wResolutionY***

设备当前分辨率高

***dwGainNightValue***

晚间增益，默认值 70

***dwSceneMode***

场景模式

```
enum _SCENE_MODE_{
    UNKOWN_SCENE_MODE = 0,
    HIGHWAY_SCENE_MODE = 1,
    SUBURBAN_SCENE_MODE = 2,
    URBAN_SCENE_MODE = 3,
    TUNNEL_SCENE_MODE = 4
}SCENE_MODE
```

***UNKOWN\_SCENE\_MODE***

未知场景模式

***HIGHWAY\_SCENE\_MODE***

高速场景模式

***SUBURBAN\_SCENE\_MODE***

郊区场景模式(保留)

***URBAN\_SCENE\_MODE***

市区场景模式

***TUNNEL\_SCENE\_MODE***

隧道场景模式(保留)

***dwRecordMode***

录像标志：0-不录像，1-录像

***struGeogLocation***

地址位置

***byTrigFlag***

触发标志，0-车头触发；1-车尾触发；2-车头/车尾都触发

***byTrigSensitive***

触发灵敏度，1-100

***byRes2***

保留

**7.167 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*

0-F

*byWDRLevel2*

0-F

*byWDRContrastLevel*

0-100

*byRes*

保留

## 7.168 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

## 7.169 NET\_DVR\_ZONEANDDST:夏令时参数结构体

```
struct{
    DWORD    dwSize;
    BYTE    byRes1[16];
    DWORD    dwEnableDST;
    BYTE    byDSTBias;
    BYTE    byRes2[3];
    NET_DVR_TIMEPOINT    struBeginPoint;
    NET_DVR_TIMEPOINT    struEndPoint;
}NET_DVR_ZONEANDDST,*LPNET_DVR_ZONEANDDST;
```

**Members****dwSize**

结构体大小

**byRes1**

保留，置为 0

**dwEnableDST**

是否启用夏时制：0—不启用，1—启用

**byDSTBias**

夏令时偏移值（以分钟计）：30min，60min，90min，120min

**byRes2**

保留，置为 0

**struBeginPoint**

夏时制开始时间

**struEndPoint**

夏时制停止时间

**7.170 NET\_IPC\_LANE\_HVT\_PARAM:IPC 混行卡口车道参数**

```

struct{
    BYTE                byLaneNO;
    BYTE                byCarDriveDirect;
    BYTE                byRes[62];
    NET\_ITC\_LINE       struLaneLine;
    NET\_ITC\_POLYGON    struPlateRecog;
    BYTE                byRes1[256];
}NET_IPC_LANE_HVT_PARAM, *LPNET_IPC_LANE_HVT_PARAM;

```

**Members****byLaneNO**

车道号

**byCarDriveDirect**

车辆行驶方向，具体定义如下：

```

enum_ITC_LANE_CAR_DRIVE_DIRECT_{
    ITC_LANE_DRIVE_UNKNOW      = 0,
    ITC_LANE_DRIVE_UP_TO_DOWN  = 1,
    ITC_LANE_DRIVE_DOWN_TO_UP  = 2
}ITC_LANE_CAR_DRIVE_DIRECT

```

**ITC\_LANE\_DRIVE\_UNKNOW**

未知

**ITC\_LANE\_DRIVE\_UP\_TO\_DOWN**

从上往下行驶(下行)

**ITC\_LANE\_DRIVE\_DOWN\_TO\_UP**

从下往上行驶(上行)

**byRes**

保留，置为 0

*struLaneLine*  
车道线

*struPlateRecog*  
牌识区域

*byRes1*  
保留，置为 0

## 7.171 NET\_IPC\_POST\_HVT\_PARAM:IPC 混行卡口触发参数

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
    BYTE                byRes[62];
    NET_ITC_LINE        struLaneBoundaryLine;
    NET_ITC_PLATE_RECOG_PARAM struPlateRecog;
    NET_IPC_LANE_HVT_PARAM struLaneParam[MAX_ITC_LANE_NUM];
    char                szSceneName[NAME_LEN];
    BYTE                byRes1[408];
}NET_IPC_POST_HVT_PARAM,*LPNET_IPC_POST_HVT_PARAM;
```

### Members

*byEnable*  
是否启用：0- 不启用，1- 启用

*byLaneNum*  
识别的车道个数

*byRes*  
保留

*struLaneBoundaryLine*  
车道边界线（最左边车道的左边界线）

*struPlateRecog*  
车牌识别参数

*struLaneParam*  
车道参数

*szSceneName*  
场景名称

*byRes1*  
保留

## 7.172 NET\_ITC\_ACCESS\_DEVINFO\_PARAM\_UNION:RS485 关联接入设备配置

```
struct{
    BYTE                uLen[128];
```

```

NET\_ITC\_RADAR\_INFO\_PARAM    struRadarInfoParam;
}NET_ITC_ACCESS_DEVINFO_PARAM_UNION, *LPNET_ITC_ACCESS_DEVINFO_PARAM_UNION;

```

### Members

*uLen*

联合体大小，128 字节

*struRadarInfoParam*

雷达参数配置

## 7.173 **NET\_ITC\_EPOLICE\_IOTL\_PARAM**:电警 IO 红绿灯参数

```

struct{
    NET\_ITC\_PLATE\_RECOG\_PARAM    struPlateRecog;
    NET\_ITC\_SINGLE\_IOTL\_PARAM    struSingleIOTL[MAX_IOSPEED_GROUP_NUM];
    BYTE                          byRes[32];
}NET_ITC_EPOLICE_IOTL_PARAM,*LPNET_ITC_EPOLICE_IOTL_PARAM;

```

### Members

*struPlateRecog*

牌识参数

*struSingleIOTL*

单组 IO 红绿灯参数

*byRes*

保留

## 7.174 **NET\_ITC\_EPOLICE\_LANE\_PARAM**:电警 RS485 车检器触发模式车道参数

```

struct{
    BYTE                          byEnable;
    BYTE                          byRelatedDriveWay;
    WORD                          wDistance;
    BYTE                          byRecordEnable;
    BYTE                          byRecordType;
    BYTE                          byPreRecordTime;
    BYTE                          byRecordDelayTime;
    BYTE                          byRecordTimeOut;
    BYTE                          bySignSpeed;
    BYTE                          bySpeedLimit;
    BYTE                          byOverlayDriveWay;
    NET\_ITC\_SERIAL\_INFO          struSerialInfo;
    BYTE                          byRelatedIOOut[MAX_IOOUT_NUM];
    BYTE                          byFlashMode;
    BYTE                          bySerialType;

```

```

BYTE byRelatedIOOutEx;
BYTE bySnapPicPreRecord;
NET_ITC_PLATE_RECOG_REGION_PARAM struPlateRecog[MAX_LANEAREA_NUM];
BYTE byBigCarSignSpeed;
BYTE byBigCarSpeedLimit;
BYTE byRedTrafficLightChan;
BYTE byYellowTrafficLightChan;
BYTE byRelaLaneDirectionType;
BYTE byRes3[11];
}NET_ITC_EPOLICE_LANE_PARAM,*LPNET_ITC_EPOLICE_LANE_PARAM;

```

### Members

#### *byEnable*

是否启用：0- 不启用，1- 启用

#### *byRelatedDriveWay*

关联的车道号

#### *wDistance*

线圈距离（保留），单位：厘米

#### *byRecordEnable*

闯红灯周期录像标志：0- 不录像，1- 录像

#### *byRecordType*

闯红灯录像类型：0- 预录，1- 延时录像

#### *byPreRecordTime*

闯红灯录像片段预录时间（默认 0），单位：秒

#### *byRecordDelayTime*

闯红灯录像片段延时时间（默认 0），单位：秒

#### *byRecordTimeOut*

闯红灯周期录像超时时间，单位：秒

#### *bySignSpeed*

标志限速(卡式电警模式有效)，单位：km/h

#### *bySpeedLimit*

限速值(卡式电警模式有效)，单位：km/h

#### *byOverlayDriveWay*

OSD 叠加的车道号

#### *struSerialInfo*

车检器参数

#### *byRelatedIOOut*

关联的 IO 输出口，可以同时关联多个

#### *byFlashMode*

闪光灯闪烁模式：0- 同时闪，1- 轮流闪

#### *bySerialType*

车检器类型：0-私有车检器，1-私有 OEM 车检器，2-其他车检器

#### *byRelatedIOOutEx*

关联的 IO 输出口，按位表示输出口号：bit0 表示 IO 输出口 1，bit1 表示 IO 输出口 2，以此类推。  
值：0-不关联，1-关联，最多支持关联到 8 个 IO 输出口(兼容 byRelatedIOOut 字段)

**bySnapPicPreRecord**

抓拍图片预录时间点：0- 默认值（第二张图片），1- 第一张图片，2- 第二张图片，3- 第三张图片

**struPlateRecog**

牌识区域参数

**byBigCarSignSpeed**

大车标志限速，单位 km/h

**byBigCarSpeedLimit**

大车限速值，单位 km/h

**byRedTrafficLightChan**

红灯通道号：1~16（红绿灯检测器）

**byYellowTrafficLightChan**

黄灯通道号：1~16（红绿灯检测器）

**byRelaLaneDirectionType**

关联车道方向类型，定义详见 [ITC\\_RELA\\_LANE\\_DIRECTION\\_TYPE](#)

**byRes3**

保留

**Remarks**

该结构体中的参数 **byRelatedDriveWay** 为关联车道号，是和车检器中的车道关联匹配，实现抓拍。  
**byOverlayDriveWay** 为叠加车道号，为实际车道号。

当 **bySerialType** 为 0 时，**struSerialInfo** 参数有效。

**7.175 NET\_ITC\_EPOLICE\_RS485\_PARAM:电警/卡式电警 RS485 车检器****触发参数**

```
struct{
    BYTE                byRelatedLaneNum;
    BYTE                byTrafficLightSignalSrc;
    BYTE                byRes1[2];
    NET\_ITC\_PLATE\_RECOG\_PARAM    struPlateRecog;
    NET\_ITC\_EPOLICE\_LANE\_PARAM    struLane[MAX_ITC_LANE_NUM];
    BYTE                byRes[32];
}NET_ITC_EPOLICE_RS485_PARAM,*LPNET_ITC_EPOLICE_RS485_PARAM;
```

**Members****byRelatedLaneNum**

关联的车道个数

**byTrafficLightSignalSrc**

交通灯信号来源：0- 车检器，1- 红绿灯检测器

**byRes1**

保留

**struPlateRecog**

牌识参数

**struLane**

关联的车道参数

*byRes*

保留

### Remarks

在交通灯信号来源为红绿灯检测器时，NET\_ITC\_EPOLICE\_LANE\_PARAM 中关联红灯通道号和关联黄灯通道号有效；在交通灯信号来源为车检器时，关联红灯通道号和关联黄灯通道号无效。

## 7.176 NET\_ITC\_EXCEPTION:智能交通摄像机异常参数

```
struct{
    DWORD                dwSize;
    NET_ITC_HANDLEEXCEPTION struSnapExceptionType[MAX_EXCEPTIONNUM_V30];
}NET_ITC_EXCEPTION,*LPNET_ITC_EXCEPTION;
```

### Members

*dwSize*

结构体大小

*struSnapExceptionType*

异常信息处理方式，每位数组表示一种异常：

数组 0—硬盘出错

数组 1—网线断

数组 2—IP 地址冲突

数组 3—车检器异常

数组 4—信号灯检测器异常

## 7.177 NET\_ITC\_FLASHOUT\_INFO:同步输出参数

```
struct{
    BYTE    byFlashOutIndex[MAX_IOOUT_NUM];
    BYTE    byRes[44];
}NET_ITC_FLASHOUT_INFO,*LPNET_ITC_FLASHOUT_INFO;
```

### Members

*byFlashOutIndex*

同步输出号

*byRes*

保留

## 7.178 NET\_ITC\_FTP\_CFG:智能交通摄像机 FTP 配置

```
struct{
    DWORD                dwSize;
    BYTE                byEnable;
    BYTE                byAddressType;
    WORD                wFTPPort;
```

```

union{
    struct{
        BYTE          szDomain[MAX_DOMAIN_NAME];
        BYTE          byRes1[80];
    }struDomain;
    struct{
        NET\_DVR\_IPADDR  strulp;
    }struAddrIP;
}unionServer;
BYTE          szUserName[NAME_LEN];
BYTE          szPassWORD[PASSWD_LEN];
BYTE          byRes4;
BYTE          byDirLevel;
BYTE          byIsFilterCarPic;
BYTE          byUploadDataType;
NET\_DVR\_PICTURE\_NAME  struPicNameRule;
BYTE          byTopDirMode;
BYTE          bySubDirMode;
BYTE          byThreeDirMode;
BYTE          byFourDirMode;
BYTE          szPicNameCustom[MAX_CUSTOMDIR_LEN];
BYTE          szTopCustomDir[MAX_CUSTOMDIR_LEN];
BYTE          szSubCustomDir[MAX_CUSTOMDIR_LEN];
BYTE          szThreeCustomDir[MAX_CUSTOMDIR_LEN];
BYTE          szFourCustomDir[MAX_CUSTOMDIR_LEN];
BYTE          byRes3[900];
}NET_ITC_FTP_CFG,*LPNET_ITC_FTP_CFG;

```

### Members

#### *dwSize*

结构体大小

#### *byEnableFTP*

是否启动 ftp 上传功能: 0- 否, 1- 是

#### *byAddressType*

地址类型: 0- 实际 ipv4/ipv6 地址, 1- 域名

#### *wFTPPort*

FTP 端口号

#### **unionServer 为 FTP 地址联合体**

##### *szDomain*

服务器地址, 域名

##### *byRes1*

保留, 置为 0

##### *strulp*

IPv4 或者 IPv6 地址, 144 字节

##### *szUserName*



用户名

*szPassWORD*

密码

*byRes4*

保留，置为 0

*byDirLevel*

目录等级：0- 不使用目录结构，直接保存在根目录；1- 使用 1 级目录；2- 使用 2 级目录；3- 使用 3 级目录；4-使用 4 级目录

*byIsFilterCarPic*

车牌小图是否上传：0- 上传，1- 不上传

*byUploadDataType*

上传数据类型：0- 全部，1- 卡口，2- 违章 (单 FTP 时默认选择全部，双 FTP 时支持卡口、违章的选择)

*struPicNameRule*

图片命名规则

*byTopDirMode*

一级目录：0x1- 使用设备名，0x2- 使用设备号，0x3- 使用设备 ip 地址，0x4- 使用监测点，0x5- 使用时间(年月)，0x6- 使用时间(年月日)，0x7- 违规类型，0x8- 方向，0x9- 地点，0xa- 通道名，0xb- 通道号，0xc-车道号，0xff- 自定义

*bySubDirMode*

二级目录：0x1- 使用设备名，0x2- 使用设备号，0x3- 使用设备 ip 地址，0x4- 使用监测点，0x5- 使用时间(年月)，0x6- 使用时间(年月日)，0x7- 违规类型，0x8- 方向，0x9- 地点，0xa- 通道名，0xb- 通道号，0xc-车道号，0xff- 自定义

*byThreeDirMode*

三级目录：0x1- 使用设备名，0x2- 使用设备号，0x3- 使用设备 ip 地址，0x4- 使用监测点，0x5- 使用时间(年月)，0x6- 使用时间(年月日)，0x7- 违规类型，0x8- 方向，0x9- 地点，0xa- 通道名，0xb- 通道号，0xc-车道号，0xff- 自定义

*byFourDirMode*

四级目录：0x1- 使用设备名，0x2- 使用设备号，0x3- 使用设备 ip 地址，0x4- 使用监测点，0x5- 使用时间(年月)，0x6- 使用时间(年月日)，0x7- 违规类型，0x8- 方向，0x9- 地点，0xa- 通道名，0xb- 通道号，0xc-车道号，0xff- 自定义

*szPicNameCustom*

图片命名自定义，当文件命名规则选择 PICNAME\_ITEM\_CUSTOM 时生效

*szTopCustomDir*

自定义一级目录

*szSubCustomDir*

自定义二级目录

*szThreeCustomDir*

自定义三级目录

*szFourCustomDir*

自定义四级目录

*byRes3*

保留，置为 0

**See Also**

设备是否支持该 FTP 配置通过能力集 [NET\\_DVR\\_SNAP\\_ABILITY](#) 中的 bySupport&0x40 进行判断。

## 7.179 NET\_ITC\_FTP\_TYPE\_COND:FTP 参数配置条件结构体

```
struct{
    DWORD   dwChannel;
    BYTE    byWorkMode;
    BYTE    byRes[7];
}NET_ITC_FTP_TYPE_COND,*LPNET_ITC_FTP_TYPE_COND;
```

### Members

#### *dwChannel*

设备通道号

#### *byWorkMode*

FTP 模式：0- FTP1(主 FTP)，1- FTP2(备 FTP)

#### *byRes*

保留

## 7.180 NET\_ITC\_HANDLEEXCEPTION:智能交通摄像机报警和异常处理参数

```
struct{
    DWORD   dwHandleType;
    BYTE    byEnable;
    BYTE    byRes;
    WORD    wDuration;
    BYTE    byAlarmOutTriggered[MAX_ITC_EXCEPTIONOUT];
    BYTE    byRes1[8];
}NET_ITC_HANDLEEXCEPTION, *LPNET_ITC_HANDLEEXCEPTION;
```

### Members

#### *dwHandleType*

处理方式，异常处理方式的"或"结果：

0x04: 上传中心

0x08: 触发报警输出（继电器输出）

#### *byEnable*

是否启用异常处理：0- 不启用，1- 启用

#### *byRes*

保留

#### *wDuration*

持续时间，单位：s

#### *byRelAlarmOut*

报警触发的输出通道，数组下标表示通道号，0-不触发，1-触发输出

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，依次

类推

*byRes1*

保留

#### Remarks

触发报警输出功能，目前只有型号带“-K”的智能交通摄像机（例如，DS-2CD9131-KS）支持（支持一个继电器），非-K 设备不支持。

## 7.181 NET\_ITC\_ICR\_ALGAOTOSWITCH\_PARAM:ICR 算法自动参数

```
struct{
    BYTE    byDetectThreshold;
    BYTE    byAbBrightnessThreshold;
    BYTE    byRes[154];
}NET_ITC_ICR_ALGAOTOSWITCH_PARAM, *LPNET_ITC_ICR_ALGAOTOSWITCH_PARAM;
```

#### Members

*byDetectThreshold*

检测阈值（交通灯滤光片切换等级阈值设置，高于阈值打开滤光片），范围：[0,100]，默认：50

*byAbBrightnessThreshold*

异常亮度阈值（ICR 自动切换后前后亮度统计差值，若比该检测阈值大，则认为检测异常；若比该值小，则认为检测正常），范围：[0,100]，默认：12

*byRes*

保留，置为 0

## 7.182 NET\_ITC\_ICR\_AOTOSWITCH\_PARAM:ICR 自动切换参数

```
struct{
    BYTE    byICRPreset[MAX_ICR_NUM];
    BYTE    byICRAutoSwitch;
    BYTE    byRes[147];
}NET_ITC_ICR_AOTOSWITCH_PARAM, *LPNET_ITC_ICR_AOTOSWITCH_PARAM;
```

#### Members

*byICRPreset*

ICR 预置点，数组下标表示预置点号（即下标 0~7 对应 1~8，实际支持预置点个数通过能力集获取），取值范围：[0,100]，值越大 ICR 滤光面积越大

*byICRAutoSwitch*

ICR 自动切换加阈值选项，取值范围：[0,100]

*byRes*

保留，置为 0

#### Remarks

设备支持的 ICR 预置点（滤光片偏移点）个数通过能力集 [NET\\_DVR\\_SNAP\\_ABILITY](#) 获取，对应其中参数：  
*byICRPresetNum*

## 7.183 NET\_ITC\_ICR\_MANUALSWITCH\_PARAM:ICR 手动切换参数

```
struct{
    BYTE    byICRPreset[MAX_ICR_NUM];
    BYTE    bySubSwitchMode;
    BYTE    byRes[147];
}NET_ITC_ICR_MANUALSWITCH_PARAM, *LPNET_ITC_ICR_MANUALSWITCH_PARAM;
```

### Members

#### *byICRPreset*

ICR 预置点，设备支持的 ICR 预置点个数大于 0 时有效，数组下标表示预置点号（即下标 0~7 对应 1~8，实际支持预置点个数通过能力集获取），取值范围：[0,100]，值越大 ICR 滤光面积越大

#### *bySubSwitchMode*

1- 白天, 2- 晚上，设备支持的 ICR 预置点个数为 0 时有效

#### *byRes*

保留，置为 0

### See Also

设备支持的 ICR 预置点（滤光片偏移点）个数通过能力集 [NET\\_DVR\\_SNAP\\_ABILITY](#) 获取，对应其中参数：*byICRPresetNum*。

## 7.184 NET\_ITC\_ICR\_PARAM\_UNION:ICR 切换配置联合体

```
struct{
    BYTE    uLen[156];
    NET\_ITC\_ICR\_AOTOSWITCH\_PARAM    struICRAutoSwitch;
    NET\_ITC\_ICR\_MANUALSWITCH\_PARAM    struCRManualSwitch;
    NET\_ITC\_ICR\_TIMESWITCH\_PARAM    struCRTimeSwitch;
    NET\_ITC\_ICR\_ALGAOTOSWITCH\_PARAM    struICRAAlgorithmAutoSwitch;
}NET_ITC_ICR_PARAM_UNION, *LPNET_ITC_ICR_PARAM_UNION;
```

### Members

#### *uLen*

联合体大小，156 字节

#### *struICRAutoSwitch*

自动切换参数

#### *struCRManualSwitch*

手动切换参数

#### *struCRTimeSwitch*

定时切换参数

#### *struICRAAlgorithmAutoSwitch*

算法自动参数

## 7.185 NET\_ITC\_ICR\_TIMESWITCH\_PARAM:ICR 定时切换参数

```
struct{
    NET\_ITC\_ICRTIMECFG    struAutoCtrlTime[MAX_TIMESEGMENT_V30];
    BYTE                  byICRPreset[MAX_ICR_NUM];
    BYTE                  byRes[20];
}NET_ITC_ICR_TIMESWITCH_PARAM, *LPNET_ITC_ICR_TIMESWITCH_PARAM;
```

### Members

#### *struAutoCtrlTime*

自动切换时间段(现在支持 4 组, 其他 4 组预留)

#### *byICRPreset*

ICR 预置点, 设备支持的 ICR 预置点个数大于 0 时有效, 数组下标表示预置点号 (即下标 0~7 对应 1~8, 实际支持预置点个数通过能力集获取), 取值范围: [0,100], 值越大 ICR 滤光面积越大

#### *byRes*

保留, 置为 0

### See Also

设备支持的 ICR 预置点(滤光片偏移点)个数和支持的 ICR 的时间段数通过能力集 [NET\\_DVR\\_SNAP\\_ABILITY](#) 获取, 分别对应其中参数: `byICRPresetNum` 和 `byICRTimeSlot`。

## 7.186 NET\_ITC\_ICRCFG:ICR 切换配置参数

```
struct{
    DWORD                dwSize;
    BYTE                 bySwitchType;
    BYTE                 byRes[3];
    NET\_ITC\_ICR\_PARAM\_UNION    uICRParam;
}NET_ITC_ICRCFG, *LPNET_ITC_ICRCFG;
```

### Members

#### *dwSize*

结构体大小

#### *bySwitchType*

切换模式: 1- 自动切换, 2- 手动切换, 3- 定时切换

#### *byRes*

保留, 置为 0

#### *uICRParam*

ICR 切换配置联合体

## 7.187 NET\_ITC\_ICRTIMECFG:ICR 定时切换时间参数

```
struct{
    NET\_DVR\_SCHEDTIME    struTime;
    BYTE                 byAssociateResetNo;
```

```

    BYTE                bySubSwitchMode;
    BYTE                byRes[10];
}NET_ITC_ICRTIMECFG, *LPNET_ITC_ICRTIMECFG;

```

### Members

#### *struTime*

自动切换时间段，XX 时 XX 分至 XX 时 XX 分

#### *byAssociateRresetNo*

预置点号（设备支持的 ICR 预置点个数大于 0 时可设）：1~8，0 代表无

#### *bySubSwitchMode*

切换模式（设备支持的 ICR 预置点个数等于 0 时有效）：0- 无，1- 白天，2- 晚上

#### *byRes*

保留，置为 0

## 7.188 NET\_ITC\_INTERVAL\_PARAM:抓拍间隔参数

```

struct{
    BYTE    byIntervalType;
    BYTE    byRes1[3];
    WORD    wInterval[MAX_INTERVAL_NUM];
    BYTE    byRes[8];
}NET_ITC_INTERVAL_PARAM,*LPNET_ITC_INTERVAL_PARAM;

```

### Members

#### *byIntervalType*

间隔类型（默认按时间）：0- 时间起效，1- 距离起效

#### *byRes1*

保留

#### *wInterval*

连拍间隔时间（单位：ms）或连拍间隔距离（单位：分米）。当 *byIntervalType* 为 0 时，表示间隔时间，当 *byIntervalType* 为 1 时，表示距离

#### *byRes*

保留

## 7.189 NET\_ITC\_IO\_LIGHT\_PARAM:IO 接入信号灯参数

```

struct{
    NET\_ITC\_SINGLE\_IO\_LIGHT\_PARAM    struIOLight[MAX_LIGHT_NUM];
    BYTE                                byRes[8];
}NET_ITC_IO_LIGHT_PARAM,*LPNET_ITC_IO_LIGHT_PARAM;

```

### Members

#### *struIOLight*

单个 IO 接入信号灯参数

#### *byRes*

保留

## 7.190 NET\_ITC\_IOOUT\_PARAM:IO 输出配置参数

```

struct{
    DWORD    dwSize;
    BYTE     byDefaultStatus;
    BYTE     byIoOutStatus;
    BYTE     byMode;
    BYTE     byIOWorkMode;
    DWORD    dwTimeDelay;
    WORD     wAheadTime;
    BYTE     byFreqMulti;
    BYTE     byDutyRate;
    BYTE     byDetectBrightness;
    BYTE     byBrightnessThreld;
    BYTE     byFlashLightEnable;
    BYTE     byStartHour;
    BYTE     byStartMinute;
    BYTE     byEndHour;
    BYTE     byEndMinute;
    BYTE     byAutoPlateBrightness;
    BYTE     byRes[8];
}NET_ITC_IOOUT_PARAM, *LPNET_ITC_IOOUT_PARAM;

```

### Members

#### *dwSize*

大小

#### *byDefaultStatus*

IO 默认状态: 0- 低电平, 1- 高电平

#### *byIoOutStatus*

IO 起效时状态: 0- 低电平, 1- 高电平, 2- 脉冲

#### *byMode*

闪光灯工作方式, 按位表示: bit0-视频, bit1-卡口, bit2-违章, 值: 0-表示工作, 1-表示不工作。

#### *byIOWorkMode*

IO 输出口工作模式: 0- 闪光灯, 1- 偏振镜, 2- 常亮灯

#### *dwTimeDelay*

IO 有效持续时间, 单位: us

#### *wAheadTime*

输出 IO 提前时间, 单位: us

#### *byFreqMulti*

倍频, 数值范围: [1,15]

#### *byDutyRate*

占空比, 取值范围: [0,40%]

#### *byDetectBrightness*

自动检测亮度使能闪光灯/偏振镜/常亮灯: 0- 不检测, 1- 检测

**byBrightnessThreld**

使能闪光灯/偏振镜/常亮灯亮度阈值(byDetectBrightness 为 1 时有效), 范围: [0,100], 高于阈值闪

**byFlashLightEnable**

设置闪光灯/偏振镜/常亮灯时间使能: 0- 关, 1- 开

**byStartHour**

开始时间(byFlashLightEnable 为 1 时有效): 小时, 取值范围: [0,23]

**byStartMinute**

开始时间(byFlashLightEnable 为 1 时有效): 分, 取值范围: [0,59]

**byEndHour**

结束时间(byFlashLightEnable 为 1 时有效): 小时, 取值范围: [0,23]

**byEndMinute**

结束时间(byFlashLightEnable 为 1 时有效): 分, 取值范围: [0,59]

**byAutoPlateBrightness**

车牌亮度自动使能闪光灯: 0- 不启用, 1- 启用。启用了该功能后, 时控和光控都将不起作用。

**byRes**

保留, 置为 0

**Remarks**

各个模式下各个车道配置的关联 IO, IO 根据 byMode 参数处于何种工作状态, 例如: byMode 为 0 表示视频、卡口、违章情况下都工作。

## 7.191 NET\_ITC\_LANE\_HVT\_PARAM\_V50:混行卡口车道参数

```

struct{
    BYTE                byLaneNO;
    BYTE                byFlashMode;
    BYTE                bySignSpeed;
    BYTE                bySpeedLimit;
    BYTE                bySignLowSpeed;
    BYTE                byLowSpeedLimit;
    BYTE                byBigCarSignSpeed;
    BYTE                byBigCarSpeedLimit;
    BYTE                byBigCarSignLowSpeed;
    BYTE                byBigCarLowSpeedLimit;
    BYTE                bySnapTimes;
    BYTE                byDriveLineSnapTime;
    BYTE                byHighSpeedSnapTime;
    BYTE                byLowSpeedSnapTime;
    BYTE                byBanSnapTime;
    BYTE                byReverseSnapTime;
    BYTE                byRelatedDriveWay;
    BYTE                byLaneType;
    BYTE                byRelaLaneDirectionType;
    BYTE                byRes1[29];
    DWORD               dwVioDetectType;
}

```



```

DWORD                dwRelatedIOOut;
NET_ITC_LINE         struTrigLine;
NET_ITC_LINE         struLineLeft;
NET_ITC_POLYGON      struPlateRecog;
NET_ITC_LANE_LOGIC_PARAM struLane;
NET_ITC_INTERVAL_PARAM struInterval;
BYTE                 byRes2[280];
}NET_ITC_LANE_HVT_PARAM_V50,*LPNET_ITC_LANE_HVT_PARAM_V50;

```

### Members

#### *byLaneNO*

关联的车道号(用于叠加和上传)，取值范围：1~255

#### *byFlashMode*

闪光灯闪烁模式：0- 同时闪，1- 轮流闪

#### *bySignSpeed*

小车标志限高速，单位：km/h

#### *bySpeedLimit*

小车限高速值，单位：km/h

#### *bySignLowSpeed*

小车标志限低速，单位：km/h

#### *byLowSpeedLimit*

小车限低速值，单位：km/h

#### *byBigCarSignSpeed*

大车标志限高速，单位：km/h（新交规）

#### *byBigCarSpeedLimit*

大车限高速值，单位：km/h（新交规）

#### *byBigCarSignLowSpeed*

大车标志限低速，单位：km/h

#### *byBigCarLowSpeedLimit*

大车限低速值，单位：km/h

#### *bySnapTimes*

卡口抓拍张数，取值范围：1~3

#### *byDriveLineSnapTime*

压线抓拍张数，取值范围：1~3

#### *byHighSpeedSnapTime*

超高速抓拍张数，取值范围：1~3

#### *byLowSpeedSnapTime*

超低速抓拍张数，取值范围：1~3

#### *byBanSnapTime*

违反禁令抓拍张数，取值范围：1~3

#### *byReverseSnapTime*

逆行抓拍张数，取值范围：1~3

#### *byRelatedDriveWay*

关联车道号，用于匹配车检器

#### *byLaneType*

车道类型：0- 未配置，1- 高速公路，2- 城市快速路，0xff- 其他道路

*byRelaLaneDirectionType*

关联车道方向类型，类型详见 [ITC\\_RELA\\_LANE\\_DIRECTION\\_TYPE](#)。该参数为车道方向参数，与关联车道号 *byRelatedDriveWay* 对应，确保车道唯一性

*byRes1*

保留，置为 0

*dwVioDetectType*

违规检测类型，按位表示，取值：0-不启用，1-启用参考，类型详见 [ITC\\_VIOLATION\\_DETECT\\_TYPE](#)

*dwRelatedIOOut*

关联的 IO 输出口(可以同时关联多个)，按位表示 IO 输出口，第 0 位表示 IO 输出口 1，以此类推，取值：0-不关联，1-关联

*struTrigLine*

触发线，目前仅使用第一个车道

*struLineLeft*

左车道线

*struPlateRecog*

牌识区域

*struLane*

车道属性，其中 *byUseageType* 和 *byCarDriveDirect* 有效

*struInterval*

抓拍间隔参数

*byRes2*

保留，置为 0

## 7.192 NET\_ITC\_LANE\_IMT\_PARAM:智慧监控车道参数

```
struct{
    BYTE                byLaneNO;
    BYTE                byRelaLaneDirectionType;
    BYTE                byRes[146];
    NET\_ITC\_LANE      struLaneLine;
    BYTE                byRes1[256];
}NET_ITC_LANE_IMT_PARAM,*LPNET_ITC_LANE_IMT_PARAM;
```

### Members

*byLaneNO*

叠加车道号

*byRelaLaneDirectionType*

关联车道方向类型，类型详见 [ITC\\_RELA\\_LANE\\_DIRECTION\\_TYPE](#)。该参数为车道方向参数，与关联车道号 *byRelatedDriveWay* 对应，确保车道唯一性

*byRes*

保留，置为 0

*struLaneLine*

车道线

*byRes1*

保留，置为 0

### 7.193 NET\_ITC\_LANE\_LOGIC\_PARAM:车道属性参数

```
struct{
    BYTE    byUseageType;
    BYTE    byDirectionType;
    BYTE    byCarDriveDirect;
    BYTE    byRes[33];
}NET_ITC_LANE_LOGIC_PARAM, *LPNET_ITC_LANE_LOGIC_PARAM;
```

#### Members

##### *byUseageType*

车道用途类型，定义详见 [ITC\\_LANE\\_USEAGE\\_TYPE](#)

##### *byDirectionType*

车道方向类型，定义详见 [ITC\\_LANE\\_DIRECTION\\_TYPE](#)

##### *byCarDriveDirect*

车辆行驶方向，具体定义如下：

```
enum _ITC_LANE_CAR_DRIVE_DIRECT_{
    ITC_LANE_DRIVE_UNKNOW      = 0,
    ITC_LANE_DRIVE_UP_TO_DOWN  = 1,
    ITC_LANE_DRIVE_DOWN_TO_UP  = 2
}ITC_LANE_CAR_DRIVE_DIRECT
```

*ITC\_LANE\_DRIVE\_UNKNOW*

未知

*ITC\_LANE\_DRIVE\_UP\_TO\_DOWN*

从上往下行驶(下行)

*ITC\_LANE\_DRIVE\_DOWN\_TO\_UP*

从下往上行驶(上行)

##### *byRes*

保留，置为 0

### 7.194 NET\_ITC\_LANE\_MPR\_PARAM:多帧检测模式车道参数

```
struct{
    BYTE    byLaneNO;
    union{
        BYTE    uLen[4];
        struct{
            BYTE    byIONo;
            BYTE    byTriggerType;
            BYTE    byRes1[2];
        }struIO;
        struct{
```

```

        BYTE                byRelateChan;
        BYTE                byRes2[3];
    }struRS485;
}uTssParamInfo;
BYTE                byCarDriveDirect;
BYTE                byRes[58];
NET_ITC_LANE       struLaneLine;
NET_ITC_POLYGON    struPlateRecog;
BYTE                byRelaLaneDirectionType;
BYTE                byRes1[255];
}NET_ITC_LANE_MPR_PARAM,*LPNET_ITC_LANE_MPR_PARAM;

```

### Members

*byLaneNO*

车道号

***uTssParamInfo*** 为触发参数信息联合体

*uLen*

联合体大小，4 个字节

***struIO*** 为 IO 触发方式相关参数结构体

*byIONo*

关联的 IO 号，从 1 开始

*byTriggerType*

触发模式：0- 下降沿，1- 上升沿

*byRes1*

保留，置为 0

***struRS485*** 为 RS485 触发方式相关参数结构体

*byRelateChan*

关联车检器通道号，取值范围：[1,16]

*byRes2*

保留，置为 0

*byCarDriveDirect*

车辆行驶方向，具体定义如下：

```

enum _ITC_LANE_CAR_DRIVE_DIRECT_{
    ITC_LANE_DRIVE_UNKNOW    = 0,
    ITC_LANE_DRIVE_UP_TO_DOWN = 1,
    ITC_LANE_DRIVE_DOWN_TO_UP = 2
}ITC_LANE_CAR_DRIVE_DIRECT

```

*ITC\_LANE\_DRIVE\_UNKNOW*

未知

*ITC\_LANE\_DRIVE\_UP\_TO\_DOWN*

从上往下行驶(下行)

*ITC\_LANE\_DRIVE\_DOWN\_TO\_UP*

从下往上行驶(上行)

*byRes*

保留

*struLaneLine*

车道线

*struPlateRecog*

牌识区域参数

*byRelaLaneDirectionType*

关联车道方向类型，定义详见 [ITC\\_RELA\\_LANE\\_DIRECTION\\_TYPE](#)

*byRes1*

保留

### Remarks

NET\_ITC\_POST\_MPR\_PARAM 中 bySourceType 为 1 时联合体 uTssParamInfo 中 struIO 有效，bySourceType 为 2 时联合体 uTssParamInfo 中 struRS485 有效。

## 7.195 NET\_ITC\_LANE\_NOCOMITY\_PEDESTRIAN\_PARAM:不礼让行人触发模式车道参数

struct{

```

    BYTE                byRelatedDriveWay;
    BYTE                byRelaLaneDirectionType;
    BYTE                byPedestriansNum;
    BYTE                byVehicleSpeed;
    DWORD               dwVehicleInterval;
    BYTE                byPedesDetRule;
    BYTE                byRes[3];
    NET\_ITC\_LINE       struLaneLine;
    NET\_ITC\_LINE       struStopLine;
    NET\_ITC\_POLYGON    struPlateRecog;
    BYTE                byRes1[280];

```

}NET\_ITC\_LANE\_NOCOMITY\_PEDESTRIAN\_PARAM,\*LPNET\_ITC\_LANE\_NOCOMITY\_PEDESTRIAN\_PARAM;

### Members

*byRelatedDriveWay*

关联的车道号

*byRelaLaneDirectionType*

关联车道方向类型

*byPedestriansNum*

行人数量阈值，取值范围：1~100，默认值：1

*byVehicleSpeed*

车辆速度阈值，取值范围：1~100，默认值：0

*dwVehicleInterval*

跟车检测阈值，取值范围：0~65536，默认值：0

*byPedesDetRule*

行人检测规则，代表行人检测区域中行人行驶方向：0- 向左，1- 向右

*byRes*

保留，置为 0

*struLaneLine*  
车道线

*struStopLine*  
停止线

*struPlateRecog*  
牌识区域

*byRes1*  
保留，置为 0

## 7.196 NET\_ITC\_LANE\_PARAM: 车道参数

```

struct{
    BYTE                byEnable;
    BYTE                byRelatedDriveWay;
    WORD               wDistance;
    WORD               wTrigDelayTime;
    BYTE               byTrigDelayDistance;
    BYTE               bySpeedCapEn;
    BYTE               bySignSpeed;
    BYTE               bySpeedLimit;
    BYTE               bySnapTimes;
    BYTE               byOverlayDriveWay;
    NET\_ITC\_INTERVAL\_PARAM    struInterval;
    BYTE               byRelatedIOOut[MAX_IOOUT_NUM];
    BYTE               byFlashMode;
    BYTE               byCartSignSpeed;
    BYTE               byCartSpeedLimit;
    BYTE               byRelatedIOOutEx;
    NET\_ITC\_PLATE\_RECOG\_REGION\_PARAM struPlateRecog[MAX_LANEAREA_NUM];
    BYTE               byLaneType;
    BYTE               byUseageType;
    BYTE               byRelaLaneDirectionType;
    BYTE               byLowSpeedLimit;
    BYTE               byBigCarLowSpeedLimit;
    BYTE               byLowSpeedCapEn;
    BYTE               byEmergencyCapEn;
    BYTE               byRes[9];
}NET_ITC_LANE_PARAM,*LPNET_ITC_LANE_PARAM;

```

### Members

*byEnable*  
是否启用该车道：0- 不启用，1- 启用

*byRelatedDriveWay*  
关联的车道号

*wDistance*

线圈距离，计算速度

*wTrigDelayTime*

触发延迟时间（默认 200），单位：毫秒

*byTrigDelayDistance*

触发延迟距离（默认 0），单位：分米

*bySpeedCapEn*

是否启用超速抓拍：0- 否，1- 是

*bySignSpeed*

标志限速，单位：km/h

*bySpeedLimit*

限速值，单位：km/h

*bySnapTimes*

抓拍次数（默认 1）：0- 不抓拍；非 0- 连拍次数，最大 5

*byOverlayDriveWay*

OSD 叠加的车道号

*struInterval*

抓拍间隔参数

*byRelatedIOOut*

关联的 IO 输出口，可以同时关联多个

*byFlashMode*

闪光灯闪烁模式：0- 同时闪，1- 轮流闪

*byCartSignSpeed*

标志限速(大车)，单位：km/h

*byCartSpeedLimit*

限速值（大车），单位：km/h

*byRelatedIOOutEx*

关联的 IO 输出口，按位表示输出口号：bit0 表示 IO 输出口 1，bit1 表示 IO 输出口 2，以此类推。  
值：0-不关联，1-关联，最多支持关联到 8 个 IO 输出口(兼容 *byRelatedIOOut* 字段)

*struPlateRecog*

车道牌识区域参数

*byLaneType*

车道类型：0- 未配置，1- 高速公路，2- 城市快速路，0xff- 其他道路

*byUseageType*

车道用途类型，定义详见 [ITC\\_LANE\\_USEAGE\\_TYPE](#)

*byRelaLaneDirectionType*

关联车道方向类型，定义详见 [ITC\\_RELA\\_LANE\\_DIRECTION\\_TYPE](#)

*byLowSpeedLimit*

小车限低速值，单位：km/h

*byBigCarLowSpeedLimit*

大车限低速值，单位：km/h

*byLowSpeedCapEn*

是否启用低速抓拍：0- 否，1- 是

*byEmergencyCapEn*

是否启用应急车道抓拍：0- 否，1- 是

*byRes*

保留

### Remarks

该结构体中的参数 *byRelatedDriveWay* 为关联车道号，是和车检器中的车道关联匹配，实现抓拍。*byOverlayDriveWay* 为叠加车道号，为实际车道号。原 *bySignSpeed* 和 *bySpeedLimit* 为小车适用，卡口模式一般使用于高速场景，新增 *byCartSignSpeed* 和 *byCartSpeedLimit* 为卡车适用，电警模式下不需要区分大小车。

## 7.197 NET\_ITC\_LANE\_PRS\_PARAM: 视频检测触发模式车道参数

```
struct{
    BYTE                byLaneNO;
    union{
        BYTE            uLen[4];
        struct{
            BYTE        byIONo;
            BYTE        byTriggerType;
            BYTE        byRes1[2];
        }struIO;
        struct{
            BYTE        byRelateChan;
            BYTE        byRes2[3];
        }struRS485;
    }uTssParamInfo;
    BYTE                byRes[59];
    NET_ITC_LINE        struLaneLine;
    NET_ITC_POLYGON     struPlateRecog;
    BYTE                byRelaLaneDirectionType;
    BYTE                byRes1[255];
}NET_ITC_LANE_PRS_PARAM,*LPNET_ITC_LANE_PRS_PARAM;
```

### Members

*byLaneNO*

车道号

***uTssParamInfo*** 为触发参数信息联合体

*uLen*

联合体大小，4 个字节

***struIO*** 为 IO 触发方式相关参数结构体

*byIONo*

关联的 IO 号，从 1 开始

*byTriggerType*

触发模式：0- 下降沿，1- 上升沿

*byRes1*

保留，置为 0

***struRS485*** 为 RS485 触发方式相关参数结构体

*byRelateChan*



关联车检器通道号，取值范围：[1,16]

*byRes2*

保留，置为 0

*byRes*

保留

*struLaneLine*

车道线

*struPlateRecog*

牌识区域

*byRelaLaneDirectionType*

关联车道方向类型，定义详见 [ITC\\_RELA\\_LANE\\_DIRECTION\\_TYPE](#)

*byRes1*

保留

## 7.198 **NET\_ITC\_LANE\_VIDEO\_EPOLICE\_PARAM**:视频电警触发模式车道

### 参数

```

struct{
    BYTE                byLaneNO;
    BYTE                bySensitivity;
    BYTE                byEnableRadar;
    BYTE                byRes1;
    NET\_ITC\_LANE\_LOGIC\_PARAM    struLane;
    NET\_ITC\_VIOLATION\_DETECT\_PARAM    struVioDetect;
    NET\_ITC\_VIOLATION\_DETECT\_LINE    struLine;
    NET\_ITC\_POLYGON                struPlateRecog;
    BYTE                byRecordEnable;
    BYTE                byRecordType;
    BYTE                byPreRecordTime;
    BYTE                byRecordDelayTime;
    BYTE                byRecordTimeOut;
    BYTE                byCarSpeedLimit;
    BYTE                byCarSignSpeed;
    BYTE                bySnapPicPreRecord;
    NET\_ITC\_INTERVAL\_PARAM        struInterval;
    BYTE                byRes[36];
}NET_ITC_LANE_VIDEO_EPOLICE_PARAM, *LPNET_ITC_LANE_VIDEO_EPOLICE_PARAM;

```

#### Members

*byLaneNO*

关联的车道号

*bySensitivity*

线圈灵敏度，取值范围：[1,100]

*byEnableRadar*

启用雷达测试：0- 不启用，1- 启用

*byRes1*

保留，置为 0

*struLane*

车道参数

*struVioDetect*

违规检测参数

*struLine*

违规检测线

*struPlateRecog*

牌识区域参数

*byRecordEnable*

闯红灯周期录像标志：0-不录像，1-录像

*byRecordType*

闯红灯录像类型：0-预录，1-延时录像

*byPreRecordTime*

闯红灯录像片段预录时间（默认 0），单位：秒

*byRecordDelayTime*

闯红灯录像片段延时时间（默认 0），单位：秒

*byRecordTimeOut*

闯红灯周期录像超时时间，单位：秒

*byCarSpeedLimit*

车速限制值，单位：km/h

*byCarSignSpeed*

标志限速，单位：km/h

*bySnapPicPreRecord*

抓拍图片预录时间点：0- 默认值（第二张图片），1- 第一张图片，2- 第二张图片，3- 第三张图片

*struInterval*

抓拍间隔参数

*byRes*

保留，置为 0

## 7.199 NET\_ITC\_LIGHT\_ACCESSPARAM\_UNION:交通信号灯接入参数联

### 合体

```
union {
    DWORD uLen[122];
    NET_ITC_IO_LIGHT_PARAM struIOLight;
    NET_ITC_RS485_LIGHT_PARAM struRS485Light;
    NET_ITC_VIDEO_DETECT_LIGHT_PARAM struVideoDelectLight;
}NET_ITC_LIGHT_ACCESSPARAM_UNION, *LPNET_ITC_LIGHT_ACCESSPARAM_UNION;
```

#### Members

*uLen*

联合体大小

*struOLight*

IO 接入信号灯参数

*struRS485Light*

RS485 接入信号灯参数

*struRS485Light*

视频检测信号灯参数

## 7.200 NET\_ITC\_LINE: 视频电警线结构体

```
struct{
    NET_VCA_LINE    struLine;
    BYTE            byLineType;
    BYTE            byRes[7];
}NET_ITC_LINE, *LPNET_ITC_LINE;
```

### Members

*struLine*

线参数

*byLineType*

线类型，具体定义如下：

```
enum _ITC_LINE_TYPE_{
    ITC_LINT_UNKNOW        = 0,
    ITC_LINE_WHITE        = 1,
    ITC_LINE_STOP          = 2,
    ITC_LINE_SINGLE_YELLOW = 3,
    ITC_LINE_DOUBLE_YELLOW = 4,
    ITC_LINE_GUARD_RAIL    = 5,
    ITC_LINE_NO_CROSS      = 6
}ITC_LINE_TYPE
```

*ITC\_LINT\_UNKNOW*

未知

*ITC\_LINE\_WHITE*

车道间的白实线

*ITC\_LINE\_STOP*

车道停止线

*ITC\_LINE\_SINGLE\_YELLOW*

单黄线

*ITC\_LINE\_DOUBLE\_YELLOW*

双黄线

*ITC\_LINE\_GUARD\_RAIL*

车道线上有护栏

*ITC\_LINE\_NO\_CROSS*

车辆无法跨越的车道线

*byRes*

保留，置为 0

## 7.201 NET\_ITC\_NOCOMITY\_PEDESTRIAN\_PARAM:不礼让行人触发参数

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
    BYTE                byRes[74];
    NET\_ITC\_LINE        struLaneBoundaryLine;
    NET\_ITC\_LINE        struTriggerLine;
    NET\_ITC\_POLYGON     struPedesDetRecog;
    NET\_ITC\_LANE\_NOCOMITY\_PEDESTRIAN\_PARAM struLaneParam[MAX_ITC_LANE_NUM];
    NET\_ITC\_PLATE\_RECOG\_PARAM struPlateRecog;
    BYTE                byRes1[400];
}NET_ITC_NOCOMITY_PEDESTRIAN_PARAM,*LPNET_ITC_NOCOMITY_PEDESTRIAN_PARAM;
```

### Members

#### *byEnable*

使能：0- 禁用，1- 启用

#### *byLaneNum*

识别的车道个数，1~3

#### *byRes*

保留，置为 0

#### *struLaneBoundaryLine*

车道边界线（最右边车道的右车道线）

#### *struTriggerLine*

不礼让行人触发线

#### *struPedesDetRecog*

行人检测区域

#### *struLaneParam*

车道参数，每位数组表示一个车道

#### *struPlateRecog*

车牌识别参数

#### *byRes1*

保留，置为 0

## 7.202 NET\_ITC\_PLATE\_RECOG\_PARAM:牌识参数

```
struct{
    BYTE                byDefaultCHN[MAX_CHJC_NUM];
    BYTE                byEnable;
    DWORD               dwRecogMode;
    BYTE                byVehicleLogoRecog;
    BYTE                byProvince;
```

```

BYTE    byRegion;
BYTE    byRes1;
WORD    wPlatePixelWidthMin;
WORD    wPlatePixelWidthMax;
BYTE    byRes[24];
}NET_ITC_PLATE_RECOG_PARAM,*LPNET_ITC_PLATE_RECOG_PARAM;

```

### Members

#### *byDefaultCHN*

设备运行省份的汉字简写

#### *byEnable*

是否启用该区域牌识：0- 否，1- 是

#### *dwRecogMode*

识别的类型， bit0-背向识别：0- 正向车牌识别，1- 背向识别(尾牌识别)

bit1-大车牌识别或小车牌识别：0- 小车牌识别，1- 大车牌识别

bit2-车身颜色识别：0- 不采用车身颜色识别，在背向识别或小车牌识别时禁止启用；1- 车身颜色识别

bit3-农用车识别：0- 不采用农用车识别，1- 农用车识别

bit4-模糊识别：0- 不采用模糊识别，1- 模糊识别

bit5-帧定位或场定位：0- 帧定位，1- 场定位

bit6-帧识别或场识别：0- 帧识别，1- 场识别

bit7-晚上或白天：0- 白天，1- 晚上

bit8-摩托车识别：0- 不采用摩托车识别，1- 摩托车识别

bit9-场景模式：0- 电警/多帧，1- 卡口

bit10-微小车牌：0- 不启用，1- 启用微小车牌识别(像素 60~80)

bit12-民航车牌识别：0-不启用，1-开启民航车牌识别

bit13-车牌过渡倾斜处理：0-不启用，1-开启过渡倾斜处理（PRS）

bit14-超大车牌识别：0-不启用，1-开启超大车牌识别（PRS）

bit15-遮阳板检测：0-不启用，1-启用遮阳板检测

bit16-黄标车检测：0-不启用，1-启用黄标车检测

bit17-危险品车辆检测：0-不启用，1-启用危险品车辆检测

bit18-使馆车牌识别：0-不启用，1-启用使馆车牌识别

bit19-车辆子品牌识别：0-不启用，1-启用车牌子品牌识别

#### *byVehicleLogoRecog*

汽车品牌识别：0- 不启用，1- 启用

#### *byProvince*

省份索引值：0-保留，1-澳，2-京，3-渝，4-闽，5-甘，6-粤，7-桂，8-贵，9-琼，10-冀，11-豫，12-黑，13-鄂，14-湘，15-吉，16-苏，17-赣，18-辽，19-蒙，20-宁，21-青，22-鲁，23-晋，24-陕，25-沪，26-川，27-台，28-津，29-藏，30-港，31-新，32-云，33-浙，34-皖，0xff-全部

#### *byRegion*

区域索引值：0- 保留，1- 欧洲(Europe Region)，2- 俄罗斯(Russian Region)，3- 欧洲&俄罗斯(EU&CIS)

#### *byRes1*

保留

#### *wPlatePixelWidthMin*

车牌像素识别宽度最小值，单位：像素，当前推荐范围：[130,500]

**wPlatePixelWidthMax**

车牌像素识别宽度最大值，单位：像素，当前推荐范围：[130,500]

**byRes**

保留

**Remarks**

摩托车牌识别用于背向识别，也就是说电警车检器、卡警车检器和视频电警会用到，卡口均不使用。

**7.203 NET\_ITC\_PLATE\_RECOG\_REGION\_PARAM:牌识区域参数**

```
struct{
    BYTE    byMode;
    BYTE    byRes1[3];
    union{
        NET\_VCA\_RECT          struRect;
        NET\_ITC\_POLYGON       struPolygon;
    }uRegion;
    BYTE    byRes[16];
}NET_ITC_PLATE_RECOG_REGION_PARAM,*LPNET_ITC_PLATE_RECOG_REGION_PARAM;
```

**Members****byMode**

区域类型：0- 矩形，1- 多边形

**byRes1**

保留

**struRect**

矩形区域

**struPolygon**

多边形区域

**byRes**

保留

**7.204 NET\_ITC\_POLYGON:多边形**

```
struct{
    DWORD    dwPointNum;
    NET\_VCA\_POINT    struPos[ITC_MAX_POLYGON_POINT_NUM];
}NET_ITC_POLYGON,*LPNET_ITC_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，最多 20 个

## 7.205 NET\_ITC\_POST\_HVT\_PARAM\_V50:混行卡口参数

```

struct{
    BYTE                byLaneNum;
    BYTE                byCapType;
    BYTE                byCapMode;
    BYTE                bySceneMode;
    BYTE                bySpeedMode;
    BYTE                byLineRuleEffect;
    BYTE                byRes1[78];
    NET_ITC_LINE        struLeftTrigLine;
    NET_ITC_LINE        struRightTrigLine;
    NET_ITC_LINE        struLaneBoundaryLine;
    NET_ITC_POLYGON     struDetectArea;
    NET_DVR_GEOGLOCATION struGeogLocation;
    NET_ITC_LANE_HVT_PARAM_V50 struLaneParam[MAX_ITC_LANE_NUM];
    NET_ITC_PLATE_RECOG_PARAM struPlateRecog;
    BYTE                byRes2[260];
}NET_ITC_POST_HVT_PARAM_V50,*LPNET_ITC_POST_HVT_PARAM_V50;

```

### Members

#### *byLaneNum*

识别的车道个数，1~6

#### *byCapType*

抓拍类型：0- 机、非、人（默认），1- 机动车

#### *byCapMode*

抓拍方式：0- 视频抽帧，1- 打断抓拍，2- 混合模式

#### *bySceneMode*

场景模式：0- 城区道路（默认），1- 小区出入口，2- 高速公路

#### *bySpeedMode*

测速模式：0- 无测速，1- 雷达测速，2- 视频测速

#### *byLineRuleEffect*

触发规则线有效性，每一位代表一条触发线，取值：0- 无效，1- 有效

bit0- 左触发线

bit1- 右触发线

bit2- 视频检测区域

#### *byRes1*

保留，置为 0

#### *struLeftTrigLine*

左触发线(一条垂直线)

#### *struRightTrigLine*

右触发线(一条垂直线)

#### *struLaneBoundaryLine*

车道边界线（最右边车道的右车道线）

*struDetectArea*

视频检测区域

*struGeogLocation*

地址位置（默认浙江）

*struLaneParam*

单车道属性，每位数组表示一个车道信息

*struPlateRecog*

车牌识别参数

*byRes2*

保留，置为 0

## 7.206 NET\_ITC\_POST\_IMT\_PARAM:智慧监控配置参数

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
    BYTE                bySnapMode;
    BYTE                byRes[61];
    NET\_ITC\_PLATE\_RECOG\_PARAM    struPlateRecog;
    NET\_ITC\_LINE                struLaneBoundaryLine;
    NET\_ITC\_LANE\_IMT\_PARAM    struLaneParam[MAX_ITC_LANE_NUM];
    BYTE                byRes1[1584];
}NET_ITC_POST_IMT_PARAM,*LPNET_ITC_POST_IMT_PARAM;
```

### Members

*byEnable*

是否启用智慧监控模式：0- 禁用，1- 启用

*byLaneNum*

识别的车道个数，1~6

*bySnapMode*

抓拍类型：0- 机动车，1- 机非人

*byRes*

保留，置为 0

*struPlateRecog*

车牌识别参数

*struLaneBoundaryLine*

车道边界线（最左边车道的左边界线）

*struLaneParam*

单车道属性

*byRes1*

保留，置为 0



## 7.207 NET\_ITC\_POST\_IOSPEED\_PARAM: 卡口 IO 测速参数

```
struct{
    NET_ITC_PLATE_RECOG_PARAM    struPlateRecog;
    NET_ITC_SINGLE_IOSPEED_PARAM struSingleIOSpeed[MAX_IOSPEED_GROUP_NUM];
    BYTE                          byRes[32];
}NET_ITC_POST_IOSPEED_PARAM,*LPNET_ITC_POST_IOSPEED_PARAM;
```

### Members

#### *struPlateRecog*

牌识参数

#### *struSingleIOSpeed*

单个 IO 测速组参数

#### *byRes*

保留

## 7.208 NET\_ITC\_POST\_MOBILE\_PARAM: 移动交通触发参数

```
struct{
    BYTE    byEnable;
    BYTE    bySceneMode;
    WORD    wExpressWayCapType;
    WORD    wUrbanRoadCapType;
    BYTE    byCapNum;
    BYTE    byRecordEnable;
    DWORD   dwPreRecordTime;
    DWORD   dwOverRecordTime;
    BYTE    byRes[256];
}NET_ITC_POST_MOBILE_PARAM,*LPNET_ITC_POST_MOBILE_PARAM;
```

### Members

#### *byEnable*

是否启用：0- 否，1- 是

#### *bySceneMode*

场景模式：0- 高速公路，1- 城市道路

#### *wExpressWayCapType*

高速公路抓拍类型，按位表示，bit0- 卡口，bit1- 大车占道，bit2- 压硬路肩

#### *wUrbanRoadCapType*

城市道路抓拍类型，按位表示，bit0- 卡口，bit1- 机占非，bit2- 占用专用车道

#### *byCapNum*

抓拍张数，取值范围：[2,3]

#### *byRecordEnable*

违章录像使能：0- 关闭，1- 开启

#### *dwPreRecordTime*

录像预录时间，单位：秒(s)

*dwOverRecordTime*

录像超时时间，单位：秒(s)

*byRes*

保留，置为 0

## 7.209 NET\_ITC\_POST\_MPR\_PARAM:卡口多帧检测(MPR)触发参数

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
    BYTE                bySourceType;
    BYTE                byPicUploadType;
    BYTE                byRes[60];
    NET_ITC_LINE        struLaneBoundaryLine;
    NET_ITC_PLATE_RECOG_PARAM struPlateRecog;
    NET_ITC_LANE_MPR_PARAM struLaneParam[MAX_ITC_LANE_NUM];
    BYTE                byRes1[440];
}NET_ITC_POST_MPR_PARAM,*LPNET_ITC_POST_MPR_PARAM;
```

### Members

*byEnable*

是否启用：0- 不启用，1- 启用

*byLaneNum*

识别的车道个数

*bySourceType*

信号源类型：0- MPR 触发（视频触发），1- 关联 IO 触发（地感线圈），2- 关联 RS485 触发

*byPicUploadType*

图片上传类型：0- 全部上传，1- 正向上传，2- 背向上传

*byRes*

保留

*struLaneBoundaryLine*

车道边界线（最左边车道的左边界线）

*struPlateRecog*

车牌识别参数

*struLaneParam*

多帧检测车道参数

*byRes1*

保留

## 7.210 NET\_ITC\_POST\_PRS\_PARAM:视频检测（PRS）触发参数

```
struct{
    BYTE                byEnable;
    BYTE                byLaneNum;
```

```

BYTE                bySourceType;
BYTE                bySnapMode;
BYTE                byCapMode;
BYTE                byRes[59];
NET\_ITC\_LANE      struLaneBoundaryLine;
NET\_ITC\_PLATE\_RECOG\_PARAM struPlateRecog;
NET\_ITC\_LANE\_PRS\_PARAM struLaneParam[MAX_ITC_LANE_NUM];
BYTE                byRes1[440];
}NET_ITC_POST_PRS_PARAM,*LPNET_ITC_POST_PRS_PARAM;

```

### Members

#### *byEnable*

是否启用：0- 否，1- 是

#### *byLaneNum*

识别的车道个数

#### *bySourceType*

信号源类型：0- 视频检测，1- 关联 IO 触发（地感线圈），2- 关联 RS485 的触发信号

#### *bySnapMode*

抓拍模式：0- 全景图，1- 全景图+特写

#### *byCapMode*

*bySourceType* 为 0 视频检测时使用，0- 频闪模式，1- 爆闪模式

#### *byRes*

保留，置为 0

#### *struLaneBoundaryLine*

车道边界线（最左边车道的左边界线）

#### *struPlateRecog*

牌识参数

#### *struLaneParam*

车道参数

#### *byRes1*

保留，置为 0

## 7.211 **NET\_ITC\_POST\_RS485\_PARAM:**卡口 RS485 车检器触发参数

```

struct{
  BYTE                byRelatedLaneNum;
  BYTE                byTriggerSpareMode;
  BYTE                byFaultToleranceTime;
  BYTE                byRes1;
  NET\_ITC\_PLATE\_RECOG\_PARAM struPlateRecog;
  NET\_ITC\_LANE\_PARAM struLane[MAX_ITC_LANE_NUM];
  BYTE                byRes[32];
}NET_ITC_POST_RS485_PARAM,*LPNET_ITC_POST_RS485_PARAM;

```

### Members

#### *byRelatedLaneNum*

关联的车道个数

*byTriggerSpareMode*

触发备用模式：0-无，1-卡口虚拟线圈模式，2- 卡口混合车道模式

*byFaultToleranceTime*

容错时间(单位:分钟)，用于检测车检器是否正常的最大时间

*byRes1*

保留

*struPlateRecog*

牌识参数

*struLane*

关联的车道参数

*byRes*

保留

## 7.212 NET\_ITC\_POST\_RS485\_RADAR\_PARAM:卡口 RS485 雷达触发参数

```
struct{
    BYTE                byRelatedLaneNum;
    BYTE                byRes1[3];
    NET\_ITC\_PLATE\_RECOG\_PARAM    struPlateRecog;
    NET\_ITC\_LANE\_PARAM          struLane[MAX_ITC_LANE_NUM];
    NET\_ITC\_RADAR\_PARAM        struRadar;
    BYTE                byRes[32];
}NET_ITC_POST_RS485_RADAR_PARAM,*LPNET_ITC_POST_RS485_RADAR_PARAM;
```

### Members

*byRelatedLaneNum*

关联的车道个数

*byRes1*

保留

*struPlateRecog*

牌识参数

*struLane*

关联的车道参数

*struRadar*

雷达参数

*byRes*

保留

## 7.213 NET\_ITC\_POST\_SINGLEIO\_PARAM:单 IO 触发参数

```
struct{
    NET\_ITC\_PLATE\_RECOG\_PARAM    struPlateRecog;
    NET\_ITC\_SINGLEIO\_PARAM      struSingleIO[MAX_IOIN_NUMEX];
}
```

```

    BYTE                                byRes[360];
}NET_ITC_POST_SINGLEIO_PARAM,*LPNET_ITC_POST_SINGLEIO_PARAM;

```

### Members

#### *struPlateRecog*

牌识参数

#### *struSingleIO*

单个 IO 触发参数，数组 0 表示 IO1，数组 1 表示 IO2，依次类推

#### *byRes*

保留

## 7.214 NET\_ITC\_POST\_VTCOIL\_PARAM: 卡口虚拟线圈触发参数

```

struct{
    BYTE                                byRelatedLaneNum;
    BYTE                                byIsDisplay;
    BYTE                                byLoopPos;
    BYTE                                byPolarLenType;
    BYTE                                byDayAuxLightMode;
    BYTE                                byVideoLaneNO;
    BYTE                                byVideoLowTh;
    BYTE                                byVideoHighTh;
    BYTE                                byRecordMode;
    BYTE                                bySnapMode;
    BYTE                                bySpeedDetector;
    BYTE                                byRes2;
    WORD                                wResolutionX;
    WORD                                wResolutionY;
    DWORD                               dwDayInitExp;
    DWORD                               dwDayMaxExp;
    DWORD                               dwNightExp;
    DWORD                               dwSnapExp;
    BYTE                                byDayInitGain;
    BYTE                                byDayMaxGain;
    BYTE                                byNightGain;
    BYTE                                bySnapGain;
    DWORD                               dwSceneMode;
    NET\_DVR\_GEOGLOCATION                struGeogLocation;
    NET\_ITC\_PLATE\_RECOG\_PARAM          struPlateRecog;
    NET\_ITC\_VTCOIL\_INFO                struVtCoil[MAX_VL_NUM];
    NET\_ITC\_RADAR\_PARAM                struRadar;
    NET\_VCA\_LINE                        struLine;
    DWORD                               dwVioDetectType;
    BYTE                                byDebugMode;
    BYTE                                byRes[11];

```

```
}NET_ITC_POST_VTCOIL_PARAM,*LPNET_ITC_POST_VTCOIL_PARAM;
```

### Members

*byRelatedLaneNum*

关联的车道个数

*byIsDisplay*

视频中是否显示虚拟线圈：0- 不显示，1- 显示

*byLoopPos*

晚间触发线圈的偏向（默认 10）

*byPolarLenType*

偏振镜类型：0- 不加偏振镜，1- 加施耐德偏振镜

*byDayAuxLightMode*

白天辅助照明模式：0- 无辅助照明，1- LED 灯照明，2- 闪光灯照明

*byVideoLaneNO*

视频参考亮度的参考车道号

*byVideoLowTh*

视频参考亮度低阈值初始化值（默认 40）

*byVideoHighTh*

视频参考亮度高阈值初始化值（默认 55）

*byRecordMode*

录像标志：0- 不录像，1- 录像

*bySnapMode*

抓拍模式：0- 频闪模式，1- 爆闪模式

*bySpeedDetector*

测速方式：0- 不测速，0x1- 雷达测速，0x2- 视频测速

*byRes2*

保留

*wResolutionX*

设备当前分辨率宽

*wResolutionY*

设备当前分辨率高

*dwDayInitExp*

视频白天曝光时间的初始值，默认为 2000

*dwDayMaxExp*

视频白天曝光时间的最大值，默认为 20000

*dwNightExp*

晚间视频曝光时间的设置值，默认为 3000

*dwSnapExp*

抓拍曝光时间

*byDayInitGain*

视频白天增益的初始值，默认为 200

*byDayMaxGain*

视频白天增益的最大值，默认为 400

*byNightGain*

晚间视频增益

*bySnapGain*

抓拍增益

*dwSceneMode*

场景模式

```
enum _SCENE_MODE_{
    UNKOWN_SCENE_MODE = 0,
    HIGHWAY_SCENE_MODE = 1,
    SUBURBAN_SCENE_MODE = 2,
    URBAN_SCENE_MODE = 3,
    TUNNEL_SCENE_MODE = 4
}SCENE_MODE
```

*UNKOWN\_SCENE\_MODE*

未知场景模式

*HIGHWAY\_SCENE\_MODE*

高速场景模式

*SUBURBAN\_SCENE\_MODE*

郊区场景模式(保留)

*URBAN\_SCENE\_MODE*

市区场景模式

*TUNNEL\_SCENE\_MODE*

隧道场景模式(保留)

*struGeogLocation*

地址位置(默认浙江)

*struPlateRecog*

牌识参数

*struVtCoil*

虚拟线圈参数

*struRadar*

雷达参数

*struLine*

右车道线

*dwVioDetectType*

违规检测类型，按位表示，其值：0- 不启用，1- 启用，类型定义详见 [ITC\\_VIOLATION\\_DETECT\\_TYPE](#)

*byDebugMode*

调试模式：0- 不启用，1- 启用

*byRes*

保留

## 7.215 NET\_ITC\_RADAR\_INFO\_PARAM:RS485 关联雷达配置参数

struct{

```
NET\_ITC\_RADAR\_PARAM struRadarParam;
BYTE byAssociateLaneNo;
BYTE byRes[103];
```

```
}NET_ITC_RADAR_INFO_PARAM, *LPNET_ITC_RADAR_INFO_PARAM;
```

### Members

*struRadarParam*

雷达参数

*byAssociateLaneNo*

关联车道号，按位表示车道：bit1-车道 1，bit2-车道 2，bit3-车道 3，bit4-车道 4，bit5-车道 5，bit6-车道 6，值：0- 不关联，1- 关联

*byRes*

保留，置为 0

## 7.216 NET\_ITC\_RADAR\_PARAM: 雷达参数

```
struct{
```

```
    BYTE    byRadarType;
```

```
    BYTE    byLevelAngle;
```

```
    WORD    wRadarSensitivity;
```

```
    WORD    wRadarSpeedValidTime;
```

```
    BYTE    byRes1[2];
```

```
    float   fLineCorrectParam;
```

```
    int     iConstCorrectParam;
```

```
    BYTE    byRes[8];
```

```
}NET_ITC_RADAR_PARAM,*LPNET_ITC_RADAR_PARAM;
```

### Members

*byRadarType*

雷达类型：0- 无雷达，1- 安道雷雷达，2- 奥利维亚，3- 川速微波，4- 雷达接 IO 扩展盒(此参数在卡口虚拟线圈界面中使用，卡口 RS485 雷达不使用)，5- 安道雷（无雷达控制器），0xff- 其它类型

*byLevelAngle*

与水平线所成角度，默认为 25°，取值范围：0~90 度

*wRadarSensitivity*

雷达灵敏度

*wRadarSpeedValidTime*

雷达速度有效时间，取值范围：(0,2000]，0 表示不支持

*byRes1*

保留

*fLineCorrectParam*

线性矫正参数(乘操作)，取值范围：[0.0,2.0]

*iConstCorrectParam*

常量矫正参数(加减操作)，取值范围：[-100,100]

*byRes*

保留

### Remarks

卡口 RS485 雷达触发模式下，雷达检测车辆通过时要及时进行触发，速度值和抓拍信号是几乎同时接收，因此该模式下 *wRadarSpeedValidTime* 无效。



## 7.217 NET\_ITC\_REDLIGHT\_PEDESTRIAN\_PARAM:行人闯红灯触发参数

```

struct{
    BYTE                byEnable;
    BYTE                bySnapNumTimes;
    BYTE                byPedesDir;
    BYTE                byDelayTime;
    BYTE                byStackTargetEnble;
    BYTE                byCalibRecogCtrl;
    BYTE                byRes1[2];
    NET\_ITC\_TRAFFIC\_LIGHT\_PARAM    struTrafficLight;
    NET\_ITC\_LINE                struStopLine;
    NET\_ITC\_POLYGON            struCalibRecog[MAX_CALIB_RECOG_NUM];
    BYTE                byRes[440];
}NET_ITC_REDLIGHT_PEDESTRIAN_PARAM,*LPNET_ITC_REDLIGHT_PEDESTRIAN_PARAM;

```

### Members

#### *byEnable*

使能：0- 禁用，1- 启用

#### *bySnapNumTimes*

抓拍张数，取值范围：1~3，默认 3 张

#### *byPedesDir*

行人方向：0- 正向，1- 背向，2- 双向

#### *byDelayTime*

延时时间，取值范围：1~5，单位：s（秒）

#### *byStackTargetEnble*

叠加目标框（即抓拍第一张图片上框住闯红灯的行人）：0- 不叠加，1- 叠加

#### *struTriggerLine*

标定区域控制：0- 移除标定区域，1- 添加标定区域

#### *byRes1*

保留，置为 0

#### *struTrafficLight*

交通信号灯参数

#### *struStopLine*

停止线

#### *struCalibRecog*

标定区域

#### *byRes*

保留，置为 0

## 7.218 NET\_ITC\_RS485\_ACCESS\_CFG:RS485 关联接入设备配置

```

struct{
    DWORD                dwSize;

```

```

BYTE                byModeType;
BYTE                byRes[3];
NET\_ITC\_ACCESS\_DEVINFO\_PARAM\_UNION uITCAccessDevinfoParam;
BYTE                byRes1[12];
}NET_ITC_RS485_ACCESS_CFG, *LPNET_ITC_RS485_ACCESS_CFG;

```

### Members

#### *dwSize*

结构体大小

#### *byModeType*

接入设备类型：bit0- 预留，bit1- 雷达，bit2- 车检器，bit3- 信号灯检测器，取值：0- 不关联，1- 关联，目前只支持雷达

#### *byRes*

保留，置为 0

#### *uITCAccessDevinfoParam*

关联接入设备配置联合体

#### *byRes1*

保留，置为 0

## 7.219 **NET\_ITC\_RS485\_ACCESS\_INFO\_COND:RS485** 关联接入设备配置

### 条件参数

```

struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    DWORD    dwTriggerModeType;
    BYTE     byAssociateRS485No;
    BYTE     byRes[15];
}NET_ITC_RS485_ACCESS_INFO_COND, *LPNET_ITC_RS485_ACCESS_INFO_COND;

```

### Members

#### *dwSize*

结构体大小

#### *dwChannel*

设备通道号

#### *dwTriggerModeType*

触发模式，参见 [ITC\\_TRIGGERMODE\\_TYPE](#)

#### *byAssociateRS485No*

关联的 RS485 号，取值范围：1~5

#### *byRes*

保留，置为 0

## 7.220 NET\_ITC\_RS485\_LIGHT\_PARAM:RS485 接入信号灯参数

```
struct{
    NET_ITC_SINGLE_RS485_LIGHT_PARAM    struRS485Light[MAX_LIGHT_NUM];
    BYTE                                byRes[8];
}NET_ITC_RS485_LIGHT_PARAM,*LPNET_ITC_RS485_LIGHT_PARAM;
```

### Members

*struRS485Light*  
单个 485 接入信号灯参数

*byRes*  
保留，置为 0

### Remarks

当选择闯红灯模式为按方向（NET\_ITC\_VIDEO\_EPOLICE\_PARAM 里配置）时，三组可配置，即左转灯、直行灯、右转灯这三种灯，红、黄灯可配置。数组 struRS485Light[0]、struRS485Light[1]和 struRS485Light[2] 分别表示三个方向的配置。

## 7.221 NET\_ITC\_SERIAL\_CHECKINFO:串口参数

```
struct{
    BYTE    bySerialIndex[MAX_ITC_SERIALCHECK_NUM];
    BYTE    byRes[40];
}NET_ITC_SERIAL_CHECKINFO,*LPNET_ITC_SERIAL_CHECKINFO;
```

### Members

*bySerialIndex*  
485 串口校验结果：0- 无效，1- 有效，bySerialIndex[0]表示正确的头，bySerialIndex[1]表示正确的尾

*byRes*  
保留

## 7.222 NET\_ITC\_SERIAL\_INFO:车检器参数

```
struct{
    BYTE    bySerialProtocol;
    BYTE    byIntervalType;
    WORD    wInterval;
    BYTE    byNormalPassProtocol;
    BYTE    byInverseProtocol;
    BYTE    bySpeedProtocol;
    BYTE    byRes[9];
}NET_ITC_SERIAL_INFO,*LPNET_ITC_SERIAL_INFO;
```

### Members

*bySerialProtocol*

车检器协议类型，定义如下所示：

```
enum _ITC_SERIAL_PROTOCOL_{
    ITC_PROTOCOL_UNKNOW          = 0,
    ITC_SINGLE_COIL_PROTOCOL     = 1,
    ITC_DOUBLE_COIL_PROTOCOL_MODE1 = 2,
    ITC_DOUBLE_COIL_PROTOCOL_MODE2 = 3,
    ITC_DOUBLE_COIL_PROTOCOL_MODE3 = 4,
    ITC_DOUBLE_COIL_PROTOCOL_MODE4 = 5,
    ITC_DOUBLE_COIL_PROTOCOL_MODE5 = 6,
    ITC_DOUBLE_COIL_PROTOCOL_MODE6 = 7,
    ITC_SINGLE_COIL_PROTOCOL1    = 8,
    ITC_OTHER_PROTOCOL           = 0xff
}ITC_SERIAL_PROTOCOL
```

*ITC\_PROTOCOL\_UNKNOW*

未知

*ITC\_SINGLE\_COIL\_PROTOCOL*

单线圈车检器协议（动态显示：到达、离开、离开延迟抓拍，离开牌识）

*ITC\_DOUBLE\_COIL\_PROTOCOL\_MODE1*

双线圈车检器协议模式 1（动态显示：到达一、离开一、离开二抓拍，离开一牌识）

*ITC\_DOUBLE\_COIL\_PROTOCOL\_MODE2*

双线圈车检器协议模式 2（动态显示：离开一、离开二、离开二延时抓拍，离开一牌识，默认为延时距离 1m）

*ITC\_DOUBLE\_COIL\_PROTOCOL\_MODE3*

双线圈车检器协议模式 3（动态显示：进入二、离开一、离开二延时抓拍，离开一牌识，默认为延时距离 1m）

*ITC\_DOUBLE\_COIL\_PROTOCOL\_MODE4*

双线圈车检器协议模式 4（到达 2 线圈、离开 1 线圈、离开 2 线圈、离开 2 线圈延迟抓拍，离开 1 牌识，上传图像到达 2、离开 2、离开 2 延迟）

*ITC\_DOUBLE\_COIL\_PROTOCOL\_MODE5*

双线圈车检器协议模式 5（到达 2 线圈、离开 1 线圈、离开 2 线圈、离开 2 线圈延迟抓拍，到达 2、离开 1 牌识，上传图像到达 2、离开 2、离开 2 延迟）

*ITC\_DOUBLE\_COIL\_PROTOCOL\_MODE6*

双线圈车检器协议模式 6（到达 1 线圈、到达 2 线圈、离开 1 线圈、离开 2 线圈、离开 2 线圈延迟抓拍，到达 1、离开 1 牌识，上传图像到达 1、离开 2、离开 2 延迟）

*ITC\_SINGLE\_COIL\_PROTOCOL1*

单线圈模式 2（到达线圈、离开线圈、离开线圈延迟抓拍，到达和离开牌识）

*ITC\_OTHER\_PROTOCOL*

其它车检器协议

*byIntervalType*

间隔类型（默认按时间）：0- 时间起效，1- 距离起效

*wInterval*

连拍间隔时间（单位：ms）或连拍间隔距离（单位：分米）。当 *byIntervalType* 为 0 时，表示间隔时间，当 *byIntervalType* 为 1 时，表示距离

*byNormalPassProtocol*

正常过车抓拍协议类型，定义如下所示：

```
enum _ITC_NORMAL_PASS_SERIAL_PROTOCOL_{
    ITC_NORMAL_PASS_SERIAL_UNKNOW    = 0,
    ITC_NORMAL_PASS_PROTOCOL_MODE1   = 1,
    ITC_NORMAL_PASS_PROTOCOL_MODE2   = 2,
    ITC_NORMAL_PASS_PROTOCOL_MODE3   = 3,
    ITC_NORMAL_PASS_PROTOCOL_MODE4   = 4,
    ITC_NORMAL_PASS_PROTOCOL_MODE5   = 5
}ITC_NORMAL_PASS_SERIAL_PROTOCOL
```

*ITC\_NORMAL\_PASS\_SERIAL\_UNKNOW*

未知

*ITC\_NORMAL\_PASS\_PROTOCOL\_MODE1*

单线圈，离开抓拍，牌识上传（默认）

*ITC\_NORMAL\_PASS\_PROTOCOL\_MODE2*

线圈，到达、离开抓拍，均牌识，仅上传离开车辆图片和置信度高的车牌结果

*ITC\_NORMAL\_PASS\_PROTOCOL\_MODE3*

双线圈，离开 1 抓拍，牌识上传（默认）

*ITC\_NORMAL\_PASS\_PROTOCOL\_MODE4*

双线圈，到达 1、离开 1 抓拍，均牌识，仅上传离开 1 车辆图片和置信度高的车牌结果

*ITC\_NORMAL\_PASS\_PROTOCOL\_MODE5*

双线圈，到达 2、离开 1 抓拍，均牌识，仅上传离开 1 车辆图片和置信度高的车牌结果

*byInverseProtocol*

逆行抓拍协议类型，定义如下所示：

```
enum _ITC_INVERSE_SERIAL_PROTOCOL_{
    ITC_INVERSE_SERIAL_UNKNOW    = 0,
    ITC_INVERSE_PROTOCOL_MODE1   = 1,
    ITC_INVERSE_PROTOCOL_MODE2   = 2,
    ITC_INVERSE_PROTOCOL_MODE3   = 3
}ITC_INVERSE_SERIAL_PROTOCOL
```

*ITC\_INVERSE\_SERIAL\_UNKNOW*

未知

*ITC\_INVERSE\_PROTOCOL\_MODE1*

不抓拍

*ITC\_INVERSE\_PROTOCOL\_MODE2*

到达 1 连抓 2 张、离开 2 抓拍，到达 1 两张均牌识，上传 3 张车辆图片和置信度高的车牌结果

*ITC\_INVERSE\_PROTOCOL\_MODE3*

到达 1、离开 2 抓拍，到达 1 牌识，上传 2 张车辆图片和车牌结果

*bySpeedProtocol*

超速抓拍协议类型，定义如下所示：

```
enum _ITC_SPEED_SERIAL_PROTOCOL_{
    ITC_SPEED_SERIAL_UNKNOW    = 0,
    ITC_SPEED_PROTOCOL_MODE1   = 1,
    ITC_SPEED_PROTOCOL_MODE2   = 2,
    ITC_SPEED_PROTOCOL_MODE3   = 3
}
```

```
}ITC_SPEED_SERIAL_PROTOCOL
```

```
ITC_INVERSE_SERIAL_UNKNOW
```

未知

```
ITC_INVERSE_PROTOCOL_MODE1
```

不抓拍

```
ITC_INVERSE_PROTOCOL_MODE2
```

双/三线圈，离开 1，离开 2，离开 2 延时抓拍，离开 1 牌识，上传车辆图片（卡口离开 1，超速离开 1、离开 2、离开 2 延时）和车牌结果

```
ITC_INVERSE_PROTOCOL_MODE3
```

双/三线圈，离开 1，离开 2，离开 2 延时抓拍，离开 1 牌识，上传车辆图片（卡口离开 1，超速离开 1、离开 2）和车牌结果

*byRes*

保留

## 7.223 NET\_ITC\_SINGLE\_IO\_LIGHT\_PARAM:单个 IO 接入信号灯参数

```
struct{
```

```
    BYTE    byLightType;
```

```
    BYTE    byRelatedIO;
```

```
    BYTE    byRedLightState;
```

```
    BYTE    byRes[17];
```

```
}NET_ITC_SINGLE_IO_LIGHT_PARAM,*LPNET_ITC_SINGLE_IO_LIGHT_PARAM;
```

### Members

*byLightType*

交通灯导向类型：0- 左转灯，1- 直行灯，2- 右转灯

*byRelatedIO*

关联的 IO 口号，1~6

*byRedLightState*

红灯电平状态：0- 低电平红灯，1- 高电平红灯

*byRes*

保留

## 7.224 NET\_ITC\_SINGLE\_IOSPEED\_PARAM:单组 IO 测速参数

```
struct{
```

```
    BYTE    byEnable;
```

```
    BYTE    byTrigCoil1;
```

```
    BYTE    byCoil1IOStatus;
```

```
    BYTE    byTrigCoil2;
```

```
    BYTE    byCoil2IOStatus;
```

```
    BYTE    byRelatedDriveWay;
```

```
    BYTE    byTimeOut;
```

```
    BYTE    byRelatedIOOutEx;
```

```

DWORD          dwDistance;
BYTE           byCapSpeed;
BYTE           bySpeedLimit;
BYTE           bySpeedCapEn;
BYTE           bySnapTimes1;
BYTE           bySnapTimes2;
BYTE           byBigCarSpeedLimit;
BYTE           byBigCarSignSpeed;
BYTE           byIntervalType;
WORD           wInterval1[MAX_INTERVAL_NUM];
WORD           wInterval2[MAX_INTERVAL_NUM];
BYTE           byRelatedIOOut[MAX_IOOUT_NUM];
BYTE           byFlashMode;
BYTE           byLaneType;
BYTE           byCarSignSpeed;
BYTE           byUseageType;
NET\_ITC\_PLATE\_RECOG\_REGION\_PARAM struPlateRecog[MAX_LANEAREA_NUM];
BYTE           byRelaLaneDirectionType;
BYTE           byLowSpeedLimit;
BYTE           byBigCarLowSpeedLimit;
BYTE           byLowSpeedCapEn;
BYTE           byEmergencyCapEn;
BYTE           byRes[27];
}NET_ITC_SINGLE_IOSPEED_PARAM,*LPNET_ITC_SINGLE_IOSPEED_PARAM;

```

### Members

#### *byEnable*

是否启用：0- 不启用，1- 启用

#### *byTrigCoil1*

第一线圈关联 IO：0- IO1，1- IO2，2- IO3，3- IO4，4- IO5，5- IO6

#### *byCoil1IOStatus*

第一线圈 IO 输入口状态：0- 下降沿（默认），1- 上升沿，2- 上升沿和下降沿，3- 高电平，4- 低电平

#### *byTrigCoil2*

第二线圈关联 IO：0- IO1，1- IO2，2- IO3，3- IO4，4- IO5，5- IO6

#### *byCoil2IOStatus*

第二线圈 IO 输入口状态：0- 下降沿（默认），1- 上升沿，2- 上升沿和下降沿，3- 高电平，4- 低电平

#### *byRelatedDriveWay*

关联的车道号

#### *byTimeOut*

超时时间（默认 10），单位：s

#### *byRelatedIOOutEx*

关联的 IO 输出口，按位表示输出口号：bit0 表示 IO 输出口 1，bit1 表示 IO 输出口 2，以此类推。  
值：0-不关联，1-关联，最多支持关联到 8 个 IO 输出口(兼容 byRelatedIOOut 字段)

***dwDistance***

线圈距离 (默认 1000), 单位: 厘米

***byCapSpeed***

起拍速度 (默认 30), 单位: km/h

***bySpeedLimit***

限速值 (默认 60), 单位: km/h

***bySpeedCapEn***

是否启用超速抓拍: 0- 否, 1- 是

***bySnapTimes1***

线圈 1 抓拍次数 (默认不抓拍): 0- 不抓拍, 非 0- 连拍次数, 最大 5 次

***bySnapTimes2***

线圈 2 抓拍次数 (默认 1), 0- 不抓拍, 非 0- 连拍次数, 最大 5 次

***byBigCarSpeedLimit***

大车车速限制值

***byBigCarSignSpeed***

大车标志限速, 单位: km/h

***byIntervalType***

间隔类型 (默认按时间), 0- 时间起效, 1- 距离起效

***wInterval1***

线圈 1 连拍间隔时间 (单位: ms) 或连拍间隔距离 (单位: 分米)。当 *byIntervalType* 为 0 时, 表示间隔时间, 当 *byIntervalType* 为 1 时, 表示距离

***wInterval2***

线圈 2 连拍间隔时间 (单位: ms) 或连拍间隔距离 (单位: 分米)。当 *byIntervalType* 为 0 时, 表示间隔时间, 当 *byIntervalType* 为 1 时, 表示距离

***byRelatedIOOut***

关联的 IO 输出口(可以同时关联多个), 数组 0 表示 IO 输出口 1, 数组 1 表示 IO 输出口 2, 以此类推, 其参数值: 0- 不关联, 1- 关联

***byFlashMode***

闪光灯闪烁模式: 0- 同时闪, 1- 轮流闪

***byLaneType***

车道类型: 0- 未配置, 1- 高速公路, 2- 城市快速路, 0xff-其他道路

***byCarSignSpeed***

小车标志限速, 单位: km/h

***byUseageType***

车道用途类型, 定义详见 [ITC\\_LANE\\_USEAGE\\_TYPE](#)

***struPlateRecog***

牌识参数(可用牌识区域 1 个, 保留一个)

***byRelaLaneDirectionType***

关联车道方向类型, 定义详见 [ITC\\_RELA\\_LANE\\_DIRECTION\\_TYPE](#)

***byLowSpeedLimit***

小车限低速值, 单位: km/h

***byBigCarLowSpeedLimit***

大车限低速值, 单位: km/h

***byLowSpeedCapEn***



是否启用低速抓拍：0- 否，1- 是

*byEmergencyCapEn*

是否启用应急车道抓拍：0- 否，1- 是

*byRes*

保留

## 7.225 NET\_ITC\_SINGLE\_IOTL\_PARAM:单组 IO 红绿灯参数

```

struct{
    BYTE                byEnable;
    BYTE                byLightIO;
    BYTE                byTrafficLight;
    BYTE                byTrigIO;
    BYTE                byTrigIOStatus;
    BYTE                byRelatedDriveWay;
    BYTE                byRecordEnable;
    BYTE                byRecordType;
    BYTE                byPreRecordTime;
    BYTE                byRecordDelayTime;
    BYTE                byRecordTimeOut;
    BYTE                byRedSnapTimes;
    BYTE                byGreenSnapTimes;
    BYTE                byRelatedIOOutEx;
    BYTE                byRes1;
    BYTE                byIntervalType;
    WORD                wRedInterval[MAX_INTERVAL_NUM];
    WORD                wGreenInterval[MAX_INTERVAL_NUM];
    BYTE                byRelatedIOOut[MAX_IOOUT_NUM];
    BYTE                byFlashMode;
    BYTE                byRes2[3];
    NET_ITC_PLATE_RECOG_REGION_PARAM struPlateRecog[MAX_LANEAREA_NUM];
    BYTE                byRes[32];
}NET_ITC_SINGLE_IOTL_PARAM,*LPNET_ITC_SINGLE_IOTL_PARAM;

```

### Members

*byEnable*

是否启用：0- 不启用，1- 启用

*byLightIO*

红绿灯 IO：0- IO1，1- IO2，2- IO3，3- IO4，4- IO5，5- IO6

*byTrafficLight*

红绿灯有效状态：0- 高电平红灯，低电平绿灯；1- 高电平绿灯，低电平红灯

*byTrigIO*

触发的 IO 号：0- IO1，1- IO2，2- IO3，3- IO4，4- IO5，5- IO6

*byTrigIOStatus*

触发 IO 口的状态（默认 0）：0- 下降沿，1- 上升沿，2- 上升沿和下降沿，3- 高电平，4- 低电平

**byRelatedDriveWay**

关联的车道号

**byRecordEnable**

闯红灯周期录像标志：0- 不录像，1- 录像

**byRecordType**

闯红灯录像类型：0- 预录，1- 延时录像

**byPreRecordTime**

闯红灯录像片段预录时间（默认 0），单位：秒

**byRecordDelayTime**

闯红灯录像片段延时时间（默认 0），单位：秒

**byRecordTimeOut**

闯红灯周期录像超时时间，单位：秒

**byRedSnapTimes**

红灯抓拍次数：0- 不抓拍；非 0- 连拍次数，最大 5 次

**byGreenSnapTimes**

绿灯抓拍次数：0- 不抓拍；非 0- 连拍次数，最大 5 次

**byRelatedIOOutEx**

关联的 IO 输出口，按位表示输出口号：bit0 表示 IO 输出口 1，bit1 表示 IO 输出口 2，以此类推。  
值：0-不关联，1-关联，最多支持关联到 8 个 IO 输出口(兼容 byRelatedIOOut 字段)

**byRes1**

保留

**byIntervalType**

间隔类型（默认按时间）：0- 时间起效，1- 距离起效

**wRedInterval**

红灯连拍间隔时间（单位：ms）或连拍间隔距离（单位：分米）。当 byIntervalType 为 0 时，表示间隔时间，当 byIntervalType 为 1 时，表示距离高

**wGreenInterval**

绿灯连拍间隔时间（单位：ms）或连拍间隔距离（单位：分米）。当 byIntervalType 为 0 时，表示间隔时间，当 byIntervalType 为 1 时，表示距离

**byRelatedIOOut**

关联的 IO 输出口，可以同时关联多个

**byFlashMode**

闪光灯闪烁模式：0- 同时闪，1- 轮流闪

**byRes2**

保留

**struPlateRecog**

牌识区域参数

**byRes**

保留

## 7.226 NET\_ITC\_SINGLE\_RS485\_LIGHT\_PARAM: 单个 485 接入信号灯参数

```
struct{
    BYTE    byLightType;
    BYTE    byRelatedLightChan;
    BYTE    byInputLight;
    BYTE    byRelatedYLightChan;
    BYTE    byRes[16];
}NET_ITC_SINGLE_RS485_LIGHT_PARAM, *LPNET_ITC_SINGLE_RS485_LIGHT_PARAM;
```

### Members

#### *byLightType*

交通灯导向类型：0- 左转灯，1- 直行灯，2- 右转灯

#### *byRelatedLightChan*

红灯关联的红绿灯检测器通道号，0~16，0 表示无（即没有该种类型的灯）

#### *byInputLight*

接入的信号灯类型：0- 接红灯，1- 接绿灯，智能交通摄像机不使用

#### *byRelatedYLightChan*

黄灯关联的红绿灯检测器通道号，0~16，0 表示无（即没有该种类型的灯）

#### *byRes*

保留，置为 0

### Remarks

如路口信号灯有两个：左转+直行，右转。需要配置三个数组表示，第一个数组配左转灯，关联检测器通道 1；第二组配直行，关联通道 1；第三组配右转，关联通道 2。

## 7.227 NET\_ITC\_SINGLE\_TRIGGERCFG: 单个触发参数

```
struct{
    BYTE    byEnable;
    BYTE    byRes1[3];
    DWORD   dwTriggerType;
    NET\_ITC\_TRIGGER\_PARAM\_UNION uTriggerParam;
    BYTE    byRes[64];
}NET_ITC_SINGLE_TRIGGERCFG, *LPNET_ITC_SINGLE_TRIGGERCFG;
```

### Members

#### *byEnable*

启用状态：0- 不启用，1- 启用

#### *byRes1*

保留

#### *dwTriggerType*

触发类型，详见 [ITC\\_TRIGGERMODE\\_TYPE](#)

#### *uTriggerParam*

触发参数

*byRes*

保留

## 7.228 NET\_ITC\_SINGLE\_VIDEO\_DETECT\_LIGHT\_PARAM:单组视频检测

### 交通信号灯参数

```
struct{
    BYTE                byLightNum;
    BYTE                byStraightLight;
    BYTE                byLeftLight;
    BYTE                byRightLight;
    BYTE                byRedLight;
    BYTE                byGreenLight;
    BYTE                byYellowLight;
    BYTE                byYellowLightTime;
    NET\_POS\_PARAM    struLightRect;
    BYTE                byRes[24];
}NET_ITC_SINGLE_VIDEO_DETECT_LIGHT_PARAM, *LPNET_ITC_SINGLE_VIDEO_DETECT_LIGHT_PARAM;
```

#### Members

*byLightNum*

交通灯个数

*byStraightLight*

是否有直行标志灯：0- 否，1- 是

*byLeftLight*

是否有左转标志灯：0- 否，1- 是

*byRightLight*

是否有右转标志灯：0- 否，1- 是

*byRedLight*

是否有红灯：0- 否，1- 是

*byGreenLight*

是否有绿灯：0- 否，1- 是

*byYellowLight*

是否有黄灯：0- 否，1- 是

*byYellowLightTime*

黄灯持续时间（用于修正红绿灯识别偏差），取值范围：0~10，单位：s

*struLightRect*

交通灯区域

*byRes*

保留，置为 0

## 7.229 NET\_ITC\_SINGLEIO\_PARAM:单 IO 参数

```

struct{
    BYTE                byDefaultStatus;
    BYTE                byRelatedDriveWay;
    BYTE                bySnapTimes;
    BYTE                byRelatedIOOutEx;
    NET\_ITC\_INTERVAL\_PARAM    struInterval;
    BYTE                byRelatedIOOut[MAX_IOOUT_NUM];
    BYTE                byFlashMode;
    BYTE                byEnable;
    BYTE                byUseageType;
    BYTE                byEmergencyCapEn;
    NET\_ITC\_PLATE\_RECOG\_REGION\_PARAM    struPlateRecog[MAX_LANEAREA_NUM];
    BYTE                byRes[32];
}NET_ITC_SINGLEIO_PARAM,*LPNET_ITC_SINGLEIO_PARAM;

```

### Members

#### *byDefaultStatus*

IO 触发默认状态：0- 低电平，1- 高电平

#### *byRelatedDriveWay*

关联的车道号

#### *bySnapTimes*

抓拍次数（默认 1）：0- 不抓拍；非 0- 连拍次数，最大 5

#### *byRelatedIOOutEx*

关联的 IO 输出口，按位表示输出口号：bit0 表示 IO 输出口 1，bit1 表示 IO 输出口 2，以此类推。  
值：0-不关联，1-关联，最多支持关联到 8 个 IO 输出口(兼容 *byRelatedIOOut* 字段)

#### *struInterval*

抓拍间隔参数

#### *byRelatedIOOut*

关联的 IO 输出口，可以同时关联多个

#### *byFlashMode*

闪光灯闪烁模式：0- 同时闪，1- 轮流闪

#### *byEnable*

单 IO 使能标志：0- 不启用，1- 启用

#### *byUseageType*

车道用途类型，定义详见 [ITC\\_LANE\\_USEAGE\\_TYPE](#)

#### *byEmergencyCapEn*

是否启用应急车道抓拍：0- 否，1- 是

#### *struPlateRecog*

牌识区域参数

#### *byRes*

保留

## 7.230 NET\_ITC\_STATUS\_DETECT\_RESULT:实时状态检测结果

```
struct{
    ITC_STATUS_DETECT_TYPE      dwStatusType;
    NET_ITC_STATUS_UNION        uStatusParam;
    DWORD                       dwHoldTime;
    BYTE                        byRes[32];
}NET_ITC_STATUS_DETECT_RESULT,*LPNET_ITC_STATUS_DETECT_RESULT;
```

### Members

#### *dwStatusType*

状态检测类型，定义如下所示：

```
enum _ITC_STATUS_DETECT_TYPE_{
    ITC_STATUS_DETECT_NULL = 0,
    ITC_STATUS_DETECT_TRIGGERIO,
    ITC_STATUS_DETECT_FLASHOUT,
    ITC_STATUS_DETECT_RS485,
    ITC_STATUS_DETECT_TRIGGERTYPE,
    ITC_STATUS_DETECT_TRAFFICLIGHT_COLOR,
    ITC_STATUS_DETECT_TRAFFICLIGHT_TURN
}ITC_STATUS_DETECT_TYPE
```

*ITC\_STATUS\_DETECT\_NULL*

未知

*ITC\_STATUS\_DETECT\_TRIGGERIO*

触发 IO

*ITC\_STATUS\_DETECT\_FLASHOUT*

同步输出

*ITC\_STATUS\_DETECT\_RS485*

RS485

*ITC\_STATUS\_DETECT\_TRIGGERTYPE*

触发类型（保留）

*ITC\_STATUS\_DETECT\_TRAFFICLIGHT\_COLOR*

交通灯（区分颜色）

*ITC\_STATUS\_DETECT\_TRAFFICLIGHT\_TURN*

交通灯（区分转向）

#### *uStatusParam*

状态检测结果

#### *dwHoldTime*

灯亮持续时间（保留），单位：ms

#### *byRes*

保留

## 7.231 NET\_ITC\_STATUS\_UNION:状态检测结果联合体

```
union{
    BYTE uLen[48];
    NET_ITC_TRIGGERIO_INFO struTrigIO;
    NET_ITC_FLASHOUT_INFO struFlashOut;
    NET_ITC_SERIAL_CHECKINFO struSerial;
    NET_ITC_TRIGGERTYPE_INFO struTrigType;
    NET_ITC_TRAFFIC_LIGHT_COLOR struTrafficLightColor;
    NET_ITC_TRAFFIC_LIGHT_TURN struTrafficLightTurn;
}NET_ITC_STATUS_UNION,*LPNET_ITC_STATUS_UNION;
```

### Members

*uLen*

联合体的最大长度

*struTrigIO*

触发 IO 口号

*struFlashOut*

同步输出

*struSerial*

串口

*struTrigType*

触发类型（保留）

*struTrafficLightColor*

交通灯（区分颜色）

*struTrafficLightTurn*

交通灯（区分转向）

## 7.232 NET\_ITC\_TRAFFIC\_LIGHT\_COLOR:视频电警交通灯颜色参数

```
struct{
    BYTE byLeftLight;
    BYTE byRightLight;
    BYTE byStraightLight;
    BYTE byRes[45];
}NET_ITC_TRAFFIC_LIGHT_COLOR,*LPNET_ITC_TRAFFIC_LIGHT_COLOR;
```

### Members

*byLeftLight*

左转向灯：0- 无效，1- 红灯，2- 绿灯 3- 黄灯，4- 灯都不亮

*byRightLight*

右转向灯：0- 无效，1- 红灯，2- 绿灯，3- 黄灯，4- 灯都不亮

*byStraightLight*

直行灯：0- 无效，1- 红灯，2- 绿灯，3- 黄灯，4- 灯都不亮

*byRes*

保留，置为 0

### 7.233 NET\_ITC\_TRAFFIC\_LIGHT\_PARAM:交通信号灯参数

```
struct{
    BYTE                bySource;
    BYTE                byRes1[3];
    NET\_ITC\_LIGHT\_ACCESSPARAM\_UNION    struLightAccess;
    BYTE                byRes[32];
}NET_ITC_TRAFFIC_LIGHT_PARAM, *LPNET_ITC_TRAFFIC_LIGHT_PARAM;
```

#### Members

##### *bySource*

交通信号灯接入源：0- IO 接入，1- RS485 接入，2- 视频检测

##### *byRes1*

保留，置为 0

##### *struLightAccess*

信号灯接入参数

##### *byRes*

保留，置为 0

### 7.234 NET\_ITC\_TRAFFIC\_LIGHT\_TURN:交通灯转向参数

```
struct{
    BYTE    byLightType[MAX_ITC_LANE_NUM];
    BYTE    byRes[42];
}NET_ITC_TRAFFIC_LIGHT_TURN,*LPNET_ITC_TRAFFIC_LIGHT_TURN;
```

#### Members

##### *byLightType*

每一位数组对应一个车道，数组的值：0- 无效，1- 红灯，2- 绿灯，3- 黄灯，4- 灯都不亮

##### *byRes*

保留

### 7.235 NET\_ITC\_TRIGGERCFG:触发参数

```
struct{
    DWORD                dwSize;
    NET\_ITC\_SINGLE\_TRIGGERCFG    struTriggerParam;
    BYTE                byRes[32];
}NET_ITC_TRIGGERCFG,*LPNET_ITC_TRIGGERCFG;
```

#### Members

##### *dwSize*

结构体大小



**struTriggerParam**

单个触发参数

**byRes**

保留

**7.236 NET\_ITC\_TRIGGERIO\_INFO:触发 IO 口参数**

```

struct{
    BYTE    byTriggerIOIndex[MAX_JOIN_NUM];
    BYTE    byRes[40];
}NET_ITC_TRIGGERIO_INFO,*LPNET_ITC_TRIGGERIO_INFO;

```

**Members****byTriggerIOIndex**

触发 IO 号

**byRes**

保留

**7.237 NET\_ITC\_TRIGGERMODE\_ABILITY:触发模式能力**

```

struct{
    DWORD    dwSize;
    DWORD    dwTriggerType;
    BYTE     byRes[16];
}NET_ITC_TRIGGERMODE_ABILITY,*LPNET_ITC_TRIGGERMODE_ABILITY;

```

**Members****dwSize**

大小

**dwTriggerType**

触发类型，按位表示，定义如下所示：

```

enum _ITC_TRIGGERMODE_TYPE_{
    ITC_POST_IOSPEED_TYPE           = 0x1,
    ITC_POST_SINGLEIO_TYPE          = 0x2,
    ITC_POST_RS485_TYPE              = 0x4,
    ITC_POST_RS485_RADAR_TYPE        = 0x8,
    ITC_POST_VIRTUALCOIL_TYPE        = 0x10,
    ITC_EPOLICE_IO_TRAFFICLIGHTS_TYPE = 0x100,
    ITC_EPOLICE_RS485_TYPE           = 0x200,
    ITC_PE_RS485_TYPE                = 0x10000,
    ITC_VIDEO_EPOLICE_TYPE           = 0x20000
}ITC_TRIGGERMODE_TYPE

```

*ITC\_POST\_IOSPEED\_TYPE*

IO 测速（卡口）

*ITC\_POST\_SINGLEIO\_TYPE*

单 IO 触发（卡口）  
*ITC\_POST\_RS485\_TYPE*  
 RS485 车检器触发（卡口）  
*ITC\_POST\_RS485\_RADAR\_TYPE*  
 RS485 雷达触发（卡口）  
*ITC\_POST\_VIRTUALCOIL\_TYPE*  
 虚拟线圈触发（卡口）  
*ITC\_EPOLICE\_IO\_TRAFFICLIGHTS\_TYPE*  
 IO 红绿灯（电警）  
*ITC\_EPOLICE\_RS485\_TYPE*  
 RS485 车检器电警触发（电警）  
*ITC\_PE\_RS485\_TYPE*  
 RS485 车检器卡式电警触发（卡式电警）  
*ITC\_VIDEO\_EPOLICE\_TYPE*  
 视频电警触发（卡式电警）

*byRes*

保留

## 7.238 **NET\_ITC\_TRIGGERTYPE\_INFO**:触发类型参数

```
struct{
    BYTE    byTrigTypeIndex;
    BYTE    byRes[47];
}NET_ITC_TRIGGERTYPE_INFO,*LPNET_ITC_TRIGGERTYPE_INFO;
```

### Members

*byTrigTypeIndex*

触发类型值

*byRes*

保留

## 7.239 **NET\_ITC\_TRIGGER\_PARAM\_UNION**:触发参数联合体

```
union{
    DWORD    uLen[1070];
    NET\_ITC\_POST\_IOSPEED\_PARAM    struIOSpeed;
    NET\_ITC\_POST\_SINGLEIO\_PARAM    struSingleIO;
    NET\_ITC\_POST\_RS485\_PARAM        struPostRs485;
    NET\_ITC\_POST\_RS485\_RADAR\_PARAM    struPostRadar;
    NET\_ITC\_POST\_VTCOIL\_PARAM        struVtCoil;
    NET\_ITC\_POST\_HVT\_PARAM\_V50        struHvtV50;
    NET\_ITC\_POST\_MPR\_PARAM            struPostMpr;
    NET\_ITC\_POST\_PRS\_PARAM            struPostPrs;
    NET\_ITC\_EPOLICE\_IOTL\_PARAM        struIOTL;
```

<a href="#">NET_ITC_EPOLICE_RS485_PARAM</a>	struEpoliceRs485;
<a href="#">NET_ITC_EPOLICE_RS485_PARAM</a>	struPERs485;
<a href="#">NET_ITC_POST_IMT_PARAM</a>	struPostImt;
<a href="#">NET_IPC_POST_HVT_PARAM</a>	strulpcHvt;
<a href="#">NET_ITC_POST_MOBILE_PARAM</a>	struPostMobile;
<a href="#">NET_ITC_REDLIGHT_PEDESTRIAN_PARAM</a>	struRedLightPed;
<a href="#">NET_ITC_NOCOMITY_PEDESTRIAN_PARAM</a>	struNoComityPed;

}NET\_ITC\_TRIGGER\_PARAM\_UNION,\*LPNET\_ITC\_TRIGGER\_PARAM\_UNION;

## Members

### *uLen*

联合体长度，1070\*4 共 4280 字节

### *struIOSpeed*

（卡口）IO 测速参数，触发类型为 ITC\_POST\_IOSPEED\_TYPE 时有效

### *struSingleIO*

（卡口）单 IO 参数，触发类型为 ITC\_POST\_SINGLEIO\_TYPE 时有效

### *struPostRs485*

（卡口）RS485 车检器参数，触发类型为 ITC\_POST\_RS485\_TYPE 时有效

### *struPostRadar*

（卡口）RS485 雷达参数，触发类型为 ITC\_POST\_RS485\_RADAR\_TYPE 时有效

### *struVtCoil*

（卡口）虚拟线圈参数，触发类型为 ITC\_POST\_VIRTUALCOIL\_TYPE 时有效

### *struHvtV50*

（卡口）混行卡口视频触发 V50 参数，触发类型为 ITC\_POST\_HVT\_TYPE\_V50 时有效

### *struPostMpr*

（卡口）多帧检测触发参数（MPR），触发类型为 ITC\_POST\_MPR\_TYPE 时有效

### *struPostPrs*

（PRS）视频检测触发，触发类型为 ITC\_POST\_PRS\_TYPE 时有效

### *struIOTL*

（电警）IO 红绿灯参数，触发类型为 ITC\_EPOLICE\_IO\_TRAFFICLIGHTS\_TYPE 时有效

### *struEpoliceRs485*

（电警）RS485 车检器触发参数，触发类型为 ITC\_EPOLICE\_RS485\_TYPE 时有效

### *struPERs485*

（卡式电警）RS485 车检器触发参数，触发类型为 ITC\_PE\_RS485\_TYPE 时有效

### *struPostImt*

智慧监控配置参数，触发类型为 ITC\_POST\_IMT\_TYPE 时有效

### *strulpcHvt*

（IPC）混行卡口参数，触发类型为 IPC\_POST\_HVT\_TYPE 时有效

### *struPostMobile*

移动交通触发模式，触发类型为 ITC\_POST\_MOBILE\_TYPE 时有效

### *struRedLightPed*

行人闯红灯触发模式，触发类型为 ITC\_REDLIGHT\_PEDESTRIAN\_TYPE 时有效

### *struNoComityPed*

不礼让行人触发参数，触发类型为 ITC\_NOCOMITY\_PEDESTRIAN\_TYPE 时有效

## 7.240 **NET\_ITC\_VIDEO\_DETECT\_LIGHT\_PARAM**:视频检测交通信号灯参数

```
struct{
    NET\_ITC\_SINGLE\_VIDEO\_DETECT\_LIGHT\_PARAM struTrafficLight[MAX_VIDEO_DETECT_LIGHT_NUM];
    BYTE byRes[8];
}NET_ITC_VIDEO_DETECT_LIGHT_PARAM, *LPNET_ITC_VIDEO_DETECT_LIGHT_PARAM;
```

### Members

*struRS485Light*  
单个视频检测信号灯参数

*byRes*  
保留，置为 0

## 7.241 **NET\_ITC\_VIDEO\_EPOLICE\_PARAM**:视频电警触发参数

```
struct{
    BYTE byEnable;
    BYTE byLaneNum;
    BYTE byLogicJudge;
    BYTE byRes1;
    NET\_ITC\_PLATE\_RECOG\_PARAM struPlateRecog;
    NET\_ITC\_TRAFFIC\_LIGHT\_PARAM struTrafficLight;
    NET\_ITC\_LANE\_VIDEO\_EPOLICE\_PARAM struLaneParam[MAX_ITC_LANE_NUM];
    NET\_ITC\_LINE struLaneBoundaryLine;
    NET\_ITC\_LINE struLeftLine;
    NET\_ITC\_LINE struRightLine;
    NET\_ITC\_LINE struTopZebraLine;
    NET\_ITC\_LINE struBotZebraLine;
    BYTE byRes[32];
}NET_ITC_VIDEO_EPOLICE_PARAM, *LPNET_ITC_VIDEO_EPOLICE_PARAM;
```

### Members

*byEnable*  
是否启用：0- 不启用，1- 启用

*byLaneNum*  
识别的车道个数，1~6

*byLogicJudge*  
闯红灯违规判断逻辑：0- 按方向，1- 按车道

*byRes1*  
保留

*struPlateRecog*  
牌识参数

*struTrafficLight*

交通信号灯参数

*struLaneParam*

单车道参数

*struLaneBoundaryLine*

车道边界线（最右边车道的边界线）

*struLeftLine*

左转弯分界线

*struRightLine*

右转弯分界线

*struTopZebraLine*

上部斑马线

*struBotZebraLine*

下部斑马线

*byRes*

保留

## 7.242 NET\_ITC\_VIDEO\_TRIGGER\_COND:视频触发条件

```
struct{
    DWORD    dwSize;
    DWORD    dwChannel;
    DWORD    dwTriggerMode;
    BYTE     byRes[16];
}NET_ITC_VIDEO_TRIGGER_COND, *LPNET_ITC_VIDEO_TRIGGER_COND;
```

### Members

*dwSize*

大小

*dwChannel*

通道号

*dwTriggerMode*

视频触发模式类型，详见 [ITC\\_TRIGGERMODE\\_TYPE](#)

*byRes*

保留，置为 0

### Remarks

*dwTriggerMode* 为 ITC\_VIDEO\_EPOLICE\_TYPE 时有效。

## 7.243 NET\_ITC\_VIDEO\_TRIGGER\_PARAM:视频触发参数

```
struct{
    DWORD                                dwSize;
    DWORD                                dwMode;
    NET\_ITC\_VIDEO\_TRIGGER\_PARAM\_UNION  uVideoTrigger;
    BYTE                                  byRes[32];
}
```

```
}NET_ITC_VIDEO_TRIGGER_PARAM, *LPNET_ITC_VIDEO_TRIGGER_PARAM;
```

### Members

*dwSize*

大小

*dwMode*

触发模式，详见 [ITC\\_TRIGGERMODE\\_TYPE](#)

*uVideoTrigger*

触发模式参数

*byRes*

保留，置为 0

## 7.244 NET\_ITC\_VIDEO\_TRIGGER\_PARAM\_UNION: 视频电警触发参数联合体

```
union {
    DWORD uLen[1150];
    NET_ITC_VIDEO_EPOLICE_PARAM struVideoEP;
}NET_ITC_VIDEO_TRIGGER_PARAM_UNION, *LPNET_ITC_VIDEO_TRIGGER_PARAM_UNION;
```

### Members

*uLen*

联合体大小

*struVideoEP*

视频电警参数

## 7.245 NET\_ITC\_VIOLATION\_DETECT\_LINE: 违规检测线参数

```
struct{
    NET_ITC_LINE struLaneLine;
    NET_ITC_LINE struStopLine;
    NET_ITC_LINE struRedLightLine;
    NET_ITC_LINE struCancelLine;
    NET_ITC_LINE struWaitLine;
    NET_ITC_LINE struRes[8];
}NET_ITC_VIOLATION_DETECT_LINE, *LPNET_ITC_VIOLATION_DETECT_LINE;
```

### Members

*struLaneLine*

车道线参数

*struStopLine*

停止线参数

*struRedLightLine*

闯红灯触发线参数

*struCancelLine*

直行触发位置取消线

*struWaitLine*

待行区停止线参数

*struRes*

保留，置为 0

### See Also

闯红灯触发线是车辆在红灯状态下行驶过该线则判断为闯红灯，一般图像中闯红灯触发线在直行触发线的下方。

直行触发位置取消线是当检测到车辆经过该线后就判定为直行，不会再做左右转的判断，用于车辆不按车道行驶的违章判断。

## 7.246 NET\_ITC\_VIOLATION\_DETECT\_PARAM: 违规检测参数

```
struct{
    DWORD    dwVioDetectType;
    BYTE     byDriveLineSnapTimes;
    BYTE     byReverseSnapTimes;
    WORD     wStayTime;
    BYTE     byNonDriveSnapTimes;
    BYTE     byChangeLaneTimes;
    BYTE     bybanTimes;
    BYTE     byDriveLineSnapSen;
    WORD     wSnapPosFixPixel;
    BYTE     bySpeedTimes;
    BYTE     byTurnAroundEnable;
    BYTE     byThirdPlateRecogTime;
    BYTE     byPostSnapTimes;
    BYTE     byRes1[18];
    WORD     wStopLineDis;
    BYTE     byRes[14];
}NET_ITC_VIOLATION_DETECT_PARAM, *LPNET_ITC_VIOLATION_DETECT_PARAM
```

### Members

*dwVioDetectType*

违规检测类型，按位表示，其值：0- 不启用，1- 启用，类型定义详见 [ITC\\_VIOLATION\\_DETECT\\_TYPE](#)

*byDriveLineSnapTimes*

压车道线抓拍张数，2~3

*byReverseSnapTimes*

逆行抓拍张数，2~3

*wStayTime*

机占非停留时间（该时间后抓拍），单位：s

*byNonDriveSnapTimes*

机占非抓拍张数，2~3

*byChangeLaneTimes*

违法变道抓拍张数，2~3

**bybanTimes**

违法禁令抓拍张数，2~3

**byDriveLineSnapSen**

压线灵敏度，取值范围：0~100

**wSnapPosFixPixel**

第 2、3 张抓拍位置最小偏移(违反信号灯时起效)，单位：像素

**bySpeedTimes**

违法超速抓拍张数，2~3

**byTurnAroundEnable**

违章掉头使能：0- 关闭，1- 开启

**byThirdPlateRecogTime**

第三张牌识时间，取值范围：0~180，单位：s

**byPostSnapTimes**

卡口抓拍张数，1~2 张

**byRes1**

保留，置为 0

**wStopLineDis**

电警第 2 张违规图片与停止线的最短距离，取值范围：[0,300]，单位：像素

**byRes**

保留，置为 0

## 7.247 NET\_ITC\_VTCOIL\_INFO:虚拟线圈

struct{

```

NET_VCA_RECT          struLaneRect;
BYTE                  byTrigFlag;
BYTE                  byTrigSensitive;
BYTE                  byRelatedIOOut[MAX_IOOUT_NUM];
BYTE                  byFlashMode;
BYTE                  byLaneType;
BYTE                  byEnableRadar;
NET_ITC_VTLANE_PARAM struLane;
BYTE                  byUseageType;
BYTE                  byCarDriveDirect;
BYTE                  byRes[30];

```

}NET\_ITC\_VTCOIL\_INFO,\*LPNET\_ITC\_VTCOIL\_INFO;

### Members

**struLaneRect**

虚拟线圈区域

**byTrigFlag**

触发标志：0- 车头触发，1- 车尾触发，2- 车头/车尾都触发

**byTrigSensitive**

触发灵敏度：1~100

**byRelatedIOOut**



关联的 IO 输出口(可以同时关联多个), 数组 0 表示 IO 输出口 1, 数组 1 表示 IO 输出口 2, 以此类推, 0-不关联, 1-关联

#### *byFlashMode*

闪光灯闪烁模式: 0- 同时闪, 1- 轮流闪

#### *byLaneType*

车道类型: 0- 未配置, 1- 高速公路, 2- 城市快速路, 0xff- 其他道路

#### *byEnableRadar*

是否启用雷达测速: 0- 否, 1- 是

#### *struLane*

关联的车道参数

#### *byUseageType*

车道用途类型, 定义详见 [ITC\\_LANE\\_USEAGE\\_TYPE](#)

#### *byCarDriveDirect*

车辆行驶方向, 具体定义如下:

```
enum _ITC_LANE_CAR_DRIVE_DIRECT_{
    ITC_LANE_DRIVE_UNKNOW      = 0,
    ITC_LANE_DRIVE_UP_TO_DOWN  = 1,
    ITC_LANE_DRIVE_DOWN_TO_UP  = 2
}ITC_LANE_CAR_DRIVE_DIRECT
```

*ITC\_LANE\_DRIVE\_UNKNOW*

未知

*ITC\_LANE\_DRIVE\_UP\_TO\_DOWN*

从上往下行驶(下行)

*ITC\_LANE\_DRIVE\_DOWN\_TO\_UP*

从下往上行驶(上行)

#### *byRes*

保留

## 7.248 **NET\_ITC\_VTLANE\_PARAM:**车道参数

```
struct{
    BYTE                byRelatedDriveWay;
    BYTE                bySpeedCapEn;
    BYTE                bySignSpeed;
    BYTE                bySpeedLimit;
    BYTE                bySnapTimes;
    BYTE                byBigCarSignSpeed;
    BYTE                byBigCarSpeedLimit;
    BYTE                byRelatedIOOutEx;
    NET\_ITC\_INTERVAL\_PARAM struInterval;
    BYTE                byRelatedIOOut[MAX_IOOUT_NUM];
    BYTE                byFlashMode;
    BYTE                byLowSpeedLimit;
    BYTE                byBigCarLowSpeedLimit;
```

```

BYTE byRelaLaneDirectionType;
NET_ITC_PLATE_RECOG_REGION_PARAM struPlateRecog[MAX_LANEAREA_NUM];
NET_VCA_LINE struLine;
}NET_ITC_VTLANE_PARAM,*LPNET_ITC_VTLANE_PARAM;

```

### Members

*byRelatedDriveWay*

关联的车道号

*bySpeedCapEn*

是否启用超速抓拍: 0- 否, 1- 是

*bySignSpeed*

标志限速, 单位: km/h

*bySpeedLimit*

限速值, 单位: km/h

*bySnapTimes*

抓拍次数 (默认 1): 0- 不抓拍; 非 0- 连拍次数, 最大 5

*byBigCarSignSpeed*

大车标志限速, 单位 km/h

*byBigCarSpeedLimit*

大车限速值, 单位 km/h

*byRelatedIOOutEx*

关联的 IO 输出口, 按位表示输出口号: bit0 表示 IO 输出口 1, bit1 表示 IO 输出口 2, 以此类推。  
值: 0-不关联, 1-关联, 最多支持关联到 8 个 IO 输出口(兼容 byRelatedIOOut 字段)

*struInterval*

抓拍间隔参数

*byRelatedIOOut*

关联的 IO 输出口, 可以同时关联多个

*byFlashMode*

闪光灯闪烁模式: 0- 同时闪, 1- 轮流闪

*byLowSpeedLimit*

小车低速限速, 单位: km/h

*byBigCarLowSpeedLimit*

大车低速限速, 单位: km/h

*byRelaLaneDirectionType*

关联车道方向类型, 定义详见 [ITC\\_RELA\\_LANE\\_DIRECTION\\_TYPE](#)

*struPlateRecog*

车道牌识参数

*struLine*

车道线

## 7.249 NET\_ITS\_ILLEGAL\_INFO:违法代码字符信息

```

struct{
    BYTE byIllegalInfo[MAX_ILLEGAL_LEN];
    BYTE byRes[256];
}

```

```
}NET_ITS_ILLEGAL_INFO,*LPNET_ITS_ILLEGAL_INFO;
```

### Members

*byIllegalInfo*

违章类型信息（字符格式）

*byRes*

保留，置为 0

## 7.250 NET\_ITS\_IMGMERGE\_CFG: 图片合成配置参数

```
struct{
    DWORD    dwSize;
    BYTE     byIsMerge;
    BYTE     byCloseupProportion;
    BYTE     byRes1[2];
    DWORD    dwOneMergeType;
    DWORD    dwTwoMergeType;
    DWORD    dwThreeMergeType;
    DWORD    dwJpegQuality;
    DWORD    dwCloseupIndex;
    DWORD    dwMerageMaxSize;
    WORD     wCloseupDeviation;
    BYTE     byRes[30];
}NET_ITS_IMGMERGE_CFG, *LPNET_ITS_IMGMERGE_CFG;
```

### Members

*dwSize*

结构体大小

*byIsMerge*

是否需要合成：0- 不合成，1- 合成

*byCloseupProportion*

特写图比例：1- 原始分辨率大小，2- 宽高为原始分辨率两倍，以此类推，取值范围：1~8

*byRes1*

保留，置为 0

*dwOneMergeType*

抓拍单张图合成方式（保留）

*dwTwoMergeType*

抓拍两张图合成方式

*dwThreeMergeType*

抓拍三张图合成方式

*dwJpegQuality*

压缩质量（0~100）

*dwCloseupIndex*

特写图抓拍第几张

*dwMerageMaxSize*

合成图片大小的上限，取值范围：300-10240，单位：KB

**wCloseupDeviation**

特写图偏移量(50~图片高度), 单位:像素

**byRes**

保留, 置为 0

**Remarks**

wCloseupDeviation 是指特写图在抓拍图片中截取的上下偏移量 (调整位置)。仅在图片合成启用下才会取特写图。

图片总数	合成编码	合成方式
2	201	两张图片上下排列合成, 合成后图片宽为单张图的 1 倍, 高为 2 倍。
	202	两张图片左右排列合成, 合成后图片宽为单张图的 2 倍, 高为 1 倍。
	203	上下合成, 特写在最上边
	204	上下合成, 特写在最下边
	205	左右合成, 特写在最右边
	206	左右合成, 特写在最左边
	207	品字合成, 特写图在上边
	208	倒品字合成, 特写图在下边
	209	唱字形, 左边 1 张为特写
	210	反唱字形, 右边 1 张为特写
3	301	三张图片上下排列合成, 合成后图片宽为单张图的 1 倍, 高为 3 倍。
	302	三张图片左右排列合成, 合成后图片宽为单张图的 3 倍, 高为 1 倍。
	303	从三张图片中取一幅车牌特写区域, 组成四张图片; 四张图片“田”字型合成, 特写在右下; 合成后图片宽为单张图的 2 倍, 高为 2 倍。
	304	从三张图片中取一幅车牌特写区域, 组成四张图片; 四张图片“田”字型合成, 特写在右下; 合成后图片宽为单张图的 1 倍, 高为 1 倍。
	305	从三张图片中取一幅车牌特写区域, 组成四张图片; 四张图片“田”字型合成, 特写在左上; 合成后图片宽为单张图的 2 倍, 高为 2 倍。
	306	从三张图片中取一幅车牌特写区域, 组成四张图片; 四张图片“田”字型合成, 特写在左上; 压缩图片分辨率, 合成后图片宽为单张图的 1 倍, 高为 1 倍。
	307	三张图片“品”字型排列合成。
	308	三张图片倒“品”字型排列合成。
	309	从三张图片中取一幅车牌特写区域, 组成四张图片; 四张图片“田”字型合成, 特写在左下; 合成后图片宽为单张图的 2 倍, 高为 2 倍。
	310	从三张图片中取一幅车牌特写区域, 组成四张图片; 四张图片“田”字型合成, 特写在左下; 合成后图片宽为单张图的 1 倍, 高为 1 倍。
	311	从三张图片中取一幅车牌特写区域, 组成四张图片; 四张图片“田”字型合成, 特写在右上; 合成后图片宽为单张图的 2 倍, 高为 2 倍。
	312	从三张图片中取一幅车牌特写区域, 组成四张图片; 四张图片“田”字型合成, 特写在右上; 压缩图片分辨率, 合成后图片宽为单张图的 1 倍, 高为 1 倍。

313	从三张图片中取一幅车牌特写区域，组成四张图片；四张图片横“一”字型合成，顺序为：特写图、场景图、场景图、场景图
314	从三张图片中取一幅车牌特写区域，组成四张图片；四张图片横“一”字型合成，顺序为：场景图、特写图、场景图、场景图
315	从三张图片中取一幅车牌特写区域，组成四张图片；四张图片横“一”字型合成，顺序为：场景图、场景图、特写图、场景图
316	从三张图片中取一幅车牌特写区域，组成四张图片；四张图片横“一”字型合成，顺序为：场景图、场景图、场景图、特写图
317	从三张图片中取一幅车牌特写区域，组成四张图片；四张图片竖“1”字型合成，顺序为：特写图、场景图、场景图、场景图
318	从三张图片中取一幅车牌特写区域，组成四张图片；四张图片竖“1”字型合成，顺序为：场景图、特写图、场景图、场景图
319	从三张图片中取一幅车牌特写区域，组成四张图片；四张图片竖“1”字型合成，顺序为：场景图、场景图、特写图、场景图
320	从三张图片中取一幅车牌特写区域，组成四张图片；四张图片竖“1”字型合成，顺序为：场景图、场景图、场景图、特写图
其他	其他

## 7.251 NET\_ITS\_OVERLAP\_CFG\_V50:字符叠加配置

```

struct{
    DWORD                dwSize;
    BYTE                 byEnable;
    BYTE                 byRes1[3];
    NET_ITS_OVERLAP_ITEM_PARAM_V50  struOverLapItem;
    NET_ITS_OVERLAP_INFO_PARAM       struOverLapInfo;
    BYTE                 byRes[120];
}NET_ITS_OVERLAP_CFG_V50,*LPNET_ITS_OVERLAP_CFG_V50;

```

### Members

#### *dwSize*

结构体大小

#### *byEnable*

是否启用：0- 不启用，1- 启用

#### *byRes1*

保留

#### *struOverLapItem*

字符串参数

#### *struOverLapInfo*

字符串内容信息

#### *byRes*

保留

**Remarks**

字符叠加配置能力：[NET\\_DVR\\_GetDeviceAbility](#)(能力集类型：DEVICE\_ABILITY\_INFO)，获取抓拍机和智能终端能力(ITDeviceAbility)，对应节点：<imageOverlayString>。

**7.252 NET\_ITS\_OVERLAP\_INFO\_PARAM:字符叠加内容信息**

```
struct{
    BYTE    bySite[128];
    BYTE    byRoadNum[32];
    BYTE    byInstrumentNum[32];
    BYTE    byDirection[32];
    BYTE    byDirectionDesc[32];
    BYTE    byLaneDes[32];
    BYTE    byRes1[32];
    BYTE    byMonitoringSite1[44];
    BYTE    byMonitoringSite2[32];
    BYTE    byRes[64];
}NET_ITS_OVERLAP_INFO_PARAM,*LPNET_ITS_OVERLAP_INFO_PARAM;
```

**Members***bySite*

地点描述

*byRoadNum*

路口编号

*byInstrumentNum*

设备编号

*byDirection*

方向编号

*byDirectionDesc*

方向描述

*byLaneDes*

车道描述

*byRes1*

保留，置为 0

*byMonitoringSite1*

监测点 1 信息

*byMonitoringSite2*

监测点 2 信息

*byRes*

保留，置为 0

**7.253 NET\_ITS\_OVERLAP\_ITEM\_PARAM\_V50:字符串参数**

```
struct{
```

```

NET_ITS_OVERLAP_SINGLE_ITEM_PARAM_V50    struSingleItem[MAX_OVERLAP_ITEM_NUM];
DWORD                                     dwLinePercent;
DWORD                                     dwItemsStlye;
WORD                                       wStartPosTop;
WORD                                       wStartPosLeft;
WORD                                       wCharStyle;
WORD                                       wCharSize;
WORD                                       wCharInterval;
BYTE                                       byRes1[2];
DWORD                                     dwForeClorRGB;
DWORD                                     dwBackClorRGB;
BYTE                                       byColorAdapt;
BYTE                                       byParamFillZeroEnble;
BYTE                                       byPlateLeftCornerEnable;
BYTE                                       byRes2;
WORD                                       wStartSPicPosTop;
WORD                                       wStartSPicPosLeft;
BYTE                                       byOsdLocate;
BYTE                                       byRes[63];
}NET_ITS_OVERLAP_ITEM_PARAM_V50,*LPNET_ITS_OVERLAP_ITEM_PARAM_V50;

```

### Members

#### *struSingleItem*

单条字符参数

#### *dwLinePercent*

叠加行百分比(0~100)，默认：100

#### *dwItemsStlye*

叠加方式：0-横排，1-竖排(默认横排)

#### *wStartPosTop*

起始上坐标，只对图片内部叠加有效，取值范围：[0~2448]，默认：0

#### *wStartPosLeft*

起始左坐标,只对图片内部叠加有效，取值范围：[0~2448]，默认：0

#### *wCharStyle*

字体类型：0-宋体，1-魏体(默认)

#### *wCharSize*

字符大小：0- 16\*16，1- 32\*32，2- 48x48，3- 64x64(默认)

#### *wCharInterval*

字符间距，取值范围：[0~16]，可设单位：像素(默认)

#### *byRes1*

保留，置为0

#### *dwForeClorRGB*

前景色的 RGB 值，bit0~bit7: B，bit8~bit15: G，bit16~bit23: R，默认：x00FFFFFF-白

#### *dwBackClorRGB*

背景色的 RGB 值，只对图片外叠加有效，bit0~bit7: B，bit8~bit15: G，bit16~bit23: R，默认：x00000000-黑

**byColorAdapt**

颜色是否自适应：0-否，1-是

**byParamFillZeroEnble**

参数补零使能：0- 补零，1- 不补零，速度、限速值不足 3 位时补 0

**byPlateLeftCornerEnable**

车牌小图叠加左上角使能 0-不叠加, 1-叠加

**byRes2**

保留，置为 0

**wStartSPicPosTop**

起始上坐标，只对图片内部叠加有效，取值范围：[0~2448](默认 0)

**wStartSPicPosLeft**

起始左坐标，只对图片内部叠加有效，取值范围：[0~2448](默认 0)

**byOsdLocate**

OSD 叠加位置：0- 图片内，1- 图片上边缘，2- 图片下边缘（合成图专用的是上边缘外）

**byRes**

保留，置为 0

## 7.254 NET\_ITS\_OVERLAP\_SINGLE\_ITEM\_PARAM\_V50:单条字符叠加信息

```
struct{
    BYTE    byRes1[2];
    BYTE    byItemType;
    BYTE    byChangeLineNum;
    BYTE    bySpaceNum;
    BYTE    byRes2[2];
    BYTE    byEnablePos;
    WORD    wStartPosTop;
    WORD    wStartPosLeft;
    BYTE    byItemTypeCustom[32];
    BYTE    byRes[8];
}NET_ITS_OVERLAP_SINGLE_ITEM_PARAM_V50,*LP NET_ITS_OVERLAP_SINGLE_ITEM_PARAM_V50;
```

### Members

**byRes1**

保留

**byItemType**

类型，定义如下所示：

```
enum _ITS_OVERLAP_ITEM_TYPE_{
    OVERLAP_ITEM_NULL    = 0,
    OVERLAP_ITEM_SITE    = 1,
    OVERLAP_ITEM_ROADNUM,
    OVERLAP_ITEM_INSTRUMENTNUM,
    OVERLAP_ITEM_DIRECTION,
```



```

OVERLAP_ITEM_DIRECTIONDESC,
OVERLAP_ITEM_DIRECTIONDESC,
OVERLAP_ITEM_LANEDES,
OVERLAP_ITEM_CAPTIME,
OVERLAP_ITEM_CAPTIME_MILLSECOND,
OVERLAP_ITEM_PLATENUM,
OVERLAP_ITEM_CARCOLOR,
OVERLAP_ITEM_CARTYPE,
OVERLAP_ITEM_CARBRAND,
OVERLAP_ITEM_CARSPEED,
OVERLAP_ITEM_SPEEDLIMIT,
OVERLAP_ITEM_CARLENGTH,
OVERLAP_ITEM_ILLEGALNUM,
OVERLAP_ITEM_MONITOR_INFO,
OVERLAP_ITEM_ILLEGALDES,
OVERLAP_ITEM_OVERSPEED_PERCENT,
OVERLAP_ITEM_RED_STARTTIME,
OVERLAP_ITEM_RED_STOPTIME,
OVERLAP_ITEM_RED_DURATION,
OVERLAP_ITEM_SECURITY_CODE,
OVERLAP_ITEM_CAP_CODE,
OVERLAP_ITEM_SEATBELT,
OVERLAP_ITEM_MONITOR_ID,
OVERLAP_ITEM_SUN_VISOR,
OVERLAP_ITEM_LANE_DIRECTION,
OVERLAP_ITEM_LICENSE_PLATE_COLOR,
OVERLAP_ITEM_SCENE_NUMBER,
OVERLAP_ITEM_SCENE_NAME,
OVERLAP_ITEM_YELLOW_SIGN_CAR,
OVERLAP_ITEM_DANGEROUS_CAR,
OVERLAP_ITEM_CAR_SUBBRAND
}ITS_OVERLAP_ITEM_TYPE

```

*ITC\_PROTOCOL\_UNKNOW*

未知

*OVERLAP\_ITEM\_NULL*

0- 未知

*OVERLAP\_ITEM\_SITE*

1- 地点

*OVERLAP\_ITEM\_ROADNUM*

2- 路口编号

*OVERLAP\_ITEM\_INSTRUMENTNUM*

3- 设备编号

*OVERLAP\_ITEM\_DIRECTION*

4- 方向编号

*OVERLAP\_ITEM\_DIRECTIONDESC*

5- 方向

*OVERLAP\_ITEM\_LANENUM*

6- 车道号

*OVERLAP\_ITEM\_LANEDES*

7- 车道

*OVERLAP\_ITEM\_CAPTIME*

8- 抓拍时间(不带毫秒)

*OVERLAP\_ITEM\_CAPTIME\_MILLSECOND*

9- 抓拍时间(带毫秒)

*OVERLAP\_ITEM\_PLATENUM*

10- 车牌号

*OVERLAP\_ITEM\_CARCOLOR*

11- 车身颜色

*OVERLAP\_ITEM\_CARTYPE*

12- 车辆类型

*OVERLAP\_ITEM\_CARBRAND*

13- 车辆品牌

*OVERLAP\_ITEM\_CARSPEED*

14- 车辆速度

*OVERLAP\_ITEM\_SPEEDLIMIT*

15- 限速标志

*OVERLAP\_ITEM\_CARLENGTH*

16- 车辆长度, 1~99m

*OVERLAP\_ITEM\_ILLEGALNUM*

17- 违法代码(违法代码叠加应该没用的, 应该直接叠加违法信息, 比如正常、低速、超速、逆行、闯红灯、占道、压黄线等)

*OVERLAP\_ITEM\_MONITOR\_INFO*

18- 监测点信息

*OVERLAP\_ITEM\_ILLEGALDES*

19- 违法行为

*OVERLAP\_ITEM\_OVERSPEED\_PERCENT*

20- 超速比

*OVERLAP\_ITEM\_RED\_STARTTIME*

21- 红灯开始时间

*OVERLAP\_ITEM\_RED\_STOPTIME*

22- 红灯结束时间

*OVERLAP\_ITEM\_RED\_DURATION*

23- 红灯已亮时间

*OVERLAP\_ITEM\_SECUNITY\_CODE*

24- 防伪码

*OVERLAP\_ITEM\_CAP\_CODE*

25- 抓拍编号

*OVERLAP\_ITEM\_SEATBELT*

26- 有无系安全带

*OVERLAP\_ITEM\_MONITOR\_ID*

27- 监测点编号

*OVERLAP\_ITEM\_SUN\_VISOR*

28- 遮阳板

*OVERLAP\_ITEM\_LANE\_DIRECTION*

29- 车道行驶方向

*OVERLAP\_ITEM\_LICENSE\_PLATE\_COLOR*

30- 车牌颜色

*OVERLAP\_ITEM\_SCENE\_NUMBER*

31- 场景编号

*OVERLAP\_ITEM\_SCENE\_NAME*

32- 场景名称

*OVERLAP\_ITEM\_YELLOW\_SIGN\_CAR*

33- 黄标车

*OVERLAP\_ITEM\_DANGEROUS\_CAR*

34- 危险品车

*OVERLAP\_ITEM\_CAR\_SUBBRAND*

35- 车辆子品牌

*byChangeLineNum*

叠加项后的换行数，取值范围：[0,10]，默认：0

*bySpaceNum*

叠加项后的空格数，取值范围：[0-255]，默认：0

*byRes2*

保留

*byEnablePos*

是否启用坐标显示：0- 否，1- 是

*wStartPosTop*

起始上坐标，只对图片内部叠加有效，取值范围：[0,2448]，默认：0

*wStartPosLeft*

起始左坐标，只对图片内部叠加有效，取值范围：[0,2448]，默认：0

*byItemCustom*

自定义类型，与 *byItemType* 参数对应，可将 *byItemType* 参数类型自定义名称。若自定义内容为空，便默认以 *byItemType* 参数中的类型命名

*byRes*

保留

## 7.255 NET\_ITS\_OVERLAPCFG\_COND:字符叠加配置条件

struct{

DWORD dwSize;

DWORD dwChannel;

DWORD dwConfigMode;

BYTE byPicModeType;

```

    BYTE        byRes[15];
}NET_ITS_OVERLAPCFG_COND,*LPNET_ITS_OVERLAPCFG_COND;

```

### Members

#### *dwSize*

结构体大小

#### *dwChannel*

通道号

#### *dwConfigMode*

配置模式：0- 终端，1- 前端(直连前端或终端接前端)

#### *byPicModeType*

图片模式：0- 小图(独立图)，1- 大图(合成图)

#### *byRes*

保留

## 7.256 NET\_ITS\_PICTURE\_INFO:图片信息结构体

```

struct{
    DWORD        dwDataLen;
    BYTE         byType;
    BYTE         byDataType;
    BYTE         byCloseUpType;
    BYTE         byPicRecogMode;
    DWORD        dwRedLightTime;
    BYTE         byAbsTime[32];
    NET\_VCA\_RECT struPlateRect;
    NET\_VCA\_RECT struPlateRecgRect;
    BYTE         *pBuffer;
    DWORD        dwUTCTime;
    BYTE         byCompatibleAblity;
    BYTE         byRes2[7];
}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，6- 人脸子图(主驾驶)，7- 人脸子图(副驾驶)，8- 非机动车，9- 行人

#### *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）

#### *byCloseUpType*

特写图类型：0- 保留，1- 非机动车，2- 行人

#### *byPicRecogMode*

图片背向识别：0- 正向车牌识别，1- 背向识别(尾牌识别)

***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

***dwUTCTime***

UTC 时间

***byCompatibleAblity***

兼容能力字段，按位表示，值：0- 无效，1- 有效

bit0- 表示 dwUTCTime 字段是否有效

***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。

**7.257 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                 byDetSceneID;
    BYTE                 byVehicleAttribute;
    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                    byRelaLaneDirectionType;
BYTE                    byRes3;
DWORD                   dwCustomIllegalType;
BYTE                    byIllegalFromatType;
BYTE                    *pIllegalInfoBuf;
BYTE                    byRes4[4];
BYTE                    byPilotSafebelt;
BYTE                    byCopilotSafebelt;
BYTE                    byPilotSunVisor;
BYTE                    byCopilotSunVisor;
BYTE                    byPilotCall;
BYTE                    byBarrierGateCtrlType;
BYTE                    byAlarmDataType;
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）

#### *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

#### *byDetSceneID*

检测场景号，0 表示无效，其他取值：[1,4]，IPC 为 0（不支持）

**byVehicleAttribute**

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

**wIllegalType**

违章类型，采用国标定义。wIllegalType 为 0 时违法类型见 dwCustomIllegalType，wIllegalType 和 dwCustomIllegalType 都为 0 时表示正常卡口抓拍

**byIllegalSubType**

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

**byRelaLaneDirectionType**

关联车道方向类型，定义详见 [ITC\\_RELA\\_LANE\\_DIRECTION\\_TYPE](#)，与关联车道号对应，确保车道唯一性

**byRes3**

保留

**dwCustomIllegalType**

违章类型定义(用户自定义)：当 wIllegalType 参数为空时，使用该参数；若 wIllegalType 参数为有值时，以 wIllegalType 参数为准，该参数无效

**byIllegalFromatType**

违章信息格式类型：0- 数字格式，1- 字符格式

**plllegalInfoBuf**

违法代码字符信息，byIllegalFromatType 为 1 时有效(此时数字格式的国标违章代码参数 wIllegalType、dwCustomIllegalType 仍有效)，字符信息指针指向结构体 [NET\\_ITS\\_ILLEGAL\\_INFO](#)

**byRes4**

保留

**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

*byPilotCall*

主驾驶员是否在打电话(保留, 暂不支持): 0- 未知, 1- 未打电话, 2- 打电话

*byBarrierGateCtrlType*

道闸控制类型: 0- 开闸, 1- 未开闸 (专用于历史数据中相机根据允许名单匹配后是否开闸成功的标志)

*byAlarmDataType*

报警数据类型: 0- 实时数据, 1- 历史数据

*struSnapFirstPicTime*

端点时间(ms) (抓拍第一张图片的时间)

*dwlllegalTime*

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

*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 6 张图, 由序号区分

## Remarks

违章代码:

违章代码	违章描述
0	普通卡口数据
1018	机动车不在机动车道内行驶的(车道禁行)
1019	机动车违反规定使用专用车道的(预留)
1208	不按导向标志行驶 (违章左转、违章直行、违章右转等)
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)
1229	违反禁令标志(预留, 目前没有提供此类违章代码)
1230	机动车禁止标线指示 (压车道线等)
1301	逆行
1302	违反信号灯 (闯红灯)
1042	机动车不按规定车道行驶
1303	机动车行驶超过规定时速 50%以下
1603	机动车行驶超过规定时速 50%



## 7.258 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

## 7.259 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];
    NET\_DVR\_TIME\_V30    struTriggerTime;
}

```

```

BYTE                byRes[4];
}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

#### *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

#### *struTriggerTime*

触发时间

#### *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

## 7.260 NET\_ITS\_TRAFFIC\_COLLECT: 交通统计数据结构体

```
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

## 7.261 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

*wAverSpeed*

平均车速(km/h)

*wAverCarDis*

平均时距，单位：毫秒

*byRes2*

保留

## 7.262 NET\_POS\_PARAM: 区域参数

```
struct{
```

```

WORD    wLeft;
WORD    wTop;
WORD    wRight;
WORD    wBottom;

```

```
}NET_POS_PARAM, *LPNET_POS_PARAM;
```

### Members

*wLeft*

边界框左上角 X 轴坐标，取值范围：[1,1000]

*wTop*

边界框左上角 Y 轴坐标，取值范围：[1,1000]

*wRight*

边界框右下角 X 轴坐标，取值范围：[1,1000]

*wBottom*

边界框右下角 Y 轴坐标，取值范围：[1,1000]

## 7.263 NET\_VCA\_LINE:线结构参数

```
struct{  
    NET_VCA_POINT    struStart;  
    NET_VCA_POINT    struEnd;  
}NET_VCA_LINE,*LPNET_VCA_LINE;
```

### Members

*struStart*

起点

*struEnd*

终点

## 7.264 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]

## 7.265 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]

## 7.266 ITC\_LANE\_DIRECTION\_TYPE: 车道方向类型枚举

```
enum _ITC_LANE_DIRECTION_TYPE_{
    ITC_LANE_DIRECTION_UNKNOW    = 0,
    ITC_LANE_LEFT                 = 1,
    ITC_LANE_STRAIGHT            = 2,
    ITC_LANE_LEFT_STRAIGHT       = 3,
    ITC_LANE_RIGHT               = 4,
    ITC_LANE_LEFT_RIGHT          = 5,
    ITC_LANE_RIGHT_STRAIGHT      = 6,
    ITC_LANE_LEFT_RIGHT_STRAIGHT = 7,
    ITC_LANE_LEFT_WAIT           = 9,
    ITC_LANE_STRAIGHT_WAIT       = 10,
    ITC_LANE_FORWARD             = 11,
    ITC_LANE_BACKWARD            = 12,
    ITC_LANE_BOTHWAY             = 13,
    ITC_LANE_STRAIGHT_WAIT_RIGHT = 14
}ITC_LANE_DIRECTION_TYPE
```

### Members

*ITC\_LANE\_DIRECTION\_UNKNOW*

未知

*ITC\_LANE\_LEFT*

左转

*ITC\_LANE\_STRAIGHT*

直行

*ITC\_LANE\_LEFT\_STRAIGHT*

左转+直行

*ITC\_LANE\_RIGHT*

右转

*ITC\_LANE\_LEFT\_RIGHT*

左转+右转

*ITC\_LANE\_RIGHT\_STRAIGHT*

右转+直行

*ITC\_LANE\_LEFT\_RIGHT\_STRAIGHT*

左转+右转+直行

*ITC\_LANE\_LEFT\_WAIT*

左转待行

*ITC\_LANE\_STRAIGHT\_WAIT*

直行待行

*ITC\_LANE\_FORWARD*

正向行驶

*ITC\_LANE\_BACKWARD*

背向行驶

*ITC\_LANE\_BOTHWAY*

双向行驶

*ITC\_LANE\_STRAIGHT\_WAIT\_RIGHT*

直行待行 + 右转

## 7.267 **ITC\_LANE\_USEAGE\_TYPE**: 车道用途类型枚举

```
enum _ITC_LANE_USEAGE_TYPE_{
    ITC_LANE_USEAGE_UNKNOW = 0,
    ITC_LANE_CARRIAGEWAY   = 1,
    ITC_LANE_BUS           = 2,
    ITC_LANE_FAST         = 3,
    ITC_LANE_SLOW         = 4,
    ITC_LANE_MOTOR        = 5,
    ITC_LANE_NONMOTOR     = 6,
    ITC_LANE_REVERSE_LANE = 7,
    ITC_LANE_BAN_TRUCKS   = 8,
    ITC_LANE_MIX          = 9,
    ITC_LANE_EMERGENCY    = 10,
    ITC_LANE_BAN_LEFT     = 11,
    ITC_LANE_BAN_RIGHT    = 12
}ITC_LANE_USEAGE_TYPE
```

### Members

*ITC\_LANE\_USEAGE\_UNKNOW*

未知

*ITC\_LANE\_CARRIAGEWAY*

普通车道

*ITC\_LANE\_BUS*

公交车专用道

*ITC\_LANE\_FAST*

快车道

*ITC\_LANE\_SLOW*

慢车道

*ITC\_LANE\_MOTOR*

摩托车道

*ITC\_LANE\_NONMOTOR*

非机动车道

*ITC\_LANE\_REVERSE\_LANE*

反向车道

*ITC\_LANE\_BAN\_TRUCKS*

禁止货车车道

*ITC\_LANE\_MIX*

混合车道

*ITC\_LANE\_EMERGENCY*



应急车道

*ITC\_LANE\_BAN\_LEFT*

禁左车道

*ITC\_LANE\_BAN\_RIGHT*

禁右车道

## 7.268 **ITC\_RELA\_LANE\_DIRECTION\_TYPE**:关联车道方向类型枚举

```
enum _ITC_RELA_LANE_DIRECTION_TYPE_{
    ITC_RELA_LANE_DIRECTION_UNKNOW    = 0,
    ITC_RELA_LANE_EAST_WEST           = 1,
    ITC_RELA_LANE_WEST_EAST           = 2,
    ITC_RELA_LANE_SOUTH_NORTH         = 3,
    ITC_RELA_LANE_NORTH_SOUTH         = 4,
    ITC_RELA_LANE_EASTSOUTH_WESTNORTH = 5,
    ITC_RELA_LANE_WESTNORTH_EASTSOUTH = 6,
    ITC_RELA_LANE_EASTNORTH_WESTSOUTH = 7,
    ITC_RELA_LANE_WESTSOUTH_EASTNORTH = 8
}ITC_RELA_LANE_DIRECTION_TYPE
```

### Members

*ITC\_RELA\_LANE\_DIRECTION\_UNKNOW*

其它

*ITC\_RELA\_LANE\_EAST\_WEST*

从东向西

*ITC\_RELA\_LANE\_WEST\_EAST*

从西向东

*ITC\_RELA\_LANE\_SOUTH\_NORTH*

从南向北

*ITC\_RELA\_LANE\_NORTH\_SOUTH*

从北向南

*ITC\_RELA\_LANE\_EASTSOUTH\_WESTNORTH*

从东南向西北

*ITC\_RELA\_LANE\_WESTNORTH\_EASTSOUTH*

从西北向东南

*ITC\_RELA\_LANE\_EASTNORTH\_WESTSOUTH*

从东北向西南

*ITC\_RELA\_LANE\_WESTSOUTH\_EASTNORTH*

从西南向东北

## 7.269 **ITC\_TRIGGERMODE\_TYPE**:触发模式

```
enum _ITC_TRIGGERMODE_TYPE_{
```

ITC_POST_IOSPEED_TYPE	= 0x1,
ITC_POST_SINGLEIO_TYPE	= 0x2,
ITC_POST_RS485_TYPE	= 0x4,
ITC_POST_RS485_RADAR_TYPE	= 0x8,
ITC_POST_VIRTUALCOIL_TYPE	= 0x10,
ITC_POST_HVT_TYPE_V50	= 0x20,
ITC_POST_MPR_TYPE	= 0x40,
ITC_POST_PRS_TYPE	= 0x80,
ITC_EPOLICE_IO_TRAFFICLIGHTS_TYPE	= 0x100,
ITC_EPOLICE_RS485_TYPE	= 0x200,
ITC_PE_RS485_TYPE	= 0x10000,
ITC_VIDEO_EPOLICE_TYPE	= 0x20000,
ITC_VIA_VIRTUALCOIL_TYPE	= 0x40000,
ITC_POST_IMT_TYPE	= 0x80000,
IPC_POST_HVT_TYPE	= 0x100000,
ITC_POST_MOBILE_TYPE	= 0x200000,
ITC_REDLIGHT_PEDESTRIAN_TYPE	= 0x400000,
ITC_NOCOMITY_PEDESTRIAN_TYPE	= 0x800000

}ITC\_TRIGGERMODE\_TYPE

### Members

*ITC\_POST\_IOSPEED\_TYPE*

IO 测速（卡口）

*ITC\_POST\_SINGLEIO\_TYPE*

单 IO 触发（卡口）

*ITC\_POST\_RS485\_TYPE*

RS485 车检器触发（卡口）

*ITC\_POST\_RS485\_RADAR\_TYPE*

RS485 雷达触发（卡口）

*ITC\_POST\_VIRTUALCOIL\_TYPE*

虚拟线圈触发（卡口）

*ITC\_POST\_HVT\_TYPE\_V50*

混行卡口视频触发 V50（卡口）

*ITC\_POST\_MPR\_TYPE*

多帧识别（卡口）

*ITC\_POST\_PRS\_TYPE*

视频检测触发配置

*ITC\_EPOLICE\_IO\_TRAFFICLIGHTS\_TYPE*

IO 红绿灯（电警）

*ITC\_EPOLICE\_RS485\_TYPE*

RS485 车检器电警触发（电警）

*ITC\_PE\_RS485\_TYPE*

RS485 车检器卡式电警触发（卡式电警）

*ITC\_VIDEO\_EPOLICE\_TYPE*

视频电警触发（卡式电警）

*ITC\_VIA\_VIRTUALCOIL\_TYPE*

VIA 触发配置

*ITC\_POST\_IMT\_TYPE*

智慧监控配置

*IPC\_POST\_HVT\_TYPE*

(IPC) 混行卡口模式

*ITC\_POST\_MOBILE\_TYPE*

移动交通触发模式

*ITC\_REDLIGHT\_PEDESTRIAN\_TYPE*

行人闯红灯触发

*ITC\_NOCOMITY\_PEDESTRIAN\_TYPE*

不礼让行人触发

### See Also

- 触发类型为 *ITC\_VIDEO\_EPOLICE\_TYPE*(卡式电警-视频电警触发)时，触发参数通过接口 [NET\\_DVR\\_GetDeviceConfig](#)(命令: *NET\_ITC\_GET\_VIDEO\_TRIGGERCFG*)、[NET\\_DVR\\_SetDeviceConfig](#)(命令: *NET\_ITC\_SET\_VIDEO\_TRIGGERCFG*)进行配置。
- 触发类型为其他类型时，相关触发参数通过 [NET\\_DVR\\_GetDVRConfig](#)(命令: *NET\_ITC\_GET\_TRIGGERCFG*)、[NET\\_DVR\\_SetDVRConfig](#)(命令: *NET\_ITC\_SET\_TRIGGERCFG*)进行配置。

## 7.270 ITC\_VIOLATION\_DETECT\_TYPE:违规检测类型枚举

```
enum _ITC_VIOLATION_DETECT_TYPE_{
    ITC_VIOLATION_POST                = 0x01,
    ITC_VIOLATION_DRIVELINE           = 0x02,
    ITC_VIOLATION_REVERSE              = 0x04,
    ITC_VIOLATION_REDLIGHT             = 0x08,
    ITC_VIOLATION_DIRECTION            = 0x10,
    ITC_VIOLATION_INTERSECTION_CONGEST = 0x20,
    ITC_VIOLATION_NONDRIVEWAY          = 0x40,
    ITC_VIOLATION_CHANGELANE           = 0x80,
    ITC_VIOLATION_BAN                  = 0x100,
    ITC_VIOLATION_INTERSECTION_PARK     = 0x200,
    ITC_VIOLATION_GREEN_PARK           = 0x400,
    ITC_VIOLATION_BAN_DRIVE            = 0x800,
    ITC_VIOLATION_ACROSS_YELLOWLINE    = 0x1000,
    ITC_VIOLATION_HIGH_SPEED           = 0x2000,
    ITC_VIOLATION_LOW_SPEED            = 0x4000,
    ITC_VIOLATION_TURN_AROUND          = 0x8000,
    ITC_VIOLATION_CONGESTION           = 0x10000
}ITC_VIOLATION_DETECT_TYPE
```

### Members

*ITC\_VIOLATION\_POST*

卡口抓拍

*ITC\_VIOLATION\_DRIVELINE*

压车道线抓拍

*ITC\_VIOLATION\_REVERSE*

逆行抓拍

*ITC\_VIOLATION\_REDLIGHT*

闯红灯抓拍

*ITC\_VIOLATION\_DIRECTION*

不按导向行驶抓拍

*ITC\_VIOLATION\_INTERSECTION\_CONGEST*

路口滞留抓拍

*ITC\_VIOLATION\_NONDRIVEWAY*

机动车占非机动车车道抓拍

*ITC\_VIOLATION\_CHANGELANE*

违法变道

*ITC\_VIOLATION\_BAN*

违法禁令

*ITC\_VIOLATION\_INTERSECTION\_PARK*

红灯越线

*ITC\_VIOLATION\_GREEN\_PARK*

绿灯停车

*ITC\_VIOLATION\_BAN\_DRIVE*

禁行

*ITC\_VIOLATION\_ACROSS\_YELLOWLINE*

跨越黄线

*ITC\_VIOLATION\_HIGH\_SPEED*

超速（卡口专用）

*ITC\_VIOLATION\_LOW\_SPEED*

低速（卡口专用）

*ITC\_VIOLATION\_TURN\_AROUND*

违法掉头

*ITC\_VIOLATION\_CONGESTION*

拥堵

#### Remarks

- 跨越黄线与违法变道区别：跨越黄线是针对相机架设在路边（不调杆），抓拍车辆变道或调头的情况；违法变道与电警中的概念是一样的。
- 禁行和违法禁令区别：禁行是指定区域的车牌都需要抓拍，违法禁令是针对车道抓拍某一类车牌。

## 7.271 VLR\_VEHICLE\_CLASS: 车辆主品牌类型

```
enum _VLR_VEHICLE_CLASS_{
    VLR_OTHER          = 0,    //其它
    VLR_VOLKSWAGEN    = 1,    //大众
    VLR_BUICK          = 2,    //别克
    VLR_BMW            = 3,    //宝马
    VLR_HONDA          = 4,    //本田
```

```
VLR_PEUGEOT      = 5,    //标致
VLR_TOYOTA       = 6,    //丰田
VLR_FORD         = 7,    //福特
VLR_NISSAN       = 8,    //日产
VLR_AUDI         = 9,    //奥迪
VLR_MAZDA        = 10,   //马自达
VLR_CHEVROLET    = 11,   //雪佛兰
VLR_CITROEN      = 12,   //雪铁龙
VLR_HYUNDAI      = 13,   //现代
VLR_CHERY        = 14,   //奇瑞
VLR_KIA          = 15,   //起亚
VLR_ROEWE        = 16,   //荣威
VLR_MITSUBISHI   = 17,   //三菱
VLR_SKODA        = 18,   //斯柯达
VLR_GEELY        = 19,   //吉利
VLR_ZHONGHUA     = 20,   //中华
VLR_VOLVO        = 21,   //沃尔沃
VLR_LEXUS        = 22,   //雷克萨斯
VLR_FIAT         = 23,   //菲亚特
VLR_EMGRAND      = 24,   //帝豪
VLR_DONGFENG     = 25,   //东风
VLR_BYD          = 26,   //比亚迪
VLR_SUZUKI       = 27,   //铃木
VLR_JINBEI       = 28,   //金杯
VLR_HAIMA        = 29,   //海马
VLR_SGMW         = 30,   //五菱
VLR_JAC          = 31,   //江淮
VLR_SUBARU       = 32,   //斯巴鲁
VLR_ENGLON       = 33,   //英伦
VLR_GREATWALL    = 34,   //长城
VLR_HAFEI        = 35,   //哈飞
VLR_ISUZU        = 36,   //五十铃
VLR_SOUEAST      = 37,   //东南
VLR_CHANA        = 38,   //长安
VLR_FOTON        = 39,   //福田
VLR_XIALI        = 40,   //夏利
VLR_BENZ         = 41,   //奔驰
VLR_FAW          = 42,   //一汽
VLR_NAVECO       = 43,   //依维柯
VLR_LIFAN        = 44,   //力帆
VLR_BESTURN      = 45,   //一汽奔腾
VLR_CROWN        = 46,   //皇冠
VLR_RENAULT      = 47,   //雷诺
VLR_JMC          = 48,   //JMC
```

```

VLR_MG      = 49, //MG 名爵
VLR_KAMA    = 50, //凯马
VLR_ZOTYE   = 51, //众泰
VLR_CHANGHE = 52, //昌河
VLR_XMKINGLONG = 53, //厦门金龙
VLR_HUIZHONG = 54, //上海汇众
VLR_SZKINGLONG = 55, //苏州金龙
VLR_HIGER   = 56, //海格
VLR_YUTONG  = 57, //宇通
VLR_CNHTC   = 58, //中国重汽
VLR_BEIBEN  = 59, //北奔重卡
VLR_XINGMA  = 60, //华菱星马
VLR_YUEJIN  = 61, //跃进
VLR_HUANGHAI = 62 //黄海
}VLR_VEHICLE_CLASS

```

## 7.272 XML 描述

### AlarmCtrlParam

```

<AlarmCtrlParam version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <id>
    <!-- req, xs:string;id, hostId-->
  </id>
  <listenPicUploadEnabled>
    <!-- req, xs:boolean, "true, false", 监听图片上传使能-->
  </listenPicUploadEnabled>
</AlarmCtrlParam>

```

## 7.273 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_INPUT	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

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