

设备网络 SDK 编程指南

(For Android)

V5.3

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	10
1.1 SDK 概述	10
1.2 适用范围	11
2 SDK 版本更新	13
3 函数说明	14
3.1 SDK 初始化	14
3.1.1 初始化 SDK NET_DVR_Init	14
3.1.2 释放 SDK 资源 NET_DVR_Cleanup	14
3.2 SDK 本地功能	14
本地参数配置	14
3.2.1 获取 SDK 本地参数 NET_DVR_GetSDKLocalConfig	14
3.2.2 设置 SDK 本地参数 NET_DVR_SetSDKLocalConfig	15
连接和接收超时时间及重连设置	15
3.2.3 设置网络连接超时时间 NET_DVR_SetConnectTime	15
3.2.4 设置重连功能 NET_DVR_SetReconnect	15
3.2.5 设置接收超时时间 NET_DVR_SetRecvTimeOut	15
SDK 版本信息和日志	16
3.2.6 获取 SDK 版本信息 NET_DVR_GetSDKVersion	16
3.2.7 获取 SDK 的版本号和 build 信息 NET_DVR_GetSDKBuildVersion	16
3.2.8 启用写日志文件 NET_DVR_SetLogToFile	16
异常消息回调	17
3.2.9 注册接收异常、重连消息回调函数 NET_DVR_SetExceptionCallBack	17
获取错误信息	18
3.2.10 返回最后操作的错误码 NET_DVR_GetLastError	18
3.2.11 返回最后操作的错误信息 NET_DVR_GetErrorMsg	18
3.3 用户注册	18
3.3.1 设备的动态 IP 和端口解析 NET_DVR_GetAddrInfoByServer	18
3.3.2 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	19
3.3.3 激活设备 NET_DVR_ActivateDevice	19
3.3.4 用户注册设备 NET_DVR_Login_V30	20
3.3.5 用户注销 NET_DVR_Logout_V30	20
3.4 获取设备能力集	20
3.4.1 获取设备能力集 NET_DVR_GetXMLAbility	20
3.5 实时预览	21
强制 I 帧 21	
3.5.1 主码流动态产生一个关键帧 NET_DVR_MakeKeyFrame	21
3.5.2 子码流动态产生一个关键帧 NET_DVR_MakeKeyFrameSub	21

实时预览	22
3.5.3 实时预览 NET_DVR_RealPlay_V40	22
3.5.4 停止预览 NET_DVR_StopRealPlay	22
3.5.5 获取预览时用来解码和显示的播放器句柄 NET_DVR_GetRealPlayerIndex.....	22
播放声音控制	23
3.5.6 播放声音 NET_DVR_OpenSound.....	23
3.5.7 关闭声音 NET_DVR_CloseSound	23
3.5.8 调节播放音量 NET_DVR_Volume	23
显示参数配置	24
3.5.9 获取预览视频显示参数 NET_DVR_ClientGetVideoEffect	24
3.5.10 设置预览视频显示参数 NET_DVR_ClientSetVideoEffect.....	24
零通道预览	24
3.5.11 开始零通道预览 NET_DVR_ZeroStartPlay	24
3.5.12 停止零通道预览 NET_DVR_ZeroStopPlay.....	25
预览抓图	25
3.5.13 设置抓图模式 NET_DVR_SetCapturePictureMode.....	25
3.5.14 预览时抓图并保存成图片文件 NET_DVR_CapturePictureBlock	26
3.5.15 单帧数据捕获并保存成图片 NET_DVR_CapturePicture.....	26
客户端录像	26
3.5.16 捕获预览数据并保存到指定文件中 NET_DVR_SaveRealData	26
3.5.17 捕获预览数据并保存到指定文件中 NET_DVR_SaveRealData_V30.....	27
3.5.18 停止数据捕获 NET_DVR_StopSaveRealData.....	27
实时数据回调	27
3.5.19 实时预览 NET_DVR_SetRealDataCallBack.....	27
3.5.20 捕获实时码流数据（标准码流）NET_DVR_RealPlay_V40	28
3.6 设备抓图	29
3.6.1 单帧数据捕获并保存成 JPEG 图片 NET_DVR_CaptureJPEGPicture	29
3.6.2 单帧数据捕获并保存成 JPEG 存放在指定的内存空间中 NET_DVR_CaptureJPEGPicture_NEW.....	29
3.7 布防、撤防	30
设置报警等信息上传的回调函数	30
3.7.1 注册回调函数，接收设备报警消息 NET_DVR_SetDVRMessageCallBack_V30	30
布防撤防	31
3.7.2 建立报警上传通道，获取报警等信息 NET_DVR_SetupAlarmChan_V41.....	31
3.7.3 撤销报警上传通道 NET_DVR_CloseAlarmChan_V30	31
3.8 远程参数配置	32
通用参数配置	32
3.8.1 获取设备配置信息 NET_DVR_GetDVRConfig.....	32
3.8.2 设置设备配置信息 NET_DVR_SetDVRConfig.....	33
3.8.3 批量获取配置信息 NET_DVR_GetDeviceConfig.....	34
3.8.4 批量设置配置信息 NET_DVR_SetDeviceConfig.....	35
报警输出配置	35
3.8.5 获取设备报警输出 NET_DVR_GetAlarmOut_V30	35
3.8.6 设置设备报警输出 NET_DVR_SetAlarmOut	36

设备支持的云台协议	36
3.8.7 获取设备支持的云台协议 NET_DVR_GetPTZProtocol	36
3.9 录像文件回放、下载、锁定及备份	36
刷新录像索引	36
3.9.1 即时刷新录像索引 NET_DVR_UpdateRecordIndex	36
录像文件的查找	37
3.9.2 根据文件类型、时间查找设备录像文件 NET_DVR_FindFile_V30	37
3.9.3 逐个获取查找到的文件信息 NET_DVR_FindNextFile_V30	37
3.9.4 关闭文件查找，释放资源 NET_DVR_FindClose_V30	38
按事件查找录像	38
3.9.5 按事件查找录像 NET_DVR_FindFileByEvent	38
3.9.6 逐个获取查找到的文件信息 NET_DVR_FindNextEvent	38
3.9.7 关闭文件查找，释放资源 NET_DVR_FindClose_V30	39
回放录像文件	39
3.9.8 注册回调函数，捕获录像数据 NET_DVR_SetPlayDataCallBack	39
3.9.9 按文件名回放录像文件 NET_DVR_PlayBackByName	40
3.9.10 按时间回放录像文件 NET_DVR_PlayBackByTime	40
3.9.11 按时间回放录像文件 NET_DVR_PlayBackByTime_V40	40
3.9.12 控制录像回放的状态 NET_DVR_PlayBackControl_V40	41
3.9.13 获取回放取流进度 NET_DVR_GetPlayBackPos	41
3.9.14 停止回放录像文件 NET_DVR_StopPlayBack	42
3.9.15 获取回放时用来解码显示的播放库句柄 NET_DVR_GetPlayBackPlayerIndex	42
3.9.16 录像回放时抓图 NET_DVR_PlayBackCaptureFile	42
3.9.17 按文件名倒放录像文件 NET_DVR_PlayBackReverseByName	42
3.9.18 按时间倒放录像文件 NET_DVR_PlayBackReverseByTime_V40	43
下载录像文件	43
3.9.19 按文件名下载录像文件 NET_DVR_GetFileByName	43
3.9.20 按时间下载录像文件 NET_DVR_GetFileByTime	43
3.9.21 控制录像下载的状态 NET_DVR_PlayBackControl_V40	44
3.9.22 获取当前下载录像文件的进度 NET_DVR_GetDownloadPos	44
3.9.23 停止下载录像文件 NET_DVR_StopGetFile	45
3.10 云台控制	45
云台控制操作	45
3.10.1 云台控制操作（需先启动预览）NET_DVR_PTZControl	45
3.10.2 云台控制操作（不用启动预览）NET_DVR_PTZControl_Other	46
3.10.3 带速度的云台控制操作（需先启动预览）NET_DVR_PTZControlWithSpeed	47
3.10.4 带速度的云台控制操作（不用启动预览）NET_DVR_PTZControlWithSpeed_Other	47
云台预置点操作	47
3.10.5 云台预置点操作，需先启动预览 NET_DVR_PTZPreset	47
3.10.6 云台预置点操作 NET_DVR_PTZPreset_Other	48
云台巡航操作	48
3.10.7 云台巡航操作，需先启动预览 NET_DVR_PTZPCruise	48
3.10.8 云台巡航操作 NET_DVR_PTZPCruise_Other	49
云台花样扫描操作	50

3.10.9	云台花样扫描操作, 需先启动预览 NET_DVR_PTZTrack.....	50
3.10.10	云台花样扫描操作 NET_DVR_PTZTrack_Other	50
	云台区域缩放控制	50
3.10.11	云台图象区域选择放大或缩小 NET_DVR_PTZSelZoomIn	50
3.10.12	云台图像区域选择放大或缩小 NET_DVR_PTZSelZoomIn_Ex	51
3.11	语音转发	51
3.11.1	获取当前生效的音频对讲音频压缩参数 NET_DVR_GetCurrentAudioCompress.....	51
3.11.2	启动语音转发, 获取编码后的音频数据 NET_DVR_StartVoiceCom_MR_V30	52
3.11.3	转发语音数据 NET_DVR_VoiceComSendData	52
3.11.4	停止语音转发 NET_DVR_StopVoiceCom	52
3.12	数据透传	53
	透明通道 53	
3.12.1	建立透明通道 NET_DVR_SerialStart_V40.....	53
3.12.2	通过透明通道向设备串口发送数据 NET_DVR_SerialSend	53
3.12.3	断开透明通道 NET_DVR_SerialStop.....	53
	向串口发送数据	54
3.12.4	直接向串口发送数据, 不需要建立透明通道 NET_DVR_SendToSerialPort.....	54
3.12.5	直接向 232 串口发送数据, 不需要建立透明通道 NET_DVR_SendTo232Port.....	54
3.13	设备手动录像	54
3.13.1	远程手动启动设备录像 NET_DVR_StartDVRRecord	54
3.13.2	远程手动停止设备录像 NET_DVR_StopDVRRecord	55
3.14	远程面板控制	55
3.14.1	远程控制面板上的按键 NET_DVR_ClickKey.....	55
3.15	硬盘管理	56
3.15.1	远程格式化设备硬盘 NET_DVR_FormatDisk	56
3.15.2	获取格式化硬盘的进度 NET_DVR_GetFormatProgress.....	56
3.15.3	关闭格式化硬盘句柄, 释放资源 NET_DVR_CloseFormatHandle	56
3.16	设备维护管理	57
	设备工作状态	57
3.16.1	获取设备的工作状态 NET_DVR_GetDVRWorkState_V30	57
	UPNP 端口映射状态	57
3.16.2	获取 UPNP 端口映射状态 NET_DVR_GetUpnpNatState	57
	远程升级 57	
3.16.3	设置远程升级时网络环境 NET_DVR_SetNetworkEnvironment	57
3.16.4	远程升级 NET_DVR_Upgrade	58
3.16.5	获取远程升级的进度 NET_DVR_GetUpgradeProgress.....	58
3.16.6	获取远程升级的状态 NET_DVR_GetUpgradeState	58
3.16.7	获取远程升级的阶段信息 NET_DVR_GetUpgradeStep	58
3.16.8	关闭远程升级句柄, 释放资源 NET_DVR_CloseUpgradeHandle	59
	远程重启 59	
3.16.9	重启设备 NET_DVR_RebootDVR	59
3.17	人脸比对功能	59
	人脸比对功能	59
3.17.1	上传文件 NET_DVR_UploadFile_V40	60

3.17.2	发送数据 NET_DVR_UploadSend	60
3.17.3	获取文件上传的进度和状态 NET_DVR_GetUploadState	61
3.17.4	获取当前上传的结果信息 NET_DVR_GetUploadResult.....	61
3.17.5	停止文件上传 NET_DVR_UploadClose.....	62
	图片查找 62	
3.17.6	查找图片 NET_DVR_FindPicture	62
3.17.7	逐个获取查找到的图片信息 NET_DVR_FindNextPicture_V40.....	63
3.17.8	逐个获取查找到的图片信息 NET_DVR_FindNextPicture_V40.....	63
4	错误代码及说明	64
4.1	网络通讯库错误码	64
4.2	RTSP 通讯库错误码	69
4.3	软解码库错误码	70
5	结构体说明	71
5.1	EAP_PEAP:EAP_PEAP 认证参数	71
5.2	EAP_TLS:EAP_TLS 认证参数	71
5.3	EAP_TTLS:EAP_TTLS 认证参数	72
5.4	NET_DVR_ACTIVATECFG:激活参数	72
5.5	NET_DVR_ALARMER:报警设备信息	73
5.6	NET_DVR_ALARMINFO_V30:报警输入参数	74
5.7	NET_DVR_ALARMINFO:设备报警信息	76
5.8	NET_DVR_ALARMINFO_V30:上传的报警信息 V30.....	77
5.9	NET_DVR_ALARMINFO_V40:上传的报警信息 V40.....	78
5.10	NET_DVR_ALARMOUTCFG_V30:报警输出参数	78
5.11	NET_DVR_ALARMOUTSTATUS_V30:报警输出状态	79
5.12	NET_DVR_ALARM_FIXED_HEADER:报警设备信息.....	79
5.13	NET_DVR_AP_INFO:单个无线资源参数	80
5.14	NET_DVR_AP_INFO_LIST:无线网络资源列表.....	81
5.15	NET_DVR_AREAINFOCFG:籍贯参数结构体	81
5.16	NET_DVR_CHANNELSTATE_V30:通道状态	84
5.17	NET_DVR_CLIENTINFO:预览参数	85
5.18	NET_DVR_COMPRESSION_AUDIO:语音对讲音频参数	86
5.19	NET_DVR_COMPRESSION_INFO_V30:码流压缩参数	86
5.20	NET_DVR_COMPRESSIONCFG_V30:通道压缩参数	88
5.21	NET_DVR_DDNSPARA_V30:DDNS 配置参数	89
5.22	NET_DVR_DECODERCFG_V30:云台解码器参数	90
5.23	NET_DVR_DEVICECFG_V40:设备参数	90
5.24	NET_DVR_DEVICEINFO_V30:设备信息	91
5.25	NET_DVR_DISKSTATE:硬盘状态	97
5.26	NET_DVR_DIGITAL_CHANNEL_STATE:数字通道状态	98
5.27	NET_DVR_ETHERNET_V30:以太网参数	99
5.28	NET_DVR_FACE_DETECTION: 人脸侦测报警信息结构体	99
5.29	NET_DVR_FACELIB_COND: 导入人脸数据(人脸图片+图片附件信息)到人脸库的条件.....	101
5.30	NET_DVR_FILECOND:录像文件查找条件	101
5.31	NET_DVR_FINDDATA_V30:录像文件查找结果	102

5.32	NET_DVR_FIND_PICTURE_PARAM: 图片查找条件结构体.....	103
5.33	NET_DVR_HANDLEEXCEPTION_V30:异常参数	107
5.34	NET_DVR_HIDEALARM_V30:遮挡报警参数	107
5.35	NET_DVR_IPADDR:IP 地址	108
5.36	NET_DVR_IPALARMOUTCFG:IP 报警输出配置	109
5.37	NET_DVR_IPALARMOUTINFO:IP 报警输出信息	109
5.38	NET_DVR_IPCHANINFO:IP 通道参数	110
5.39	NET_DVR_IPDEVINFO_V31:IP 设备信息	110
5.40	NET_DVR_IPPARACFG_V40:IP 接入参数	111
5.41	NET_DVR_JPEGPARA:JPEG 图像参数	112
5.42	NET_DVR_MOTION_V30:移动侦测参数	113
5.43	NET_DVR_MULTI_STREAM_COMPRESSIONCFG:多码流压缩参数配置	114
5.44	NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND:多码流压缩参数配置条件	114
5.45	NET_DVR_NETCFG_V30:网络参数	115
5.46	NET_DVR_NTPPARA:NTP 参数	116
5.47	NET_DVR_PLAYCOND: 回放或者下载信息结构体	117
5.48	NET_DVR_PICCFG_V30:图像参数	117
5.49	NET_DVR_FIND_PICTURE_V40: 图片查找结果信息结构体	119
5.50	NET_DVR_PLATE_INFO:车牌信息	121
5.51	NET_DVR_PLATE_RESULT:车辆识别结果	122
5.52	NET_DVR_POINT_FRAME:云台图像区域位置信息	125
5.53	NET_DVR_PPPOECFG:PPPoE 配置参数	125
5.54	NET_DVR_PRESET_NAME:单个预置点名称	126
5.55	NET_DVR_PRESET_NAME_ARRAY:预置点名称参数	126
5.56	NET_DVR_PREVIEWINFO:预览参数	127
5.57	NET_DVR_PTZCFG:云台协议配置	128
5.58	NET_DVR_PTZ_PROTOCOL:云台协议信息	128
5.59	NET_DVR_QUERY_COUNTRYID_COND:按国家编号查询服务器信息条件参数	128
5.60	NET_DVR_QUERY_COUNTRYID_RET:按国家编号查询结果	129
5.61	NET_DVR_QUERY_DDNS_COND:HIDDNS 查询或诊断条件	129
5.62	NET_DVR_QUERY_DDNS_RET:HIDDNS 查询结果	130
5.63	NET_DVR_CHECK_DDNS_RET:HIDDNS 诊断结果	130
5.64	NET_DVR_QUERY_IPSERVER_COND:IPServer 查询条件	131
5.65	NET_DVR_QUERY_IPSERVER_RET:IPServer 查询结果	131
5.66	NET_DVR_RECORDDAY:全体录像参数	131
5.67	NET_DVR_RECORDSCHED:时间段录像参数	132
5.68	NET_DVR_RECORD_V30:录像参数	132
5.69	NET_DVR_RESOLVE_DEVICEINFO:域名解析设备信息	134
5.70	NET_DVR_SCHEDTIME:时间段参数	134
5.71	NET_DVR_SDKLOCAL_CFG:SDK 本地参数	135
5.72	NET_DVR_SEARCH_EVENT_PARAM:按事件搜索的条件参数	135
5.73	NET_DVR_SEARCH_EVENT_RET:按事件搜索结果信息	136
5.74	NET_DVR_SEND_PARAM_IN:数据发送输入参数结构体	139
5.75	NET_DVR_SERIALSTART_V40:串口参数子类	140

5.76	NET_DVR_SERIAL_COND:串口参数基类	140
5.77	NET_DVR_SETUPALARM_PARAM:报警布防参数	140
5.78	NET_DVR_SHELTER:隐私遮盖参数	142
5.79	NET_DVR_SHOWSTRINGINFO:单个字符参数	143
5.80	NET_DVR_SHOWSTRING_V30:字符叠加参数	143
5.81	NET_DVR_STREAM_INFO:流 ID 信息	144
5.82	NET_DVR_SINGLE_DDNS: DDNS 服务器信息	144
5.83	NET_DVR_TIME:时间参数	145
5.84	NET_DVR_TIME_EX:时间参数	146
5.85	NET_DVR_TIME_V30:时间参数	146
5.86	NET_DVR_UPLOAD_FILE_RET:文件上传结果信息结构体	147
5.87	NET_DVR_UPNP_NAT_STATE:Upnp 端口映射状态	148
5.88	NET_DVR_UPNP_PORT_STATE:Upnp 端口映射状态	148
5.89	NET_DVR_USER_INFO_V30:单个用户信息参数	149
5.90	NET_DVR_USER_V30:用户参数	151
5.91	NET_DVR_VEHICLE_INFO:车辆信息参数	151
5.92	NET_DVR_VICOLOR:时间段图像参数	153
5.93	NET_DVR_VIDEOEFFECT:视频显示参数	153
5.94	NET_DVR_VILOST_V30:视频丢失参数	154
5.95	NET_DVR_VOD_PARA:录像回放参数	154
5.96	NET_DVR_WIFIETHERNET:无线网口参数	155
5.97	NET_DVR_WIFI_CFG:wifi 配置参数	156
5.98	NET_DVR_WIFI_CONNECT_STATUS:wifi 连接状态	157
5.99	NET_DVR_WORKSTATE_V30:设备工作状态参数	157
5.100	NET_DVR_ZEROCHANCFG:零通道压缩参数	158
5.101	NET_IPC_AUX_ALARMCFG:辅助报警参数	159
5.102	NET_IPC_CALLHELP_ALARMCFG:呼救报警参数	159
5.103	NET_IPC_PIR_ALARMCFG:PIR 报警参数	160
5.104	NET_IPC_SINGLE_AUX_ALARMCFG:单个辅助报警参数	160
5.105	NET_IPC_SINGLE_WIRELESS_ALARMCFG:单个无线报警参数	161
5.106	NET_ITS_PICTURE_INFO:抓拍图片信息	162
5.107	NET_ITS_PLATE_RESULT:车辆识别结果	164
5.108	NET_VCA_ADV_REACH_HEIGHT:折线攀高参数	167
5.109	NET_VCA_ADV_TRAVERSE_PLANE:折线警戒面参数	168
5.110	NET_VCA_AREA:区域参数	169
5.111	NET_VCA_AUDIO_ABNORMAL:音频异常参数	169
5.112	NET_VCA_BLACKLIST_INFO:授权名单信息结构体	170
5.113	NET_VCA_BLACKLIST_INFO_ALARM:授权名单报警信息	171
5.114	NET_VCA_COMBINED_RULE:组合规则参数	172
5.115	NET_VCA_DEV_INFO:前端设备信息	173
5.116	NET_VCA_EVENT_UNION:事件参数联合体	174
5.117	NET_VCA_FACESNAP_INFO_ALARM:人脸抓拍信息	176
5.118	NET_VCA_FACESNAP_RESULT:人脸抓拍结果	177
5.119	NET_VCA_FACESNAP_MATCH_ALARM:授权名单比对结果	179

5.120	NET_VCA_FALL_DOWN:倒地参数.....	180
5.121	NET_VCA_GET_UP:起身参数.....	181
5.122	NET_VCA_HIGH_DENSITY:人员聚集参数.....	182
5.123	NET_VCA_HUMAN_ATTRIBUTE:人员信息结构体.....	182
5.124	NET_VCA_HIGH_DENSITY_STATUS:人员聚集状态信息.....	183
5.125	NET_VCA_HUMAN_ENTER:人员进入参数.....	184
5.126	NET_VCA_HUMAN_FEATURE:人体属性参数结构体.....	184
5.127	NET_VCA_INTRUSION:区域入侵参数.....	186
5.128	NET_VCA_LEAVE_POSITION:离岗事件参数.....	186
5.129	NET_VCA_LEFT:丢包参数.....	187
5.130	NET_VCA_LINE:线结构参数.....	188
5.131	NET_VCA_LOITER:徘徊参数.....	188
5.132	NET_VCA_OVER_TIME:操作超时参数.....	189
5.133	NET_VCA_PARKING:停车参数.....	190
5.134	NET_VCA_PEOPLENUM_CHANGE:人数变化参数.....	190
5.135	NET_VCA_POINT:点坐标参数.....	191
5.136	NET_VCA_POLYGON:多边形参数.....	192
5.137	NET_VCA_REACH_HIGHT:攀高参数.....	192
5.138	NET_VCA_RECT:区域框参数.....	193
5.139	NET_VCA_RELATE_RULE_PARAM:关联规则参数.....	193
5.140	NET_VCA_RULE_ALARM:异常行为识别报警.....	194
5.141	NET_VCA_RULE_INFO:事件规则信息.....	196
5.142	NET_VCA_RUN:快速移动参数.....	198
5.143	NET_VCA_SCANNER:读卡器参数.....	199
5.144	NET_VCA_SPACING_CHANGE:间距变化参数.....	199
5.145	NET_VCA_STANDUP:起立参数.....	200
5.146	NET_VCA_STICK_UP:贴纸条参数.....	201
5.147	NET_VCA_TAKE:捡包参数.....	202
5.148	NET_VCA_TAKE_LEFT:丢包/捡包参数.....	202
5.149	NET_VCA_TARGET_INFO:报警目标信息.....	203
5.150	NET_VCA_TOILET_TARRY:如厕超时参数.....	203
5.151	NET_VCA_TRAIL:尾随参数.....	204
5.152	NET_VCA_TRAVERSE_PLANE:穿越警戒面参数.....	205
5.153	NET_VCA_VIOLENT_MOTION:剧烈运动参数.....	206
5.154	NET_VCA_YARD_TARRY:放风场滞留参数.....	206
5.155	struAlarmChannel:报警通道参数.....	207
5.156	struAlarmHardDisk:报警硬盘参数.....	207
5.157	struIOAlarm:报警输入输出信息.....	208
5.158	struRecordingHost:录播主机专用报警参数.....	208
5.159	uStruAlarm:报警信息联合体.....	209
5.160	WEP:wep 加密参数.....	210
5.161	WPA_PSK:WPA_PSK 加密参数.....	210
5.162	WPA_WPA2:WPA_WPA2 加密参数.....	211
5.163	国家编号说明.....	211

1 SDK 简介

1.1 SDK 概述

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。

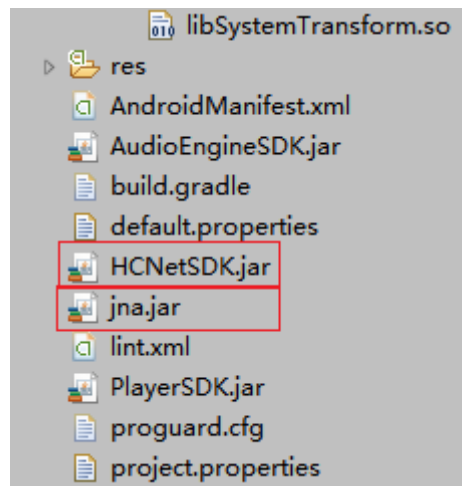
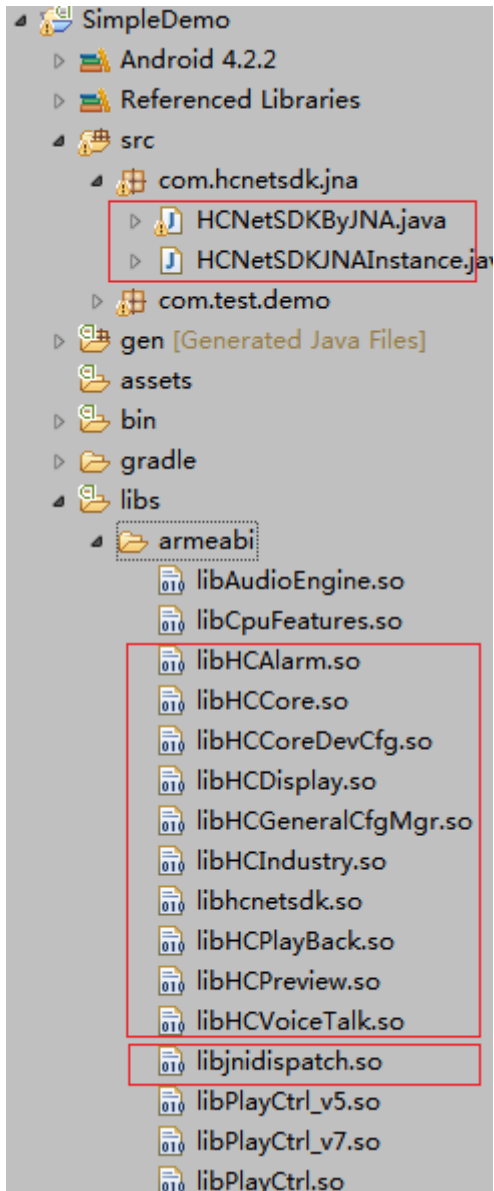
我司 Android 设备网络 SDK 在 V5.2.5.0 版本之前都是 JNI 接口，为了简化本地方法调用过程，对外接口与 Windows、Linux 版本网络 SDK 保持统一，在该版本及之后版本新增了 JNA 方式，在 HCNetSDKByJNA.java 文件中增加动态库支持接口的 Java 定义，即可以直接调用该 C++ 接口。因此，目前 Android 版本网络库包含 JNI 接口和 JNA 接口，调用 JNA 接口之前必须先调用 JNI 接口里面的初始化、登录等相关接口。

Android 设备网络 SDK 包含的主要文件如下所示：

表 1.1 开发包文件说明

库接口	文件名	描述	文件路径
C++ 的 so 动态库 (JNI 和 JNA 接口)	libhcnetsdk.so	网络 SDK 外部接口	加载在工程的 libs/armeabi 文件夹下
	libHCCore.so	核心组件	
	libHCCoreDevCfg.so	设备配置核心组件	
	libHCPreview.so	预览组件	
	libHCAAlarm.so	报警组件	
	libHCPlayBack.so	回放组件	
	libHCVoiceTalk.so	语音组件	
	libHCGeneralCfgMgr.so	维护管理配置组件	
	libHCIndustry.so	行业应用管理配置组件	
	libHCDisplay.so	显示组件	
JNI	HCNetSDK.jar	JNI 接口的 JAR 包	加载在工程目录下
JNA	libjnidispatch.so	JNA 的库文件，与 jna.jar 配套使用	加载在工程的 libs/armeabi 文件夹下
	jna.jar	JNA 的 JAR 包	加载在工程目录下
	HCNetSDKByJNA.java	用于 jna 下定义接口、结构体和宏定义，作为源码文件加到工程，可以修改包名 “package com.hcnetsdk.jna”	
	HCNetSDKJNAInstance.java	用于实现 jna 方式加载动态库，作为源码文件加到工程，可以修改包名 “package com.hcnetsdk.jna”	

Android SDK 开发包 Demo 的工程结构如下所示：



1.2 适用范围

环境要求

Android V4.0 及以上版本

主要功能

实时码流预览、抓图、云台控制、录像文件查找和回放、报警布防等。

适用于但不仅限于以下产品型号：

- 编解码设备

DVR: DS-9100、DS-8100、DS-8000-S、DS-8800、DS-7800、DS-7300、DS-7200、DS-7100、DS-7000 等系列，包括-ST、-SH、-SE、-SN、-RT、-RH、-XT 等。

NVR: DS-96000N(-F24/-F16)(/H)(/I)、DS-96000N(-H24/-H16)(/H)(/I)、DS-9600N-I8/H8/F8/ST/XT、DS-8600N-I8/H8/F8/E8/ST/XT、DS-7800N-E1/SN/SNH、DS-7600N-ST/E2/E1、DS-7700N-ST/E4、DS-9500N-ST、DS-9500N-S、DS-9600N-SH、DS-7600N-S、DS-9664N-RX 等。

HDVR(混合型 DVR): DS-9000、DS-8000-ST、DS-7600H-ST/-S 系列等。

编码器: DS-6700、DS-6600、DS-6500(-JX)、DS-6100、DS-6401HFH 系列视频服务器、DS-6000 系列编/解码器等。

- 网络摄像机, 网络球机

网络摄像机: 标清、高清、红外、热成像、鱼眼等, 如 DS-2CD7xx、DS-2CD71xx、DS-2CD72xx、DS-2CD8xx、DS-2CD81xx、DS-2CD82xx、DS-2CD84xx、DS-2CD83xx、DS-2CD11xx、DS-2CD12xx、DS-2CD13xx、DS-2CD20xx、DS-2CD21xx、DS-2CD22xx、DS-2CD23xx、DS-2CD24xx、DS-2CD25xx、DS-2CD26xx、DS-2CD27xx、DS-2CD28xx、DS-2CD29xx、DS-2CD2Axx、DS-2CD2Cxx、DS-2CD2Dxx、DS-2CD2Txx、DS-2CD2Qxx、DS-2CD30xx、DS-2CD31xx、DS-2CD32xx、DS-2CD33xx、DS-2CD34xx、DS-2CD39xx、DS-2CD3Txx、DS-2CD3Qxx、DS-2CD40xx、DS-2CD41xx、DS-2CD42xx、DS-2CD4Axx、DS-2CD62xx、DS-2CD63xx、DS-2CD64xx、DS-2CD65xx 等。

网络球机: 标清、高清、红外等, 如 DS-2DF86xx、DS-2DF85xx、DS-2DF82xx、DS-2DF72xx、DS-2DF71xx、DS-2DE71xx、DS-2DE73xx、DS-2DE72xx、DS-2DM72xx、DS-2DM71xx、DS-2DF1-7xx、DS-2DF66xx、DS-2DF62xx、DS-2DF1-6xx、DS-2DE51xx、DS-2DE52xx、DS-2DE53xx、DS-2DM52xx、DS-2DF52xx、DS-2DC52xx、DS-2DC51xx、DS-2DF1-5xx、DS-2DE45xx、DS-2DE42xx、DS-2DE41xx、DS-2DF1-4xx、DS-2DM1-7xx、DS-2DM1-6xx、DS-2DM1-5xx 等。

一体化网络摄像机: DS-2ZCN3007、DS-2ZCN3006、DS-2DZ216MF、DS-2DZ2116、DS-2ZCN2006、DS-2ZCN2007、DS-2ZMN2007、DS-2ZMN2006 等。

智能交通摄像机(抓拍机): (i)DS-2CD93xx、(i)DS-2CD92xx、(i)DS-2CD91xx、DS-2CD9xx 等系列。

2 SDK 版本更新

Version 5.3 (build20171014)

- Demo 支持内外部两种解码方式供外部客户参考；
- Android Demo 修改 UI 界面，增加按钮控件加以区分，分别对应内部解码显示、外部回调解码预览。
- Demo 同时支持 Android Studio 和 Eclipse 两种开发环境；
- Demo 支持窗口切换功能(从主 Activity 切换到另一个 Activity)；
- Android Demo 界面增加切换按钮，支持在预览时切换到其他 Activity，并在画面切回时预览正常，不出现卡顿或是黑屏等现象。
- Android SDK 支持 Android8.0；
- 发布 armeabi、armeabi-v7a、arm64-v8a 三个结构的 Android SDK 版本

Version 5.2.5.2 (build20160715)

- 采用了 JNA 方式直接调用 C++动态库接口，相关接口和结构体均基于 HCNetSDKByJNA 类。
- 报警布防接口更新为 JNA 接口（原有 JNI 接口仍支持）：
[NET_DVR_SetDVRMessageCallBack_V30](#)、
[NET_DVR_SetupAlarmChan_V41](#)、
[NET_DVR_CloseAlarmChan_V30](#)。
- 新增报警类型：
移动侦测、视频丢失、遮挡、IO 信号量等报警信息，报警数据为可变长：COMM_ALARM_V40；
异常行为识别报警，包括区域入侵、越界侦测等：COMM_ALARM_RULE；
车牌识别抓拍上传：COMM_UPLOAD_PLATE_RESULT、COMM_ITS_PLATE_RESULT。
- 新增多码流压缩参数（对应接口：[NET_DVR_GetDeviceConfig](#)、[NET_DVR_SetDeviceConfig](#)）：
命令：NET_DVR_GET_MULTI_STREAM_COMPRESSIONCFG、
NET_DVR_SET_MULTI_STREAM_COMPRESSIONCFG。

Version 5.1.3.2 (build20150605)

- 新增动态 IP 和端口解析接口：
[NET_DVR_GetAddrInfoByServer](#)。
- 新增设备激活功能接口：
[NET_DVR_ActivateDevice](#)。
- 新增参数配置功能（对应接口：[NET_DVR_GetDVRConfig](#)）：
NET_DVR_GET_DIGITAL_CHANNEL_STATE、NET_DVR_GET_PRESET_NAME。
- 新增即时刷新录像索引接口：
[NET_DVR_UpdateRecordIndex](#)。
- 新增事件录像查找接口：
[NET_DVR_FindFileByEvent](#)、[NET_DVR_FindNextEvent](#)。
- 新增透明通道扩展接口：
[NET_DVR_SerialStart_V40](#)。
- 新增远程控制面板接口：
[NET_DVR_ClickKey](#)。

3 函数说明

3.1 SDK 初始化

3.1.1 初始化 SDK **NET_DVR_Init**

函 数: public boolean NET_DVR_Init()

参 数: 无

返回值: TRUE 表示成功, FALSE 表示失败。

说 明: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。调用设备网络 SDK 其他函数的前提。

[返回目录](#)

3.1.2 释放 SDK 资源 **NET_DVR_Cleanup**

函 数: public boolean NET_DVR_Cleanup()

参 数: 无

返回值: TRUE 表示成功, FALSE 表示失败。

说 明: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。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](#)

3.2 SDK 本地功能

本地参数配置

3.2.1 获取 SDK 本地参数 **NET_DVR_GetSDKLocalConfig**

函 数: public boolean NET_DVR_GetSDKLocalConfig(NET_DVR_SDKLOCAL_CFG lpSdkCfg)

参 数: [in] lpSdkCfg 本地配置参数, 详见: [NET_DVR_SDKLOCAL_CFG](#)

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.2.2 设置 SDK 本地参数 **NET_DVR_SetSDKLocalConfig**

函 数: public boolean NET_DVR_SetSDKLocalConfig(NET_DVR_SDKLOCAL_CFG lpSdkCfg)

参 数: [in] lpSdkCfg 本地配置参数, 详见: [NET_DVR_SDKLOCAL_CFG](#)

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

连接和接收超时时间及重连设置

3.2.3 设置网络连接超时时间 **NET_DVR_SetConnectTime**

函 数: public boolean NET_DVR_SetConnectTime(int iWaitTime)

参 数: [in] iWaitTime 超时时间, 单位毫秒, 取值范围[300,60000], 实际最大超时时间因系统的 connect 超时时间而不同。

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。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](#)

3.2.4 设置重连功能 **NET_DVR_SetReconnect**

函 数: public boolean NET_DVR_SetReconnect(int iInterval, boolean bEnableRecon)

参 数: [in] iInterval 重连间隔, 单位:毫秒

[in] bEnableRecon 是否重连, false -不重连, true -重连, 参数默认为 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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。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](#)

3.2.5 设置接收超时时间 **NET_DVR_SetRecvTimeOut**

函 数: public boolean NET_DVR_SetRecvTimeOut(int 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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。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](#)

SDK 版本信息和日志

3.2.6 获取 SDK 版本信息 **NET_DVR_GetSDKVersion**

函 数: public int NET_DVR_GetSDKVersion()

参 数:

返回值: 获取 SDK 版本信息。

说 明: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。SDK 版本信息, 2 个高字节表示主版本, 2 个低字节表示次版本。如 0x00030000: 表示版本为 3.0。

[返回目录](#)

3.2.7 获取 SDK 的版本号和 build 信息 **NET_DVR_GetSDKBuildVersion**

函 数: public int NET_DVR_GetSDKBuildVersion()

参 数:

返回值: 获取 SDK 的版本号和 build 信息。

说 明: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。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](#)

3.2.8 启用写日志文件 **NET_DVR_SetLogToFile**

函 数: public boolean NET_DVR_SetLogToFile(int bLogEnable, String strLogDir, boolean 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

[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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。日志文件路径必须是绝对路径, 且以 "/" 结尾。当设置了删除超出的文件时 (即 bAutoDel 为 TRUE), 那么将会自动删除超出的文件。更改目录时到下一次写文件时才会使用新的目录写文件。

[返回目录](#)

异常消息回调

3.2.9 注册接收异常、重连消息回调函数 **NET_DVR_SetExceptionCallBack**

函 数: public boolean NET_DVR_SetExceptionCallBack(ExceptionCallBack CallBack)

参 数: [in] CallBack 接收异常消息的回调函数，回调当前异常的相关信息

```
public interface ExceptionCallBack {  
    public void fExceptionCallBack(int iType, int iUserID, int iHandle);  
}
```

[out] iType 异常或重连等消息的类型，详见表 3.1

[out] iUserID 登录 ID

[out] iHandle 出现异常的相应类型的句柄

表 3.1 异常消息类型

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

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

说 明: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

获取错误信息

3.2.10 返回最后操作的错误码 **NET_DVR_GetLastError**

函 数: public int 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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。返回值为错误码。

[返回目录](#)

3.2.11 返回最后操作的错误信息 **NET_DVR_GetErrorMsg**

函 数: public String NET_DVR_GetErrorMsg(INT_PTR pErrNo);

参 数: [out] pErrNo 错误码

返回值: 返回值为错误描述信息。

说 明: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.3 用户注册

3.3.1 设备的动态 IP 和端口解析 **NET_DVR_GetAddrInfoByServer**

函 数: public boolean NET_DVR_GetAddrInfoByServer(int dwQueryType, NET_DVR_ADDR_QUERY_COND pCond, NET_DVR_ADDR_QUERY_RET pRet)

参 数: [in] dwQueryType 查找类型, 取值见 ADDR_QUERY_TYPE, 对应关系见表 3.2

[in] pCond 查找条件, 对应关系见表 3.2

[out] pRet 查找结果, 对应关系见表 3.2

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。不同的查询类型对应不同的结构体和查找结果, 对应关系见表 3.2。

表 3.2 域名解析类型

dwQueryType 定义	含义	pCond 对应类	pRet 对应类
QUERYSVR_BY_COUNTRYID	按国家编号查询服务器地址	NET_DVR_QUERY_COUNTRYID_COND	NET_DVR_QUERY_COUNTRYID_RET
QUERYDEV_BY_NICKNAME_DDNS	按设备昵称从 hiddns 查询设备信息	NET_DVR_QUERY_DDNS_COND	NET_DVR_QUERY_DDNS_RET
QUERYDEV_BY_SERIAL_DDNS	按序列号从 hiddns 查询设备信息	NET_DVR_QUERY_DDNS_COND	NET_DVR_QUERY_DDNS_RET
CHECKDEV_BY_NICKNAME_DDNS	按设备昵称从 hiddns 诊断设备	NET_DVR_QUERY_DDNS_COND	NET_DVR_CHECK_DDNS_RET

CHECKDEV_BY_SERIAL_DDNS	按序列号从 hiddns 诊断设备	NET_DVR_QUERY_DDNS_COND	NET_DVR_CHECK_DDNS_RET
QUERYDEV_BY_NICKNAME_IPSERVER	按设备昵称从 IPServer 查询设备信息	NET_DVR_QUERY_IPSERVER_COND	NET_DVR_QUERY_IPSERVER_RET
QUERYDEV_BY_SERIAL_IPSERVER	按序列号从 IPServer 查询设备信息	NET_DVR_QUERY_IPSERVER_COND	NET_DVR_QUERY_IPSERVER_RET

[返回目录](#)

3.3.2 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： public boolean NET_DVR_GetDVRIPByResolveSvr_EX(String sServerIP, short wServerPort, String sDVRName, short wDVRNameLen, String sDVRSerialNumber, short wDVRSerialLen, NET_DVR_RESOLVE_DEVICEINFO lpDeviceInfo)

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，EasyDDNS 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] lpDeviceInfo 获取到的设备 IP 地址、端口等信息，详见：
[NET_DVR_RESOLVE_DEVICEINFO](#)

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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。通过该接口根据域名或者设备序列号解析出设备当前 IP 地址和端口号，然后调用 [NET_DVR_Login_V30](#) 登录设备。该接口中的设备名称和设备序列号不能同时为空。IPServer 是我公司提供的一款域名解析服务器软件。

[返回目录](#)

3.3.3 激活设备 NET_DVR_ActivateDevice

函数： public boolean NET_DVR_ActivateDevice(String sDvrIp, int iDvrPort, NET_DVR_ACTIVATECFG lpActivateCfg)

参数： [in] sDvrIp 设备地址
[in] iDvrPort 设备端口号
[in] lpActivateCfg 激活参数，详见：[NET_DVR_ACTIVATECFG](#)

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

说明： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.3.4 用户注册设备 **NET_DVR_Login_V30**

函 数： public int NET_DVR_Login_V30(String sDvrIp, int iDvrPort, java.lang.String sUserName, String sPassword, NET_DVR_DEVICEINFO_V30 DeviceInfo)

参 数：

[in] sDvrIp	设备 IP 地址或静态域名
[in] iDvrPort	设备端口号
[in] sUserName	登录的用户名
[in] sPassword	用户密码
[out] DeviceInfo	设备信息，详见： NET_DVR_DEVICEINFO_V30

返回值： -1 表示失败，其他值表示返回的用户 ID 值。该用户 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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** SDK 注册设备新增支持静态域名的方式，即可设置 sDVRIP="test.vicp.net"。

[返回目录](#)

3.3.5 用户注销 **NET_DVR_Logout_V30**

函 数： public boolean NET_DVR_Logout_V30 (int IUserID)

参 数： [in]IUserID 用户 ID 号，NET_DVR_Login_V30 的返回值

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

说 明： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

3.4 获取设备能力集

3.4.1 获取设备能力集 **NET_DVR_GetXMLAbility**

函 数： public boolean NET_DVR_GetXMLAbility(int IUserID, int dwAbilityType, byte[] pInBuf, int dwInBufLen, byte[] pOutBuf, int dwOutBufLen, INT_PTR lpSizeReturned)

参 数：

[in] IUserID	NET_DVR_Login_V30 的返回值
[in] dwAbilityType	能力集类型，详见表 3.3
[in] pInBuf	输入缓冲区，不同的能力集对应不同的输入内容，详见表 3.3
[in] dwInLength	输入缓冲区的长度
[out] pOutBuf	输出缓冲区，不同的能力集对应不同的输出内容，详见表 3.3
[in] dwOutLength	接收数据的缓冲区的长度
[out] lpSizeReturned	pOutBuf 实际有效长度

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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 获取设备能力集时，需要输入参数和输出参

数的格式定义如表 3.3 所示。

表 3.3 能力集类型

能力类型宏定义	能力类型说明	pInBuf	pOutBuf
DEVICE_SOFTHARDWARE_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

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

[返回目录](#)

3.5 实时预览

强制 I 帧

3.5.1 主码流动态产生一个关键帧 NET_DVR_MakeKeyFrame

函 数： public boolean NET_DVR_MakeKeyFrame(int IUserID, int IChannel)

参 数： [in] IUserID NET_DVR_Login_V30 的返回值
[in] IChannel 通道号，模拟通道从 1 开始，IP 通道一般从 33 开始

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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。此接口用于重置 I 帧，根据设置的预览参数 ([NET_DVR_PREVIEWINFO](#)) 为主码流或者子码流分别调用 NET_DVR_MakeKeyFrame 或者 [NET_DVR_MakeKeyFrameSub](#) 实现重置 I 帧。

[返回目录](#)

3.5.2 子码流动态产生一个关键帧 NET_DVR_MakeKeyFrameSub

函 数： public boolean NET_DVR_MakeKeyFrameSub(int IUserID, int IChannel)

参 数： [in] IUserID NET_DVR_Login_V30 的返回值
[in] IChannel 通道号，模拟通道从 1 开始，IP 通道一般从 33 开始

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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。此接口用于重置 I 帧，根据设置的预览参数 ([NET_DVR_PREVIEWINFO](#)) 为主码流或者子码流分别调用 [NET_DVR_MakeKeyFrame](#) 或者 NET_DVR_MakeKeyFrameSub 实现重置 I 帧。

[返回目录](#)

实时预览

3.5.3 实时预览 **NET_DVR_RealPlay_V40**

函 数: public int NET_DVR_RealPlay_V40(int IUserID, NET_DVR_PREVIEWINFO previewInfo, RealPlayCallBack CallBack)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] previewInfo 预览参数，包括码流类型、取流协议、通道号等，详见：

[NET_DVR_PREVIEWINFO](#)

[in] CallBack 码流数据回调函数

```
public interface RealPlayCallBack {  
    public void fRealDataCallBack(int iRealHandle, int iDataType, byte[] pDataBuffer, int iDataSize);  
}
```

[out] iRealHandle 当前的预览句柄

[out] iDataType 数据类型

[out] pDataBuffer 存放数据的缓冲区指针

[out] iDataSize 缓冲区大小

表 3.4 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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。通过该接口设置实时流回调函数获取实时流音视频数据，然后可以通过播放库进行解码显示。

[返回目录](#)

3.5.4 停止预览 **NET_DVR_StopRealPlay**

函 数: public boolean NET_DVR_StopRealPlay(int 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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.5.5 获取预览时用来解码和显示的播放器句柄 **NET_DVR_GetRealPlayerIndex**

函 数: public int NET_DVR_GetRealPlayerIndex(int 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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

播放声音控制

3.5.6 播放声音 NET_DVR_OpenSound

函 数: public boolean NET_DVR_OpenSound(int IRealHandle)
参 数: [in] IRealHandle NET_DVR_RealPlay 或 NET_DVR_RealPlay_V30 的返回值
返回值: 0 表示失败, 1 表示成功。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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

3.5.7 关闭声音 NET_DVR_CloseSound

函 数: public boolean NET_DVR_CloseSound()
参 数:
返回值: 0 表示失败, 1 表示成功。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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

3.5.8 调节播放音量 NET_DVR_Volume

函 数: public boolean NET_DVR_Volume(int IRealHandle, short wVolume)
参 数: [in] IRealHandle NET_DVR_RealPlay 或 NET_DVR_RealPlay_V30 的返回值
[in] wVolume 音量, 取值范围[0,0xffff]
返回值: 0 表示失败, 1 表示成功。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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

显示参数配置

3.5.9 获取预览视频显示参数 **NET_DVR_ClientGetVideoEffect**

函 数： public boolean NET_DVR_ClientGetVideoEffect(int IRealHandle, NET_DVR_VIDEOEFFECT VideoEffect)
参 数： [in] IRealHandle 预览句柄，NET_DVR_RealPlay_V40 的返回值
 [out] VideoEffect 显示参数，详见：[NET_DVR_VIDEOEFFECT](#)
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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 该接口需要在预览的前提下才能调用，但是获取的视频参数均为设备返回，是当前预览所对应通道的亮度、对比度等参数。

[返回目录](#)

3.5.10 设置预览视频显示参数 **NET_DVR_ClientSetVideoEffect**

函 数： public boolean NET_DVR_ClientSetVideoEffect(int IRealHandle, NET_DVR_VIDEOEFFECT VideoEffect)
参 数： [in] IRealHandle 预览句柄，NET_DVR_RealPlay_V40 的返回值
 [in] VideoEffect 显示参数，详见：[NET_DVR_VIDEOEFFECT](#)
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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 该接口需要在预览的前提下才能调用，设置设备上对应通道的视频参数(远程配置设备上的参数，其他客户端相同通道的预览图像也会随之变化)。SDK4.0 及以后版本中，当设置的亮度、对比度等的值超过或低于限制值时接口不返回失败，将取最接近的上下限制值作为实际的参数值。

[返回目录](#)

零通道预览

3.5.11 开始零通道预览 **NET_DVR_ZeroStartPlay**

函 数： public int NET_DVR_ZeroStartPlay(int IUserID, NET_DVR_CLIENTINFO ClientInfo, RealPlayCallBack
 Callback, boolean bBlock)
参 数： [in] IUserID NET_DVR_Login_V30 的返回值
 [in] ClientInfo 预览参数，详见：[NET_DVR_CLIENTINFO](#)
 [in] Callback 码流数据回调函数
 [in] bBlock 请求码流过程是否阻塞：0- 否，1- 是

```
public interface RealPlayCallBack {
    public void fRealDataCallBack(int iRealHandle, int iDataType, byte[] pDataBuffer, int iDataSize);
}
```

 [out] iRealHandle 当前的预览句柄

[out] iDataType	数据类型
[out] pDataBuffer	存放数据的缓冲区指针
[out] iDataSize	缓冲区大小

表 3.5 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_ZeroStopPlay 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： In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。该接口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。若设为阻塞，表示直到播放操作完成才返回成功与否。

[返回目录](#)

3.5.12 停止零通道预览 NET_DVR_ZeroStopPlay

函数： `public boolean NET_DVR_ZeroStopPlay(int iRealHandle)`

参数： [in] iRealHandle 预览句柄，NET_DVR_ZeroStartPlay 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： In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。

[返回目录](#)

预览抓图

3.5.13 设置抓图模式 NET_DVR_SetCapturePictureMode

函数： `public boolean NET_DVR_SetCapturePictureMode(int dwCaptureMode)`

参数： [in] dwCaptureMode 抓图模式

```
enum tagPDC_PARAM_KEY{
    BMP_MODE           = 0,
    JPEG_MODE          = 1
} CAPTURE_MODE
```

BMP_MODE

BMP 模式

JPEG_MODE

JPEG 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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.5.14 预览时抓图并保存成图片文件 NET_DVR_CapturePictureBlock

函数： public boolean NET_DVR_CapturePictureBlock(int iRealHandle, String sPicFileName, int dwTimeOut)

参数： [in] iRealHandle 预览句柄, NET_DVR_RealPlay_V40 的返回值
[in] sPicFileName 保存图片的文件路径, 包含文件名, 比如: "D:/test.bmp"或
"D:/test.jpg"
[in] dwTimeOut 超时时间, 目前无效

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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.5.15 单帧数据捕获并保存成图片 NET_DVR_CapturePicture

函数： public boolean NET_DVR_CapturePicture(int iRealHandle, String sPicFileName)

参数： [in] iRealHandle 预览句柄, NET_DVR_RealPlay_V40 的返回值
[in] sPicFileName 保存图片的文件路径, 包含文件名, 比如: "D:/test.bmp"或
"D:/test.jpg"

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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

客户端录像

3.5.16 捕获预览数据并保存到指定文件中 NET_DVR_SaveRealData

函数： public boolean NET_DVR_SaveRealData(int IRealHandle, String sFileName)

参数： [in] IRealHandle 预览句柄, NET_DVR_RealPlay_V40 的返回值
[in] sFileName 文件路径名称

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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。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](#)

3.5.17 捕获预览数据并保存到指定文件中 **NET_DVR_SaveRealData_V30**

函 数: public boolean NET_DVR_SaveRealData_V30(int IRealHandle, int dwTransType, String sFileName)

参 数: [in] IRealHandle 预览句柄, NET_DVR_RealPlay_V40 的返回值

[in] dwTransType 存储的码流封装格式, 见 STREAM_TYPE 定义

```
enum STREAM_TYPE {
    STREAM_PS        = 0x1,
    STREAM_3GPP      = 0x2
} CAPTURE_MODE
```

STREAM_PS

PS 文件层, 主要用于存储, 也可用于传输

STREAM_3GPP

3GPP 文件层, 用于存储

[in] sFileName 文件路径名称

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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**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](#)

3.5.18 停止数据捕获 **NET_DVR_StopSaveRealData**

函 数: public boolean NET_DVR_StopSaveRealData(int 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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

实时数据回调

3.5.19 实时预览 **NET_DVR_SetRealDataCallBack**

函 数: public native boolean NET_DVR_SetRealDataCallBack(int IRealHandle, RealDataCallBack cbRealDataCallBack)

参 数: [in] IUserID NET_DVR_RealPlay 或 NET_DVR_RealPlay_V30 的返回值

[in] CallBack 码流数据回调函数

```
public interface RealPlayCallBack {
    public void fRealDataCallBack(int iRealHandle, int iDataType, byte[] pDataBuffer, int iDataSize);
}
```

}

[out] IRealHandle 当前的预览句柄
 [out] IDataType 数据类型
 [out] pDataBuffer 存放数据的缓冲区指针
 [out] IDataSize 缓冲区大小

表 3.6 码流数据类型

dwDataType 宏定义	宏定义值	含义
NET_DVR_SYSHEAD	1	系统头数据
NET_DVR_STREAMDATA	2	流数据（包括复合流或音视频分开的视频流数据）
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：In class **com.hikvision.netsdk.HCNetSDK**, **JNI 接口**。通过该接口设置实时流回调函数获取实时流音视频数据，然后可以通过播放库进行解码显示。

[返回目录](#)

3.5.20 捕获实时码流数据（标准码流）NET_DVR_RealPlay_V40

函数：public native boolean NET_DVR_SetStandardDataCallBack(int IRealHandle, StdDataCallBack cbStdDataCallBack);

参数：[in] IUserID NET_DVR_RealPlay 或 NET_DVR_RealPlay_V30 的返回值
 [in] CallBack 码流数据回调函数

```
public interface StdDataCallBack {
    public void fStdDataCallback(int IRealHandle, int dwDataType, byte[] pBuffer, int dwBufSize);
}
```

[out] IRealHandle 当前的预览句柄
 [out] IDataType 数据类型
 [out] pBuffer 存放数据的缓冲区指针
 [out] dwBufSize 缓冲区大小

表 3.7 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_STD_VIDEODATA	4	标准视频流数据(标准 264、MPEG4)
NET_DVR_STD_AUDIODATA	5	标准音频流数据(G722 音频数据)
NET_DVR_PRIVATE_DATA	112	私有数据，包括智能信息叠加等

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：In class **com.hikvision.netsdk.HCNetSDK**, **JNI 接口**。通过该接口设置实时流回调函数获取实时流音视频数据，然后可以通过播放库进行解码显示。

[返回目录](#)

3.6 设备抓图

3.6.1 单帧数据捕获并保存成 JPEG 图片 **NET_DVR_CaptureJPEGPicture**

函 数: public boolean NET_DVR_CaptureJPEGPicture(int IUserID, int IChannel, NET_DVR_JPEGPARA lpJpegPara, String sPicFileName)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] IChannel 通道号
[in] lpJpegPara JPEG 图像参数, 详见: [NET_DVR_JPEGPARA](#)
[in] sPicFileName 保存 JPEG 图的文件路径

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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**该接口用于设备的单帧数据捕获, 并保存成 JPEG 图片。IPC 设备支持当前视频分辨率的抓取。

[返回目录](#)

3.6.2 单帧数据捕获并保存成 JPEG 存放在指定的内存空间中

NET_DVR_CaptureJPEGPicture_NEW

函 数: public boolean NET_DVR_CaptureJPEGPicture_NEW(int IUserID, int IChannel, NET_DVR_JPEGPARA lpJpegPara, byte[] lpJpegPicBuffer, int dwPicSize, INT_PTR lpSizeReturned)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] IChannel 通道号
[in] lpJpegPara JPEG 图像参数, 详见: [NET_DVR_JPEGPARA](#)
[in] sJpegPicBuffer 保存 JPEG 数据的缓冲区
[in] dwPicSize 输入缓冲区大小
[out] lpSizeReturned 返回图片数据的大小

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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**该接口用于设备的单帧数据捕获, 并保存成 JPEG 图片。IPC 设备支持当前视频分辨率的抓取。

[返回目录](#)

3.7 布防、撤防

设置报警等信息上传的回调函数

3.7.1 注册回调函数，接收设备报警消息 **NET_DVR_SetDVRMessageCallBack_V30**

函 数: public boolean NET_DVR_SetDVRMessageCallBack_V30(FMSGCallBack fMessageCallBack, Pointer pUser)

参 数: [in] fMessageCallBack 报警信息回调函数
[in] pUser 用户参数

```
public interface FMSGCallBack extends Callback{  
    public void invoke(int ICommand, NET_DVR_ALARMER pAlarmer, Pointer pAlarmInfo, int  
        dwBufLen, Pointer pUser);  
}
```

[out] ICommand 上传的消息类型，详见表 3.8
[out] pAlarmer 报警设备信息，详见: [NET_DVR_ALARMER](#)
[out] pAlarmInfo 报警信息，不同的消息类型对应不同的信息结构，详见表 3.9
[out] dwBufLen 报警信息缓存大小
[out] pUser 用户数据

表 3.8 报警信息类型

ICommand 宏定义	宏定义值	含义
COMM_ALARM	0x1100	移动侦测、视频丢失、遮挡、IO 信号量等报警信息上传(V3.0 以下版本支持的设备)
COMM_ALARM_V30	0x4000	移动侦测、视频丢失、遮挡、IO 信号量等报警信息上传(V3.0 以上版本支持的设备)
COMM_ALARM_V40	0x4007	移动侦测、视频丢失、遮挡、IO 信号量等报警信息，报警数据为可变长
COMM_ALARM_RULE	0x1102	异常行为识别信息上传
COMM_UPLOAD_PLATE_RESULT	0x2800	交通抓拍结果（老报警信息，布防参数 byAlarmInfoType 设为 0）上传
COMM_ITS_PLATE_RESULT	0x3050	交通抓拍结果（新报警信息，布防参数 byAlarmInfoType 设为 1）上传
COMM_UPLOAD_FACESNAP_RESULT	0x1112	人脸抓拍报警
COMM_SNAP_MATCH_ALARM	0x2902	人脸授权名单比对结果上传

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: In class com.hcnetsdk.jna.HCNetSDKByJNA, JNA 接口。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，其关系见下表：

表 3.9 报警信息

消息类型 ICommand	上传内容	pAlarmInfo 对应的子类
COMM_ALARM	移动侦测、视频丢失、遮挡、IO 信号量等报警信息（V3.0 以下版本支持的设备）	NET_DVR_ALARMINFO
COMM_ALARM_V30	移动侦测、视频丢失、遮挡、IO 信号量等报警信息（V3.0 以上版本支持的设备）	NET_DVR_ALARMINFO_V30
COMM_ALARM_V40	移动侦测、视频丢失、遮挡、IO 信号量等报警信息，报警数据为可变量	NET_DVR_ALARMINFO_V40
COMM_ALARM_RULE	异常行为识别信息上传	NET_VCA_RULE_ALARM
COMM_UPLOAD_PLATE_RESULT	交通抓拍结果（老报警信息，布防参数 byAlarmInfoType 设为 0）上传	NET_DVR_PLATE_RESULT
COMM_ITS_PLATE_RESULT	交通抓拍结果（新报警信息，布防参数 byAlarmInfoType 设为 1）上传	NET_ITS_PLATE_RESULT
COMM_UPLOAD_FACESNAP_RESULT	人脸抓拍报警	NET_VCA_FACESNAP_RESULT
COMM_SNAP_MATCH_ALARM	人脸授权名单比对结果上传	NET_VCA_FACESNAP_MATCH_ALARM

[返回目录](#)

布防撤防

3.7.2 建立报警上传通道，获取报警等信息 **NET_DVR_SetupAlarmChan_V41**

函数： public int NET_DVR_SetupAlarmChan_V41(int IUserID, Pointer lpSetupParam)

参数： [in] IUserID NET_DVR_Login_V30 的返回值
[in] lpSetupParam 报警布防参数，对应结构体：[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： **In class com.hcnetsdk.jna.HCNetSDKByJNA, JNA 接口。**使用该接口支持上传 V3.0 以上版本支持的设备的报警结构。启动布防前，需要调用注册回调函数的接口（如 [NET_DVR_SetDVRMessageCallBack_V30](#)）才能获取到上传的报警等信息。

[返回目录](#)

3.7.3 撤销报警上传通道 **NET_DVR_CloseAlarmChan_V30**

函数： public boolean NET_DVR_CloseAlarmChan_V30(int 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： In class `com.hcnetsdk.jna.HCNetSDKByJNA`, JNA 接口。

[返回目录](#)

3.8 远程参数配置

通用参数配置

3.8.1 获取设备配置信息 `NET_DVR_GetDVRConfig`

函数： `public boolean NET_DVR_GetDVRConfig(int IUserID, int dwCommand, int IChannel, NET_DVR_CONFIG DVRConfig)`

参数：

- [in] IUserID NET_DVR_Login_V30 的返回值
- [in] dwCommand 设备配置命令，详见表 3.10
- [in] IChannel 通道号，不同的命令对应不同的取值，如果该参数无效则置为 0xFFFFFFFF 即可，详见表 3.10
- [out] DVRConfig 配置信息，不同的配置功能对应不同的子类，详见表 3.10

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： In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。不同的获取功能对应不同的类和命令号，如表 3.10 所示。

表 3.10 获取设备参数

dwCommand 宏定义	dwCommand 含义	通道号	DVRConfig 对应子类	宏定义值
NET_DVR_GET_DEVICECFG_V40	获取设备参数	无效	NET_DVR_DEVICECFG_V40	1100
NET_DVR_GET_TIMECFG	获取时间参数	无效	NET_DVR_TIME	118
NET_DVR_GET_USERCFG_V30	获取用户参数	无效	NET_DVR_USER_V30	1006
NET_DVR_GET_PICCFG_V30	获取图像参数	通道号	NET_DVR_PICCFG_V30	1002
NET_DVR_GET_COMPRESSCFG_V30	获取压缩参数	通道号	NET_DVR_COMPRESSIONCFG_V30	1040
NET_DVR_GET_RECORDCFG_V30	获取录像参数	通道号	NET_DVR_RECORD_V30	1004
NET_DVR_GET_SHOWSTRING_V30	获取字符叠加参数	通道号	NET_DVR_SHOWSTRING_V30	1030
NET_DVR_GET_ALARMINGCFG_V30	获取报警输入参数	报警输入号	NET_DVR_ALARMINGCFG_V30	1024
NET_DVR_GET_ALARMOUTCFG_V30	获取报警输出参数	报警输出号，从 0 开始	NET_DVR_ALARMOUTCFG_V30	1026
NET_DVR_GET_DECODERCFG_V30	获取 RS485 串口参数	通道号	NET_DVR_DECODERCFG_V30	1042
NET_DVR_GET_IPPARACFG_V40	获取 IP 接入配置参数	组号	NET_DVR_IPPARACFG_V40	1062
NET_DVR_GET_IPALARMOUTCFG	获取 IP 报警输出接入参数	无效	NET_DVR_IPALARMOUTCFG	1052
NET_DVR_GET_NETCFG_V30	获取网络参数	无效	NET_DVR_NETCFG_V30	1000
NET_DVR_GET_DDNSCFG_V30	获取 DDNS 配置	无效	NET_DVR_DDNSPARA_V30	1010

NET_DVR_GET_NTPCFG	获取 NTP 参数	无效	NET_DVR_NTTPARA	224
NET_DVR_GET_WIFI_STATUS	获取 Wifi 状态	无效	NET_DVR_WIFI_CONNECT_STATUS	310
NET_DVR_GET_AP_INFO_LIST	获取无线网络资源参数	无效	NET_DVR_AP_INFO_LIST	305
NET_DVR_GET_WIFI_CFG	获取 IP 监控设备无线参数	无效	NET_DVR_WIFI_CFG	307
NET_DVR_GET_COMPRESSCFG_AUD	获取对讲音频参数	无效	NET_DVR_COMPRESSION_AUDIO	1058
NET_IPC_GET_AUX_ALARMCFG	获取辅助报警参数	通道号	NET_IPC_AUX_ALARMCFG	3209
NET_DVR_GET_ZEROCHANCFG	获取零通道压缩参数	通道号	NET_DVR_ZEROCHANCFG	1102
NET_DVR_GET_DIGITAL_CHANNEL_STATE	获取数字通道状态	无效	NET_DVR_DIGITAL_CHANNEL_STATE	6126
NET_DVR_GET_PRESET_NAME	获取预置点名称	通道号	NET_DVR_PRESET_NAME_ARRAY	3383
NET_DVR_GET_SHOWSTRING_V30	获取叠加字符参数	通道号	NET_DVR_SHOWSTRING_V30	1030

[返回目录](#)

3.8.2 设置设备配置信息 **NET_DVR_SetDVRConfig**

函 数: public boolean NET_DVR_SetDVRConfig(int UserID, int dwCommand, int IChannel, NET_DVR_CONFIG DVRConfig)

参 数: [in] UserID NET_DVR_Login_V30 的返回值
[in] dwCommand 设备配置命令，详见表 3.11
[in] IChannel 通道号，不同的命令对应不同的取值，如果该参数无效则置为 0xFFFFFFFF 即可，详见表 3.11
[in] DVRConfig 配置信息，不同的配置功能对应不同的类，详见表 3.11

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。不同的设置功能对应不同的类和命令号，如表 3.11 所示。

表 3.11 设置设备参数

dwCommand 宏定义	dwCommand 含义	通道号	IpOutBuffer 对应结构体	宏定义值
NET_DVR_SET_DEVICECFG_V40	设置设备参数	无效	NET_DVR_DEVICECFG_V40	1101
NET_DVR_SET_TIMECFG	设置时间参数	无效	NET_DVR_TIME	119
NET_DVR_SET_USERCFG_V30	设置用户参数	无效	NET_DVR_USER_V30	1007
NET_DVR_SET_PICCFG_V30	设置图像参数	通道号	NET_DVR_PICCFG_V30	1003
NET_DVR_SET_COMPRESSCFG_V30	设置压缩参数	通道号	NET_DVR_COMPRESSIONCFG_V30	1041
NET_DVR_SET_RECORDCFG_V30	设置录像参数	通道号	NET_DVR_RECORD_V30	1005
NET_DVR_SET_SHOWSTRING_V30	设置字符叠加参数	通道号	NET_DVR_SHOWSTRING_V30	1031
NET_DVR_SET_ALARMINGCFG_V30	设置报警输入参数	报警输入号	NET_DVR_ALARMINGCFG_V30	1025
NET_DVR_SET_ALARMOUTCFG_V30	设置报警输出参数	报警输出号， 从 0 开始	NET_DVR_ALARMOUTCFG_V30	1027
NET_DVR_SET_DECODERCFG_V30	设置 RS485 串口参数	通道号	NET_DVR_DECODERCFG_V30	1043

NET_DVR_SET_IPPARACFG_V40	设置 IP 接入配置参数	组号	NET_DVR_IPPARACFG_V40	1063
NET_DVR_SET_NETCFG_V30	设置网络参数	无效	NET_DVR_NETCFG_V30	1001
NET_DVR_SET_DDNSCFG_V30	设置 DDNS 配置	无效	NET_DVR_DDNSPARA_V30	1011
NET_DVR_SET_NTPCFG	设置 NTP 参数	无效	NET_DVR_NTPPARA	225
NET_DVR_SET_WIFI_CFG	设置 IP 监控设备无线参数	无效	NET_DVR_WIFI_CFG	306
NET_IPC_SET_AUX_ALARMCFG	设置辅助报警参数	通道号	NET_IPC_AUX_ALARMCFG	3210
NET_DVR_SET_ZEROCHANCFG	设置零通道压缩参数	通道号	NET_DVR_ZEROCHANCFG	1103

[返回目录](#)

3.8.3 批量获取配置信息 **NET_DVR_GetDeviceConfig**

- 函 数： public boolean NET_DVR_GetDeviceConfig(int lUserID, int dwCommand, int dwCount, Pointer lpInBuffer, int dwInBufferSize, Pointer lpStatusList, Pointer lpOutBuffer, int 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 3.12
 - [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 3.13
 - [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 3.13），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： **In class com.hcnetsdk.jna.HCNetSDKByJNA, JNA 接口。**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 3.13 所示。

表 3.12 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

表 3.13 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 NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND	dwCount 个 NET_DVR_MULTI_STREAM_COMPRESSIONCFG

[返回目录](#)

3.8.4 批量设置配置信息 **NET_DVR_SetDeviceConfig**

- 函 数: public boolean NET_DVR_SetDeviceConfig (int IUserID, int dwCommand, int dwCount, Pointer lpInBuffer, int dwInBufferSize, Pointer lpStatusList, Pointer lpInParamBuffer, int dwInParamBufferSize)
- 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 3.14
 - [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 3.15
 - [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 或者 1 表示成功, 其他值为失败对应的错误号
 - [in] lpInParamBuffer 需要设置给设备的参数内容 (详见表 3.15), 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] dwInParamBufferSize 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: **In class com.hcnetsdk.jna.HCNetSDKByJNA, JNA 接口。**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 3.15 所示。

表 3.14 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

表 3.15 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 NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND	dwCount 个 NET_DVR_MULTI_STREAM_COMPRESSIONCFG

[返回目录](#)

报警输出配置

3.8.5 获取设备报警输出 **NET_DVR_GetAlarmOut_V30**

- 函 数: public boolean NET_DVR_GetAlarmOut_V30(int IUserID, NET_DVR_ALARMOUTSTATUS_V30 AlarmStatus)
- 参 数: [in] IUserID NET_DVR_Login_V30 的返回值

[out] AlarmStatus 报警输出状态，详见：[NET_DVR_ALARMOUTSTATUS_V30](#)

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：In class **com.hikvision.netsdk.HCNetSDK**, **JNI 接口**。

[返回目录](#)

3.8.6 设置设备报警输出 **NET_DVR_SetAlarmOut**

函数：public boolean NET_DVR_SetAlarmOut(int IUserID, int IAlarmOutPort, int IAlarmOutStatic)

参数：[in] IUserID NET_DVR_Login_V30 的返回值

[in] IAlarmOutPort 报警输出口，初始输出口从 0 开始。0x00ff 表示全部模拟输出，0xff00 表示全部数字输出。设备支持对 IP 接入的报警输出进行处理时，对应 32-95 为数字报警输出

[in] IAlarmOutStatic 报警输出状态：0- 停止输出，1- 输出

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：In class **com.hikvision.netsdk.HCNetSDK**, **JNI 接口**。

[返回目录](#)

设备支持的云台协议

3.8.7 获取设备支持的云台协议 **NET_DVR_GetPTZProtocol**

函数：public boolean NET_DVR_GetPTZProtocol(int IUserID, NET_DVR_PTZCFG struPtz)

参数：[in] IUserID NET_DVR_Login_V30 的返回值

[out] struPtz 设备的云台协议，详见：[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：In class **com.hikvision.netsdk.HCNetSDK**, **JNI 接口**。在配置前端云台协议时调用该接口获取当前设备支持的云台协议。

[返回目录](#)

3.9 录像文件回放、下载、锁定及备份

刷新录像索引

3.9.1 即时刷新录像索引 **NET_DVR_UpdateRecordIndex**

函数：public boolean NET_DVR_UpdateRecordIndex(int IUserID, int dwChannel)

参数：[in] IUserID NET_DVR_Login_V30 的返回值

[in] dwChannel 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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。录像索引刷新后, 即可回放刷新前的录像文件。和通道相关, 需要设备支持, 设备默认每 2 分钟刷新一次。

[返回目录](#)

录像文件的查找

3.9.2 根据文件类型、时间查找设备录像文件 NET_DVR_FindFile_V30

函 数: public int NET_DVR_FindFile_V30(int lUserID, NET_DVR_FILECOND pFindCond)

参 数: [in] lUserID NET_DVR_Login_V30 的返回值

[in] pFindCond 待查找的文件信息结构, 详见: [NET_DVR_FILECOND](#)

返回值: -1 表示失败, 其他值作为 NET_DVR_FindClose 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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。该接口指定了要查找的录像文件的信息, 调用成功后, 就可以调用 [NET_DVR_FindNextFile_V30](#) 接口来获取文件信息。

[返回目录](#)

3.9.3 逐个获取查找到的文件信息 NET_DVR_FindNextFile_V30

函 数: public int NET_DVR_FindNextFile_V30(int lFindHandle, NET_DVR_FINDDATA_V30 lpFindData)

参 数: [in] lFindHandle 文件查找句柄, NET_DVR_FindFile_V30 的返回值

[in] lpFindData 保存文件信息的指针, 详见: [NET_DVR_FINDDATA_V30](#)

返回值: -1 表示失败, 其他值表示当前的获取状态等信息, 详见表 3.16。

表 3.16 查找结果信息

宏定义	宏定义值	含义
NET_DVR_FILE_SUCCESS	1000	获取文件信息成功
NET_DVR_FILE_NOFIND	1001	未查找到文件
NET_DVR_ISFINDING	1002	正在查找请等待
NET_DVR_NOMOREFILE	1003	没有更多的文件, 查找结束
NET_DVR_FILE_EXCEPTION	1004	查找文件时异常

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。在调用该接口获取查找文件之前, 必须先调用 NET_DVR_FindFile_V30 得到当前的查找句柄。此接口用于获取一条已查找到的文件信息, 若要获取全部的已查找到的文件信息, 需要循环调用此接口。通过此接口可以同时获取到与当前录像文件相关的卡号信息和文件是否被锁定的信息。

每次可查询文件最大个数为 4000。

[返回目录](#)

3.9.4 关闭文件查找，释放资源 **NET_DVR_FindClose_V30**

函 数： public boolean NET_DVR_FindClose_V30(int IFindHandle)
参 数： [in] IFindHandle 文件查找句柄，NET_DVR_FindFile_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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

按事件查找录像

3.9.5 按事件查找录像 **NET_DVR_FindFileByEvent**

函 数： public int NET_DVR_FindFileByEvent(int IUserID, NET_DVR_SEARCH_EVENT_PARAM lpSearchEventParam)
参 数： [in] IUserID NET_DVR_Login_V30 的返回值
[in] lpSearchEventParam 事件搜索条件，详见：[NET_DVR_SEARCH_EVENT_PARAM](#)
返回值： -1 表示失败，其他值作为 NET_DVR_FindNextEvent 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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 该接口指定了要查找的录像文件的信息，调用成功后，就可以调用 NET_DVR_FindNextEvent 接口来获取文件信息。该功能需要设备的支持，若设备不支持则接口返回失败，错误号为 23。

[返回目录](#)

3.9.6 逐个获取查找到的文件信息 **NET_DVR_FindNextEvent**

函 数： public int NET_DVR_FindNextEvent(int IFindHandle, NET_DVR_SEARCH_EVENT_RET lpSearchEventRet)
参 数： [in] IFindHandle 文件查找句柄，NET_DVR_FindFileByEvent 的返回值
[in] lpSearchEventRet 保存文件信息的指针，详见：[NET_DVR_SEARCH_EVENT_RET](#)
返回值： -1 表示失败，其他值表示当前的获取状态等信息，详见表 3.17。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 3.17 查找结果信息

宏定义	宏定义值	含义
NET_DVR_FILE_SUCCESS	1000	获取文件信息成功
NET_DVR_FILE_NOFIND	1001	未查找到文件
NET_DVR_ISFINDING	1002	正在查找请等待
NET_DVR_NOMOREFILE	1003	没有更多的文件，查找结束
NET_DVR_FILE_EXCEPTION	1004	查找文件时异常

说 明： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 在调用该接口获取查找文件之前，必须先调

用 `NET_DVR_FindFile_V30` 得到当前的查找句柄。此接口用于获取一条已查找到的文件信息，若要获取全部的已查找到的文件信息，需要循环调用此接口。通过此接口可以同时获取到与当前录像文件相关的卡号信息和文件是否被锁定的信息。

有的设备每次可查询文件最大个数为 2000，有的每次设备可查询文件最大个数为 4000，所以查询到个数刚好为 2000 或者 4000 时，需要传入新的时间段查找更多文件。

[返回目录](#)

3.9.7 关闭文件查找，释放资源 `NET_DVR_FindClose_V30`

函 数: `public boolean NET_DVR_FindClose_V30(int IFindHandle)`

参 数: `[in] IFindHandle` 文件查找句柄，`NET_DVR_FindFileByEvent` 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: **In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。**

[返回目录](#)

回放录像文件

3.9.8 注册回调函数，捕获录像数据 `NET_DVR_SetPlayDataCallBack`

函 数: `public boolean NET_DVR_SetPlayDataCallBack(int IPlayHandle, PlaybackCallBack cbPlayDataCallBack)`

参 数: `[in] IPlayHandle` 播放句柄，`NET_DVR_PlayBackByName` 或
`NET_DVR_PlayBackByTime` 的返回值

`[in] PlaybackCallBack` 录像数据回调函数

```
public void fPlayDataCallBack(int iPlayHandle, int iDataType, byte[] pDataBuffer, int iDataSize)
```

```
public interface PlaybackCallBack {
```

```
    public void fPlayDataCallBack(int iPlayHandle, int iDataType, byte[] pDataBuffer, int iDataSize);
```

```
}
```

`[out] iPlayHandle` 当前的录像播放句柄

`[out] iDataType` 数据类型，详见表 3.18

`[out] pDataBuffer` 存放数据的缓冲区指针

`[out] iDataSize` 缓冲区大小

表 3.18 回放数据类型

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

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

说 明: **In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。**此函数包括开始和停止用户处理 SDK 捕获的数据，当回调函数 `cbPlayDataCallBack` 设为非 `NULL` 值时，表示回调和处理数据；当设为 `NULL` 时表示停止回调和处理数据。回调的第一个包是 40 个字节的文件头，供后续解码使用，之后回调

的是压缩的码流。

[返回目录](#)

3.9.9 按文件名回放录像文件 **NET_DVR_PlayBackByName**

函 数: public int NET_DVR_PlayBackByName(int IUserID, String sPlayBackFileName)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] sPlayBackFileName 回放的文件名, 长度不能超过 100 字节

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

说 明: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**需要先调用接口 NET_DVR_SetPlayDataCallBack 注册回调函数, 捕获录像的码流数据并自行处理 (比如解码显示)。该接口指定了当前要播放的录像文件, 调用成功后, 还必须调用 NET_DVR_PlayBackControl_V40 接口的开始播放控制命令 (NET_DVR_PLAYSTART) 才能实现回放。

[返回目录](#)

3.9.10 按时间回放录像文件 **NET_DVR_PlayBackByTime**

函 数: public int NET_DVR_PlayBackByTime(int IUserID, int IChannel, NET_DVR_TIME lpStartTime, NET_DVR_TIME lpStopTime)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] IChannel 通道号
[in] lpStartTime 开始时间, 详见: [NET_DVR_TIME](#)
[in] lpStopTime 结束时间, 详见: [NET_DVR_TIME](#)

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

说 明: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**需要先调用接口 NET_DVR_SetPlayDataCallBack 注册回调函数, 捕获录像的码流数据并自行处理 (比如解码显示)。该接口指定了当前要回放录像的起止时间, 调用成功后, 还必须调用 NET_DVR_PlayBackControl_V40 接口的开始播放控制命令 (NET_DVR_PLAYSTART) 才能实现回放。当回放的是按事件搜索出的录像文件时, 由于每个文件都会有预录和延迟的部分, 因此在设置本接口的开始和结束时间参数时可以适当提前开始时间和延长结束时间。建议值: 最多 10 分钟, 最少 5 秒。

[返回目录](#)

3.9.11 按时间回放录像文件 **NET_DVR_PlayBackByTime_V40**

函 数: public int NET_DVR_PlayBackByTime(int IUserID, [NET_DVR_VOD PARA](#) param pVodPara)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] pVodPara 查找条件, 详见: [NET_DVR_VOD PARA](#)

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

说 明: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**需要先调用接口 NET_DVR_SetPlayDataCallBack

注册回调函数，捕获录像的码流数据并自行处理（比如解码显示）。该接口指定了当前要回放录像的起止时间，调用成功后，还必须调用 `NET_DVR_PlayBackControl_V40` 接口的开始播放控制命令（`NET_DVR_PLAYSTART`）才能实现回放。当回放的是按事件搜索出的录像文件时，由于每个文件都会有预录和延迟的部分，因此在设置本接口的开始和结束时间参数时可以适当提前开始时间和延长结束时间。建议值：最多 10 分钟，最少 5 秒。

[返回目录](#)

3.9.12 控制录像回放的状态 `NET_DVR_PlayBackControl_V40`

函数： `public boolean NET_DVR_PlayBackControl_V40(int IPlayHandle, int dwControlCode, byte[] lpInBuffer, int dwInLen, NET_DVR_PLAYBACK_INFO lpOutValue)`

参数：

- [in] IPlayHandle 播放句柄，`NET_DVR_PlayBackByName` 或 `NET_DVR_PlayBackByTime` 的返回值
- [in] dwControlCode 控制录像回放状态命令，详见表 3.19
- [in] lpInBuffer 指向输入参数的指针，设为 `NULL`
- [in] dwInLen 输入参数的长度，无效，设置为 0
- [out] lpOutBuffer 指向输出参数的指针，无效，设置为 `NULL`

表 3.19 回放控制命令

dwControlCode 宏定义	宏定义值	含义
<code>NET_DVR_PLAYSTART</code>	1	开始播放
<code>NET_DVR_PLAYPAUSE</code>	3	暂停播放
<code>NET_DVR_PLAYRESTART</code>	4	恢复播放

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： **In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。**

[返回目录](#)

3.9.13 获取回放取流进度 `NET_DVR_GetPlayBackPos`

函数： `public int NET_DVR_GetPlayBackPos(int IPlayHandle)`

参数：

- [in] IPlayHandle 回放取流句柄，`NET_DVR_PlayBackByName` 或 `NET_DVR_PlayBackByTime` 的返回值

返回值： 0~100 表示进度，200 表示回放出现异常，-1 表示接口失败。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： **In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。** 这里获取的进度指的是网络上收到的数据百分比，比如回放的文件大小是 100M，当前已收到 10M，则返回 10。如果回调中有进行解码显示，在判断回放是否结束时应先调用本接口确认进度 100，再调用播放库接口确认解码缓冲区中已没有数据，才能认为本次回放结束。只有按文件回放时才会有 0~100 之间的进度，按时间回放时进度只有 0 和 100。

[返回目录](#)

3.9.14 停止回放录像文件 **NET_DVR_StopPlayBack**

函 数: public boolean NET_DVR_StopPlayBack(int IPlayHandle)

参 数: [in] IPlayHandle 回放句柄, NET_DVR_PlayBackByName 或
NET_DVR_PlayBackByTime 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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.9.15 获取回放时用来解码显示的播放库句柄 **NET_DVR_GetPlayBackPlayerIndex**

函 数: public int NET_DVR_GetPlayBackPlayerIndex(int IPlayHandle)

参 数: [in] iRealHandle 回放句柄, NET_DVR_PlayBackByName 或
NET_DVR_PlayBackByTime_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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.9.16 录像回放时抓图 **NET_DVR_PlayBackCaptureFile**

函 数: public boolean NET_DVR_PlayBackCaptureFile(int IPlayHandle, String sFileName)

参 数: [in] iRealHandle 回放句柄, NET_DVR_PlayBackByName 或
NET_DVR_PlayBackByTime_V40 的返回值
[in] sPicFileName 保存图片的文件路径, 包含文件名

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.9.17 按文件名倒放录像文件 **NET_DVR_PlayBackReverseByName**

函 数: public int NET_DVR_PlayBackReverseByName(int IUserID, String sPlayBackFileName, Surface hWnd)

参 数: [in] IUserID NET_DVR_Login_V40 等登录接口的返回值
[in] sPlayBackFileName 回放的文件名, 长度不能超过 100 字节
[in] hWnd 回放的窗口句柄, 若置为空, SDK 仍能收到码流数据, 但不解码显示。

返回值: -1 表示失败, 其他值作为 NET_DVR_StopPlayBack 等函数的参数。接口返回失败请调用 NET_DVR_GetLastError 获取错误码, 通过错误码判断出错原因。

说 明: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.9.18 按时间倒放录像文件 **NET_DVR_PlayBackReverseByTime_V40**

函 数: public int NET_DVR_PlayBackReverseByTime_V40(int paramInt, Surface hWnd, [NET_DVR_PLAYCOND](#) PlayParam)

参 数: [in] IUserID NET_DVR_Login_V40 等登录接口的返回值
[in] hWnd 回放的窗口句柄, 若置为空, SDK 仍能收到码流数据, 但不解码显示
[in] PlayParam 回放条件, 详见: [NET_DVR_PLAYCOND](#)

返回值: -1 表示失败, 其他值作为 NET_DVR_StopPlayBack 等函数的参数。接口返回失败请调用 NET_DVR_GetLastError 获取错误码, 通过错误码判断出错原因。

说 明: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

下载录像文件

3.9.19 按文件名下载录像文件 **NET_DVR_GetFileByName**

函 数: public int NET_DVR_GetFileByName(int IUserID, String sDVRFileName, String sSavedFileName)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] sDVRFileName 要下载的录像文件名, 文件名长度需小于 100 字节
[in] sSavedFileName 下载后保存到 PC 机的文件路径, 需为绝对路径

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

说 明: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 在使用该接口下载录像文件前, 可以先调用录像文件查找的接口获取文件名。该接口指定了当前要下载的录像文件, 调用成功后, 还需要调用 NET_DVR_PlayBackControl 接口的开始播放控制命令 (NET_DVR_PLAYSTART) 才能实现下载。

[返回目录](#)

3.9.20 按时间下载录像文件 **NET_DVR_GetFileByTime**

函 数: public int NET_DVR_GetFileByTime(int IUserID, int IChannel, NET_DVR_TIME lpStartTime, NET_DVR_TIME lpStopTime, String sSavedFileName)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] IChannel 通道号
[in] lpStartTime 开始时间, 详见: [NET_DVR_TIME](#)
[in] lpStopTime 结束时间, 详见: [NET_DVR_TIME](#)

[in]sSavedFileName 下载后保存到 PC 机的文件路径，需为绝对路径

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

说明: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。该接口指定了当前要下载的录像文件，调用成功后，还需要调用 NET_DVR_PlayBackControl 接口的开始播放控制命令（NET_DVR_PLAYSTART）才能实现下载。

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](#)

3.9.21 控制录像下载的状态 NET_DVR_PlayBackControl_V40

函数: public boolean NET_DVR_PlayBackControl_V40(int IPlayHandle, int dwControlCode, byte[] lpInBuffer, int dwInLen, NET_DVR_PLAYBACK_INFO lpOutValue)

参数: [in]IPlayHandle 下载句柄，NET_DVR_GetFileByName 或 NET_DVR_GetFileByTime 的返回值

[in]dwControlCode 控制录像回放状态命令，详见表 3.20

[in]lpInBuffer 指向输入参数的指针，设为 NULL

[in]dwInLen 输入参数的长度

[out]lpOutValue 指向输出参数的指针，无效，设为 NULL

表 3.20 下载控制命令

dwControlCode 宏定义	宏定义值	含义
NET_DVR_PLAYSTART	1	开始下载
NET_DVR_PLAYPAUSE	3	暂停下载
NET_DVR_PLAYRESTART	4	恢复下载
NET_DVR_PLAYFAST	5	快放
NET_DVR_PLAYSLow	6	慢放
NET_DVR_PLAYSTARTAUDIO	9	打开声音
NET_DVR_PLAYSTOPAUDIO	10	关闭声音
NET_DVR_PLAYAUDIOVOLUME	11	调节音量

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.9.22 获取当前下载录像文件的进度 NET_DVR_GetDownloadPos

函数: public int NET_DVR_GetDownloadPos(int IFileHandle);

参数: [in]IFileHandle 下载句柄，NET_DVR_GetFileByName 或 NET_DVR_GetFileByTime 的返回值

返回值: -1 表示失败, 0~100 表示下载的进度, 100 表示下载结束, 200 表示网络异常。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: In class **com.hikvision.netsdk.HCNetSDK**, **JNI 接口**。只有按文件下载时才会有 0~100 之间的进度, 按时间下载时进度只有 0 和 100。

[返回目录](#)

3.9.23 停止下载录像文件 **NET_DVR_StopGetFile**

函数: public boolean NET_DVR_StopGetFile(int IFileHandle)

参数: [in]IFileHandle 下载句柄, NET_DVR_GetFileByName 或 NET_DVR_GetFileByTime 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: In class **com.hikvision.netsdk.HCNetSDK**, **JNI 接口**。

[返回目录](#)

3.10 云台控制

云台控制操作

3.10.1 云台控制操作（需先启动预览）**NET_DVR_PTZControl**

函数: public boolean NET_DVR_PTZControl(int IRealHandle, int dwPTZCommand, int dwStop)

参数: [in] IRealHandle NET_DVR_RealPlay_V40 的返回值

[in] dwPTZCommand 云台控制命令, 详见表 3.21

[in] dwStop 云台停止动作或开始动作: 0- 开始, 1- 停止

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: In class **com.hikvision.netsdk.HCNetSDK**, **JNI 接口**。对云台实施的每一个动作都需要调用该接口两次, 分别是开始和停止控制, 由接口中的最后一个参数（dwStop）决定。在调用此接口之前需要先开启预览。与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码, 设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配, 需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持, 则无法用该接口控制。

云台默认以最大速度动作。

表 3.21 云台控制命令

dwPTZCommand 宏定义	宏定义值	含义
LIGHT_PWRON	2	接通灯光电源
WIPER_PWRON	3	接通雨刷开关
FAN_PWRON	4	接通风扇开关

HEATER_PWRON	5	接通加热器开关
AUX_PWRON1	6	接通辅助设备开关
AUX_PWRON2	7	接通辅助设备开关
ZOOM_IN	11	焦距变大(倍率变大)
ZOOM_OUT	12	焦距变小(倍率变小)
FOCUS_NEAR	13	焦点前调
FOCUS_FAR	14	焦点后调
IRIS_OPEN	15	光圈扩大
IRIS_CLOSE	16	光圈缩小
TILT_UP	21	云台上仰
TILT_DOWN	22	云台下俯
PAN_LEFT	23	云台左转
PAN_RIGHT	24	云台右转
UP_LEFT	25	云台上仰和左转
UP_RIGHT	26	云台上仰和右转
DOWN_LEFT	27	云台下俯和左转
DOWN_RIGHT	28	云台下俯和右转
PAN_AUTO	29	云台左右自动扫描

[返回目录](#)

3.10.2 云台控制操作（不用启动预览）NET_DVR_PTZControl_Other

函 数： public boolean NET_DVR_PTZControl_Other(int IUserID, int IChannel, int dwPTZCommand, int dwStop)

参 数： [in] IUserID NET_DVR_Login_V30 的返回值

[in] IChannel 通道号

[in] dwPTZCommand 云台控制命令，详见表 3.21

[in] dwStop 云台停止动作或开始动作：0- 开始；1- 停止

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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**对云台实施的每一个动作都需要调用该接口两次，分别是开始和停止控制，由接口中的最后一个参数（dwStop）决定。在调用此接口之前需要先注册设备。与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码，设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配，需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持，则无法用该接口控制。
云台默认以最大速度动作。

[返回目录](#)

3.10.3 带速度的云台控制操作（需先启动预览） **NET_DVR_PTZControlWithSpeed**

函 数： public boolean NET_DVR_PTZControlWithSpeed(int IRealHandle, int dwPTZCommand, int dwStop, int dwSpeed)

参 数：

[in] IRealHandle	NET_DVR_RealPlay_V40 的返回值
[in] dwPTZCommand	云台控制命令，详见表 3.21
[in] dwStop	云台停止动作或开始动作：0- 开始；1- 停止
[in] dwSpeed	云台控制的速度，用户按不同解码器的速度控制值设置。取值范围[1,7]

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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

3.10.4 带速度的云台控制操作（不用启动预览）

NET_DVR_PTZControlWithSpeed_Other

函 数： public boolean NET_DVR_PTZControlWithSpeed_Other(int IUserID, int IChannel, int dwPTZCommand, int dwStop, int dwSpeed)

参 数：

[in] IUserID	NET_DVR_Login_V30 的返回值
[in] IChannel	通道号
[in] dwPTZCommand	云台控制命令，详见表 3.21
[in] dwStop	云台停止动作或开始动作：0- 开始；1- 停止
[in] dwSpeed	云台控制的速度，用户按不同解码器的速度控制值设置。取值范围： [1,7]

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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**对云台实施的每一个动作都需要调用该接口两次，分别是开始和停止控制，由接口中的最后一个参数（dwStop）决定。在调用此接口之前不需要先开启预览，登录设备后即可实现控制。与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码，设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配，需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持，则无法用该接口控制。

[返回目录](#)

云台预置点操作

3.10.5 云台预置点操作，需先启动预览 **NET_DVR_PTZPreset**

函 数： public boolean NET_DVR_PTZPreset(int IRealHandle, int dwPTZPresetCmd, int dwPresetIndex)

参 数：

[in] IRealHandle	NET_DVR_RealPlay_V40 的返回值
------------------	---------------------------

[in] dwPTZPresetCmd 云台预置点操作命令，详见表 3.22

[in] dwPresetIndex 预置点的序号（从 1 开始），最多支持 255 个预置点

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码，设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配，需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持，则无法用该接口控制。

表 3.22 预置点操作命令

dwPTZPresetCmd 宏定义	宏定义值	含义
SET_PRESET	8	设置预置点
CLE_PRESET	9	清除预置点
GOTO_PRESET	39	转到预置点

[返回目录](#)

3.10.6 云台预置点操作 NET_DVR_PTZPreset_Other

函 数: public boolean NET_DVR_PTZPreset_Other(int IUserID, int IChannel, int dwPTZPresetCmd, int dwPresetIndex)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值

[in] IChannel 通道号

[in] dwPTZPresetCmd 云台预置点操作命令，详见表 3.22

[in] dwPresetIndex 预置点的序号（从 1 开始），最多支持 255 个预置点

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码，设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配，需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持，则无法用该接口控制。

通过 NET_DVR_PTZPreset 控制云台，设备接收到控制命令后云台进行相应的动作，如果操作失败则返回错误，运行正常才返回成功。而通过 NET_DVR_PTZPreset_Other 控制云台，设备接收到控制命令后直接返回成功

[返回目录](#)

云台巡航操作

3.10.7 云台巡航操作，需先启动预览 NET_DVR_PTZPCruise

函 数: public boolean NET_DVR_PTZCruise(int IRealHandle, int dwPTZCruiseCmd, byte byCruiseRoute, byte byCruisePoint, short wInput)

参 数: [in] IRealHandle NET_DVR_RealPlay_V40 的返回值

[in] dwPTZCruiseCmd 操作云台巡航命令，详见表 3.23

[in] byCruiseRoute 巡航路径，最多支持 32 条路径（序号从 1 开始）
[in] byCruisePoint 巡航点，最多支持 32 个点（序号从 1 开始）
[in] wInput 不同巡航命令时的值不同，预置点(最大 255)、时间(最大 255)、速度(最大 40)

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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码，设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配，需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持，则无法用该接口控制。

表 3.23 巡航操作命令

dwPTZCruiseCmd 宏定义	宏定义值	含义
FILL_PRE_SEQ	30	将预置点加入巡航序列
SET_SEQ_DWELL	31	设置巡航点停顿时间
SET_SEQ_SPEED	32	设置巡航速度
CLE_PRE_SEQ	33	将预置点从巡航序列中删除
RUN_SEQ	37	开始巡航
STOP_SEQ	38	停止巡航

[返回目录](#)

3.10.8 云台巡航操作 **NET_DVR_PTZCruise_Other**

函数： public boolean NET_DVR_PTZCruise_Other(int IUserID, int IChannel, int dwPTZCruiseCmd, byte byCruiseRoute, byte byCruisePoint, short wInput)

参数： [in] IUserID NET_DVR_Login_V30 的返回值
[in] IChannel 通道号
[in] dwPTZCruiseCmd 操作云台巡航命令，详见表 3.23
[in] byCruiseRoute 巡航路径，最多支持 32 条路径（序号从 1 开始）
[in] byCruisePoint 巡航点，最多支持 32 个点（序号从 1 开始）
[in] wInput 不同巡航命令时的值不同，预置点(最大 255)、时间(最大 255)、速度(最大 40)

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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码，设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配，需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持，则无法用该接口控制。

[返回目录](#)

云台花样扫描操作

3.10.9 云台花样扫描操作，需先启动预览 **NET_DVR_PTZTrack**

函 数： public boolean NET_DVR_PTZTrack(int IRealHandle, int dwPTZTrackCmd)
参 数： [in] IRealHandle NET_DVR_RealPlay_V40 的返回值
[in] dwPTZTrackCmd 操作云台巡航命令，详见表 3.24
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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码，设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配，需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持，则无法用该接口控制。

表 3.24 花样扫描操作命令

dwPTZTrackCmd 宏定义	宏定义值	含义
STA_MEM_CRUISE	34	开始记录花样扫描
STO_MEM_CRUISE	35	停止记录花样扫描
RUN_CRUISE	36	开始花样扫描

[返回目录](#)

3.10.10 云台花样扫描操作 **NET_DVR_PTZTrack_Other**

函 数： public boolean NET_DVR_PTZTrack_Other(int IUserID, int IChannel, int dwPTZTrackCmd)
参 数： [in] IUserID NET_DVR_Login_V30 的返回值
[in] IChannel 通道号
[in] dwPTZTrackCmd 操作云台巡航命令，详见表 3.24
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： **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码，设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配，需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持，则无法用该接口控制。

[返回目录](#)

云台区域缩放控制

3.10.11 云台图象区域选择放大或缩小 **NET_DVR_PTZSelZoomIn**

函 数： public boolean NET_DVR_PTZSelZoomIn(int IRealHandle, NET_DVR_POINT_FRAME pPointFrame)

参 数: [in] IRealHandle NET_DVR_RealPlay_V40 的返回值
[in] pStruPointFrame 云台图像区域位置信息, 详见: [NET_DVR_POINT_FRAME](#)

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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**该接口实现 3D 定位功能, 需要前端设备的支持。
假设当前预览显示图像的框为 352*288, 原点即该显示框的左上角的顶点。参数 pStruPointFrame 中各坐标值的计算方法 (以 X 轴方向上为例): xTop=鼠标当前所选区域的左上点的值*255/352。
缩小条件: xTop 减去 xBottom 的值大于 2。放大条件: xBottom 大于 xTop。

[返回目录](#)

3.10.12 云台图像区域选择放大或缩小 NET_DVR_PTZSelZoomIn_Ex

函 数: public boolean NET_DVR_PTZSelZoomIn_EX(int IUserID, int IChannel, NET_DVR_POINT_FRAME pPointFrame)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] IChannel 通道号
[in] pStruPointFrame 云台图像区域位置信息, 详见: [NET_DVR_POINT_FRAME](#)

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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**该接口实现 3D 定位功能, 需要前端设备的支持。
假设当前预览显示图像的框为 352*288, 原点即该显示框的左上角的顶点。参数 pStruPointFrame 中各坐标值的计算方法 (以 X 轴方向上为例): xTop=鼠标当前所选区域的左上点的值*255/352。
缩小条件: xTop 减去 xBottom 的值大于 2。放大条件: xBottom 大于 xTop。

[返回目录](#)

3.11 语音转发

3.11.1 获取当前生效的音频对讲音频压缩参数

NET_DVR_GetCurrentAudioCompress

函 数: public boolean NET_DVR_GetCurrentAudioCompress(int IUserID, NET_DVR_COMPRESSION_AUDIO lpCompressAudio)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[out] lpCompressAudio 对讲音频压缩参数, 详见: [NET_DVR_COMPRESSION_AUDIO](#)

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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

3.11.2 启动语音转发，获取编码后的音频数据 **NET_DVR_StartVoiceCom_MR_V30**

函 数: public int NET_DVR_StartVoiceCom_MR_V30(int IUserID, int IVoiceChan, VoiceDataCallBack CallBack)
参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] IVoiceChan 语音通道号，从 1 开始
[in] CallBack 音频数据回调函数，得到的数据是编码以后的音频数据，需调用音频库进行解码播放

```
public interface VoiceDataCallBack {  
    public void fVoiceDataCallBack(int IVoiceComHandle, byte[] pDataBuffer, int iDataSize, int iAudioFlag);  
}
```

[out] IVoiceComHandle NET_DVR_StartVoiceCom_MR_V30 的返回值
[out] pDataBuffer 存放音频数据的缓冲区指针
[out] iDataSize 音频数据大小
[out] iAudioFlag 音频数据类型：1—设备发送过来的音频数据

返回值: -1 表示失败，其他值作为 NET_DVR_VoiceComSendData、NET_DVR_StopVoiceCom 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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。在调用开始语音转发之前可通过接口 [NET_DVR_GetCurrentAudioCompress](#) 获取设备的音频编码类型。

[返回目录](#)

3.11.3 转发语音数据 **NET_DVR_VoiceComSendData**

函 数: public boolean NET_DVR_VoiceComSendData(int IVoiceComHandle, byte[] pSendBuf, int lBufSize)
参 数: [in] IVoiceComHandle NET_DVR_StartVoiceCom_MR_V30 的返回值
[in] pSendBuf 存放语音数据的缓冲区
[in] dwBufSize 语音数据大小

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。该接口实现将获取到的经过编码后的音频数据转发给设备。G722 音频编码类型时，每次发送的数据为 80 字节；当前是 G711 音频编码类型时，每次发送的数据为 160 字节。

[返回目录](#)

3.11.4 停止语音转发 **NET_DVR_StopVoiceCom**

函 数: public boolean NET_DVR_StopVoiceCom(int IVoiceComHandle)
参 数: [in] IVoiceComHandle NET_DVR_StartVoiceCom_MR_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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.12 数据透传

透明通道

3.12.1 建立透明通道 **NET_DVR_SerialStart_V40**

函 数: public int NET_DVR_SerialStart_V40(int IUserID, NET_DVR_SERIAL_COND serialCond, SerialDataCallBackV40 fSerialDataCallBackV40)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] serialCond 串口参数, 详见 [NET_DVR_SERIAL_COND](#)
[in] fSerialDataCallBackV40 透明通道数据回调函数

```
public interface SerialDataCallBackV40{
    public void fSerialDataCallBackV40(int ISerialHandle, int IChannel, byte[] pDataBuffer, int iDataSize);
}
```

[out] ISerialHandle NET_DVR_SerialStart_V40 的返回值
[out] IChannel 串口通道号
[out] pDataBuffer 存放数据的缓冲区指针
[out] iDataSize 数据大小

返回值: -1 表示失败, 其他值作为 NET_DVR_SerialSend 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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。需要从回调函数得到数据解码器必须支持数据回传, 否则发送成功, 回调依然不会有返回。

[返回目录](#)

3.12.2 通过透明通道向设备串口发送数据 **NET_DVR_SerialSend**

函 数: public boolean NET_DVR_SerialSend(int ISerialHandle, int IChannel, byte[] lpSendBuf, int dwBufSize)

参 数: [in] ISerialHandle NET_DVR_SerialStart_V40 的返回值
[in] IChannel 使用 485 串口时有效, 从 1 开始;
232 串口作为透明通道时该值设置为 0
[in] lpSendBuf 发送数据的缓冲区指针
[in] dwBufSize 缓冲区的大小, 最多 1016 字节

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.12.3 断开透明通道 **NET_DVR_SerialStop**

函 数: public boolean NET_DVR_SerialStop(int ISerialHandle)

参 数: [in] ISerialHandle NET_DVR_SerialStart_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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

向串口发送数据

3.12.4 直接向串口发送数据, 不需要建立透明通道 **NET_DVR_SendToSerialPort**

函数: public boolean NET_DVR_SendToSerialPort(int IUserID, int dwSerialPort, int dwSerialIndex, byte[] lpSendBuf, int dwBufSize)

参数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] dwSerialPort 串口类型: 1- 232, 2- 485
[in] dwSerialIndex 表示第几个 232 或者 485, 从 1 开始
[in] lpSendBuf 发送数据的缓冲区指针
[in] dwBufSize 缓冲区的大小, 最多 1016 字节

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.12.5 直接向 232 串口发送数据, 不需要建立透明通道 **NET_DVR_SendTo232Port**

函数: public boolean NET_DVR_SendTo232Port(int IUserID, byte[] lpSendBuf, int dwBufSize)

参数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] lpSendBuf 发送数据的缓冲区指针
[in] dwBufSize 缓冲区的大小, 最多 1016 字节

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: In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.13 设备手动录像

3.13.1 远程手动启动设备录像 **NET_DVR_StartDVRRecord**

函数: public boolean NET_DVR_StartDVRRecord(int IUserID, int IChannel, int IRecordType)

参数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] IChannel 通道号, 0x00ff 表示所有模拟通道, 0xff00 表示所有数字通道, 0xffff 表示所有模拟和数字通道
[in] IRecordType 录像类型: 0-手动, 1-报警, 2-回传, 3-信号, 4-移动, 5-遮挡

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: In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。录像类型设置需要设备支持, 不支持默认为手动录像。

当某通道已经开启定时录像的前提下首次开启手动录像, 此次操作未生效, 仍保持定时录像状态, 且查询设备状态 (见 [NET_DVR_GetDVRWorkState_V30](#), 类 [NET_DVR_WORKSTATE_V30](#)) 中的录像状态仍为录像; 此时关闭手动录像, 停止了定时录像, 且查询录像状态为不录像; 第二次开启手动录像, 此时手动录像开始; 停止手动录像后, 重启设备, 定时录像重新打开。

[返回目录](#)

3.13.2 远程手动停止设备录像 `NET_DVR_StopDVRRecord`

函数: `public boolean NET_DVR_StopDVRRecord(int IUserID, int IChannel)`

参数: [in] IUserID `NET_DVR_Login_V30` 的返回值
[in] IChannel 通道号, 0x00ff 表示所有模拟通道, 0xff00 表示所有数字通道, 0xffff 表示所有模拟和数字通道

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: In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。

[返回目录](#)

3.14 远程面板控制

3.14.1 远程控制面板上的按键 `NET_DVR_ClickKey`

函数: `public boolean NET_DVR_ClickKey(int IUserID, int IKeyIndex)`

参数: [in] IUserID `NET_DVR_Login_V30` 的返回值
[in] IKeyIndex 面板上的按键, 含义详见表 3.25

表 3.25 面板按键

IKeyIndex 宏定义	宏定义值	含义
KEY_CODE_MENU	12	MENU
KEY_CODE_ENTER	13	ENTER
KEY_CODE_CANCEL	14	ESCS
KEY_CODE_UP/KEY_PTZ_UP_START	15	"上"或者"云台上开始"
KEY_CODE_DOWN/KEY_PTZ_DOWN_START	16	"下"或者"云台下开始"
KEY_CODE_LEFT/KEY_PTZ_LEFT_START	17	"左"或者"云台左开始"

KEY_CODE_RIGHT/KEY_PTZ_RIGHT_START	18	"右"或者"云台右开始"
------------------------------------	----	--------------

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: In class **com.hikvision.netsdk.HCNetSDK**, JNI 接口。

[返回目录](#)

3.15 硬盘管理

3.15.1 远程格式化设备硬盘 **NET_DVR_FormatDisk**

函数: public int NET_DVR_FormatDisk(int IUserID, int IDiskNumber)

参数: [in] IUserID NET_DVR_Login_V30 的返回值

[in] IDiskNumber 硬盘号, 从 0 开始, 0xff 表示对所有硬盘有效 (不包括只读硬盘)

返回值: -1 表示失败, 其他值作为 NET_DVR_CloseFormatHandle 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: In class **com.hikvision.netsdk.HCNetSDK**, JNI 接口。格式化过程中如果网络断了, 设备上的格式化操作依然会继续, 但是客户端无法收到状态。

[返回目录](#)

3.15.2 获取格式化硬盘的进度 **NET_DVR_GetFormatProgress**

函数: public boolean NET_DVR_GetFormatProgress(int IFormatHandle, INT_PTR pCurrentFormatDisk, INT_PTR pCurrentDiskPos, INT_PTR pFormatStatic)

参数: [in] IFormatHandle 格式化硬盘句柄, NET_DVR_FormatDisk 的返回值

[out] pCurrentFormatDisk 指向保存当前正在格式化的硬盘号的指针, 硬盘号从 0 开始, -1 为初始状态

[out] pCurrentDiskPos 指向保存当前正在格式化的硬盘的进度的指针, 进度是 0~100

[out] FormatStatic 指向保存硬盘格式化状态的指针: 0-正在格式化; 1-硬盘全部格式化完成; 2-格式化当前硬盘出错, 不能继续格式化此硬盘, 本地和网络硬盘都会出现此错误; 3-由于网络异常造成网络硬盘丢失而不能开始格式化当前硬盘

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: In class **com.hikvision.netsdk.HCNetSDK**, JNI 接口。

[返回目录](#)

3.15.3 关闭格式化硬盘句柄, 释放资源 **NET_DVR_CloseFormatHandle**

函数: public boolean NET_DVR_CloseFormatHandle(int IFormatHandle)

参数: [in] IFormatHandle 格式化硬盘句柄, NET_DVR_FormatDisk 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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.16 设备维护管理

设备工作状态

3.16.1 获取设备的工作状态 **NET_DVR_GetDVRWorkState_V30**

函数： public boolean NET_DVR_GetDVRWorkState_V30(int IUserID, NET_DVR_WORKSTATE_V30 lpWorkState)

参数： [in] IUserID NET_DVR_Login_V30 的返回值
[out] lpWorkState 获取的设备工作状态参数，详见：[NET_DVR_WORKSTATE_V30](#)

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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

UPNP 端口映射状态

3.16.2 获取 UPNP 端口映射状态 **NET_DVR_GetUpnpNatState**

函数： public boolean NET_DVR_GetUpnpNatState(int IUserID, NET_DVR_UPNP_NAT_STATE lpState)

参数： [in] IUserID NET_DVR_Login_V30 的返回
[out] lpState UPNP 端口映射状态，详见：[NET_DVR_UPNP_NAT_STATE](#)

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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

远程升级

3.16.3 设置远程升级时网络环境 **NET_DVR_SetNetworkEnvironment**

函数： public boolean NET_DVR_SetNetworkEnvironment(int 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: In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。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](#)

3.16.4 远程升级 `NET_DVR_Upgrade`

函数: `public int NET_DVR_Upgrade(int IUserID, String sFileName)`

参数: [in] IUserID `NET_DVR_Login_V30` 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

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: In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。

[返回目录](#)

3.16.5 获取远程升级的进度 `NET_DVR_GetUpgradeProgress`

函数: `public int NET_DVR_GetUpgradeProgress(int 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: In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。

[返回目录](#)

3.16.6 获取远程升级的状态 `NET_DVR_GetUpgradeState`

函数: `public int NET_DVR_GetUpgradeState(int 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: In class `com.hikvision.netsdk.HCNetSDK`, JNI 接口。

[返回目录](#)

3.16.7 获取远程升级的阶段信息 `NET_DVR_GetUpgradeStep`

函数: `public int NET_DVR_GetUpgradeStep(int IUpgradeHandle, INT_PTR pSubProgress)`

参数: [in] IUpgradeHandle `NET_DVR_Upgrade` 的返回值
[out] pSubProgress 升级阶段子进度, 0~100

返回值： -1 表示失败，其他值定义如表 3.26 所示。

表 3.26 升级阶段

返回值	含义
1	接收升级包数据
2	升级系统
3	备份系统
255	设备正在搜索升级文件

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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.16.8 关闭远程升级句柄，释放资源 NET_DVR_CloseUpgradeHandle

函数： public boolean NET_DVR_CloseUpgradeHandle(int 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： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

远程重启

3.16.9 重启设备 NET_DVR_RebootDVR

函数： public boolean NET_DVR_RebootDVR(int IUserID)

参数： [in] IUserID NET_DVR_Login_V30 的返回值

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

说明： In class com.hikvision.netsdk.HCNetSDK, JNI 接口。

[返回目录](#)

3.17 人脸比对功能

人脸比对功能

3.17.1 上传文件 **NET_DVR_UploadFile_V40**

函 数: public int NET_DVR_UploadFile_V40(int IUserID, int dwUploadType, Pointer lpInBuffer, int dwInBufferLen, String sFileName, Pointer lpOutBuffer, int dwOutBufferLen)

参 数: [in] IUserID NET_DVR_Login_V30 的返回值
[in] dwUploadType 上传文件类型，详细参见下文列表
[in] lpInBuffer 输入参数。不同的 dwUploadType，输入参数不同，具体参见下文列表
[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，具体参见下文列表
[out] dwOutBufferLen 输出缓冲区大小

dwUploadType 宏定义	宏定义值	dwUploadType 含义	lpInBuffer 对应结构体
IMPORT_DATA_TO_FACELIB	39	导入人脸数据(人脸图片+图片附加信息)到人脸库	NET_DVR_FACELIB_COND

返回值: -1 表示失败，其他值作为 NET_DVR_UploadSend、NET_DVR_UploadClose 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:

- **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**
- 上传接口调用流程:
 - 1) 调用 NET_DVR_UploadFile_V40 开始上传数据。
 - 2) 调用 NET_DVR_UploadSend 发送一张人脸图片数据和附加信息。
 - 3) 上传过程中循环调用 NET_DVR_GetUploadState 获取上传状态和进度。
 - 4) 上传成功之后调用 NET_DVR_GetUploadResult 获取结果信息（图片 ID 等）。
 - 5) 重复步骤 2、3、4，顺序上传其他人脸数据。
 - 6) 调用 NET_DVR_UploadClose 停止上传，释放资源。
- 设备是否支持导入人脸图片数据到人脸库能力，可以通过设备能力集进行判断，对应人脸对比库能力集(FDLIBCap)，相关接口: NET_DVR_STDXMLConfig，能力集获取命令: GET /ISAPI/Intelligent/FDLIB/capabilities，节点: <FaceAppendData>。

[返回目录](#)

3.17.2 发送数据 **NET_DVR_UploadSend**

函 数: public int NET_DVR_UploadSend(int IUploadHandle, Pointer pstruSendParamIN, Pointer lpOutBuffer)

参 数: [in] IUploadHandle 文件上传句柄，NET_DVR_UploadFile_V40 的返回值
[in] pstruSendParamIN 上传的数据内容
[out] lpOutBuffer 输出参数，保留，置为 NULL

返回值: -1 表示失败，大于等于 0 表示本次发送成功的字节数，未发送成功的剩余字节数可以再次调用该接口进行发送。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： In class `com.hikvision.netsdk.HCNetSDK`, **JNI 接口**。通过该接口将人脸数据(人脸图片+图片附件信息)发送到设备的人脸库。图片格式要求：JPG 或者 JPEG，像素在 40x40 以上，大小在 300KB 以下。

[返回目录](#)

3.17.3 获取文件上传的进度和状态 `NET_DVR_GetUploadState`

函数： `public int NET_DVR_GetUploadState(int IUploadHandle, IntByReference pProgress)`

参数： [in] `IUploadHandle` 文件上传句柄，`NET_DVR_UploadFile_V40` 的返回值
[out] `IntByReference` 返回的进度值，0~100

返回值： -1 表示函数调用失败，其他为上传的状态值见下表说明。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： In class `com.hikvision.netsdk.HCNetSDK`, **JNI 接口**。-1 表示函数调用失败，其他为上传的状态值见下表说明。

返回值	描述
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	文件个数超过最大值
11	文件过大
12	文件类型错误
19	文件格式不正确
20	文件内容不正确
21	上传音频采样率不支持
26	名称错误
27	图片分辨率无效错误
28	图片目标个数超过上限
30	图片数据识别失败
31	分析引擎异常
32	解析图片数据出错
34	安全校验密钥错误

[返回目录](#)

3.17.4 获取当前上传的结果信息 `NET_DVR_GetUploadResult`

函数： `public boolean NET_DVR_GetUploadResult(int IUploadHandle, Pointer lpOutBuffer, int`

dwOutBufferSize)
 参 数: [in] IUploadHandle 文件上传句柄, NET_DVR_UploadFile_V40 的返回值
 [in] lpOutBuffer 上传的数据内容
 [out] dwOutBufferSize 缓冲区大小

dwUploadType	功能描述	lpOutBuffer
IMPORT_DATA_TO_FACELIB	导入人脸数据(人脸图片+图片附加信息)到人脸库	NET_DVR_UPLOAD_FILE_RET

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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。** 上传时调用 NET_DVR_GetUploadState 获取进度和状态, 上传完成之后调用该接口获取当前结果信息, 然后再上传下一个。

[返回目录](#)

3.17.5 停止文件上传 NET_DVR_UploadClose

函 数: public boolean NET_DVR_UploadClose(int IUploadHandle)

参 数: [in] IUploadHandle 文件上传句柄, 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: **In class com.hikvision.netsdk.HCNetSDK, JNI 接口。**

[返回目录](#)

图片查找

3.17.6 查找图片 NET_DVR_FindPicture

函 数:	public int NET_DVR_FindPicture(int IUserID, Pointer pFindParam)	
参 数:	[in] IUserID	NET_DVR_Login_V30 的返回值
	[in] pFindParam	图片查找条件信息
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:	<ul style="list-style-type: none"> In class com.hikvision.netsdk.HCNetSDK, JNI 接口。 <p>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)。</p>	

[返回目录](#)

3.17.7 逐个获取查找到的图片信息 **NET_DVR_FindNextPicture_V40**

函 数:	public int NET_DVR_FindNextPicture_V40(int IFindHandle, Pointer lpFindData)	
参 数:	[in] IFindHandle [out] lpFindData	图片查找句柄，NET_DVR_FindPicture 的返回值 保存图片信息的指针
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:	In class com.hikvision.netsdk.HCNetSDK, JNI 接口。 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](#)

3.17.8 逐个获取查找到的图片信息 **NET_DVR_FindNextPicture_V40**

函 数:	public boolean NET_DVR_CloseFindPicture(int 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:	In class com.hikvision.netsdk.HCNetSDK, JNI 接口。	

[返回目录](#)

;

4 错误代码及说明

4.1 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_PLATE_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_RADAR_TYPE_CONFLICT	1601	相同 485 口关联雷达类型冲突

4.2 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_DESCRIBESENDDTIMEOUT	411	RTSP DESCRIBE 发送超时
NET_DVR_RTSP_DESCRIBESENDERROR	412	RTSP DESCRIBE 发送失败
NET_DVR_RTSP_DESCRIBERECDTIMEOUT	413	RTSP DESCRIBE 接收超时
NET_DVR_RTSP_DESCRIBERECDATALOST	414	RTSP DESCRIBE 接收数据错误
NET_DVR_RTSP_DESCRIBERECDERROR	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_SETUPSENDDTIMEOUT	421	RTSP SETUP 发送超时
NET_DVR_RTSP_SETUPSENDERROR	422	RTSP SETUP 发送错误
NET_DVR_RTSP_SETUPRECDTIMEOUT	423	RTSP SETUP 接收超时
NET_DVR_RTSP_SETUPRECVDATALOST	424	RTSP SETUP 接收数据错误
NET_DVR_RTSP_SETUPRECDError	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_PLAYSENDDTIMEOUT	431	RTSP PLAY 发送超时
NET_DVR_RTSP_PLAYSENDERROR	432	RTSP PLAY 发送错误
NET_DVR_RTSP_PLAYRECDTIMEOUT	433	RTSP PLAY 接收超时
NET_DVR_RTSP_PLAYRECVDATALOST	434	RTSP PLAY 接收数据错误
NET_DVR_RTSP_PLAYRECDError	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_TEARDOWNSENDDTIMEOUT	441	RTSP TEARDOWN 发送超时
NET_DVR_RTSP_TEARDOWNSENDERROR	442	RTSP TEARDOWN 发送错误
NET_DVR_RTSP_TEARDOWNRECDTIMEOUT	443	RTSP TEARDOWN 接收超时

NET_DVR_RTSP_TEARDOWNRECVDATALOST	444	RTSP TEARDOWN 接收数据错误
NET_DVR_RTSP_TEARDOWNRECVERERROR	445	RTSP TEARDOWN 接收失败
NET_DVR_RTSP_TEARDOWNSERVERERR	446	RTSP TEARDOWN 设备返回错误状态

4.3 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

5 结构体说明

5.1 EAP_PEAP:EAP_PEAP 认证参数

In class com.hikvision.netsdk.HCNetSDK

```
public class EAP_PEAP
{
    public byte    byEapolVersion;
    public byte    byAuthType;
    public byte    byPeapVersion;
    public byte    byPeapLabel;
    public byte[]  byAnonyIdentity = new byte[HCNetSDK.NAME_LEN];
    public byte[]  byUserName = new byte[HCNetSDK.NAME_LEN];
    public byte[]  byPassword = new byte[HCNetSDK.NAME_LEN];
}
```

Members

byEapolVersion

EAPOL 版本: 0- 版本 1, 1- 版本 2

byAuthType

内部认证方式: 0- GTC, 1- MD5, 2- MSCHAPV2

byPeapVersion

PEAP 版本: 0- 版本 0, 1- 版本 1

byPeapLabel

PEAP 标签: 0- 老标签, 1- 新标签

byAnonyIdentity

匿名身份

byUserName

用户名

byPassword

密码

5.2 EAP_TLS:EAP_TLS 认证参数

In class com.hikvision.netsdk.HCNetSDK

```
public class EAP_TLS
{
    public byte    byEapolVersion;
    public byte[]  byIdentity = new byte[HCNetSDK.NAME_LEN];
    public byte[]  byPrivateKeyPswd = new byte[HCNetSDK.NAME_LEN];
}
```


Members

byEapolVersion

EAPOL 版本: 0- 版本 1, 1- 版本 2

byIdentity

身份

byPrivateKeyPswd

私钥密码

5.3 EAP_TTLS:EAP_TTLS 认证参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class EAP_TTLS
{
    public byte    byEapolVersion;
    public byte    byAuthType;
    public byte[]  byAnonyIdentity = new byte[HCNetSDK.NAME_LEN];
    public byte[]  byUserName = new byte[HCNetSDK.NAME_LEN];
    public byte[]  byPassword = new byte[HCNetSDK.NAME_LEN];
}
```

Members

byEapolVersion

EAPOL 版本: 0- 版本 1, 1- 版本 2

byAuthType

内部认证方式: 0- PAP, 1- MSCHAPV2

byRes1

保留, 置为 0

byAnonyIdentity

匿名身份

byUserName

用户名

byPassword

密码

5.4 NET_DVR_ACTIVATECFG:激活参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_ACTIVATECFG
{
    public byte[] sPassword = new byte[HCNetSDK.PASSWD_LEN];
}
```

Members

sPassword

初始密码，密码等级弱或者以上

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：
 - 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++。

5.5 NET_DVR_ALARMER:报警设备信息

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_DVR_ALARMER extends Structure
{
    public byte    byUserIDValid;
    public byte    bySerialValid;
    public byte    byVersionValid;
    public byte    byDeviceNameValid;
    public byte    byMacAddrValid;
    public byte    byLinkPortValid;
    public byte    byDeviceIPValid;
    public byte    bySocketIPValid;
    public int     lUserID;
    public byte[]  sSerialNumber = new byte[SERIALNO_LEN];
    public int     dwDeviceVersion;
    public byte[]  sDeviceName = new byte[NAME_LEN];
    public byte[]  byMacAddr = new byte[MACADDR_LEN];
    public short   wLinkPort;
    public byte[]  sDeviceIP = new byte[128];
    public byte[]  sSocketIP = new byte[128];
    public byte    byIpProtocol;
    public byte[]  byRes2 = new byte[11];

    protected List getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("byUserIDValid", "bySerialValid", "byVersionValid", "byDeviceNameValid",
            "byMacAddrValid", "byLinkPortValid", "byDeviceIPValid", "bySocketIPValid", "lUserID", "sSerialNumber",
            "dwDeviceVersion", "sDeviceName", "byMacAddr", "wLinkPort", "sDeviceIP", "sSocketIP",
            "byIpProtocol", "byRes2");
    }
}
```

```

    }
}

```

Members

byUserIDValid

userid 是否有效: 0- 无效, 1- 有效

bySerialValid

序列号是否有效: 0- 无效, 1- 有效

byVersionValid

版本号是否有效: 0- 无效, 1- 有效

byDeviceNameValid

设备名字是否有效: 0- 无效, 1- 有效

byMacAddrValid

MAC 地址是否有效: 0- 无效, 1- 有效

byLinkPortValid

Login 端口是否有效: 0- 无效, 1- 有效

byDeviceIPValid

设备 IP 是否有效: 0- 无效, 1- 有效

bySocketIPValid

Socket IP 是否有效: 0- 无效, 1- 有效

lUserID

NET_DVR_Login 或 NET_DVR_Login_V30 返回值, 布防时有效

sSerialNumber

序列号

dwDeviceVersion

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

sDeviceName

设备名称

byMacAddr

MAC 地址

wLinkPort

设备通讯端口

sDeviceIP

设备 IP 地址

sSocketIP

报警主动上传时的 Socket IP 地址

byIpProtocol

IP 协议: 0- IPV4, 1- IPV6

byRes2

保留, 置为 0

5.6 NET_DVR_ALARMINGCFG_V30:报警输入参数

In class com.hikvision.netsdk.HCNetSDK

```

public class NET_DVR_ALARMINGCFG_V30 extends NET_DVR_CONFIG
{
    public byte[]                sAlarmInName = new byte[HCNetSDK.NAME_LEN];
    public byte                  byAlarmType;
    public byte                  byAlarmInHandle;
    public byte                  byChannel;
    public NET\_DVR\_HANDLEEXCEPTION\_V30 struAlarmHandleType = new NET_DVR_HANDLEEXCEPTION_V30();
    public NET\_DVR\_SCHEDTIME[][] struAlarmTime = new
NET_DVR_SCHEDTIME[HCNetSDK.MAX_DAYS][HCNetSDK.MAX_TIMESEGMENT_V30];
    public byte[]                byRelRecordChan = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]                byEnablePreset = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]                byPresetNo = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]                byEnableCruise = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]                byCruiseNo = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]                byEnablePtzTrack = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]                byPTZTrack = new byte[HCNetSDK.MAX_CHANNUM_V30];

    public NET_DVR_ALARMINGCFG_V30()
    {
        for(int i=0; i<HCNetSDK.MAX_DAYS; i++)
        {
            for(int j=0; j<HCNetSDK.MAX_TIMESEGMENT_V30; j++)
            {
                struAlarmTime[i][j] = new NET_DVR_SCHEDTIME();
            }
        }
    }
}

```

Members

sAlarmInName

报警输入名称

byAlarmType

报警器类型：0-常开，1-常闭

byAlarmInHandle

是否处理：0-不处理，1-处理

byChannel

报警输入触发智能识别通道

struAlarmHandleType

处理方式

struAlarmTime

布防时间参数

byRelRecordChan

报警触发的录象通道，为 1 表示触发该通道

byEnablePreset

通道是否启动调用预置点（每个数组对应一个通道）：0-否，1-是

byPresetNo

通道调用的云台预置点序号，一个报警输入可以调用多个通道的云台预置点（每个通道仅限一个预置点）

byEnableCruise

是否调用巡航：0-否，1-是

byCruiseNo

巡航路径

byEnablePtzTrack

是否调用花样扫描：0-否，1-是

byPTZTrack

调用的云台的花样扫描序号

5.7 NET_DVR_ALARMINFO:设备报警信息

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_DVR_ALARMINFO extends Structure
{
    public int    dwAlarmType;
    public int    dwAlarmInputNumber;
    public int[]  dwAlarmOutputNumber = new int[MAX_ALARMOUT];
    public int[]  dwAlarmRelateChannel = new int[MAX_CHANNUM];
    public int[]  dwChannel = new int[MAX_CHANNUM];
    public int[]  dwDiskNumber = new int[MAX_DISKNUM] ;

    @Override
    protected List getFieldOrder() {
        //字段顺序必须跟C++结构体参数顺序一致
        return Arrays.asList("dwAlarmType", "dwAlarmInputNumber", "dwAlarmOutputNumber",
            "dwAlarmRelateChannel", "dwChannel", "dwDiskNumber");
    }
}
```

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

dwAlarmInputNumber

报警输入端口。当报警类型为 9 时该变量表示串口状态：0 表示正常，0xffffffff 表示异常

dwAlarmOutputNumber

触发的报警输出端口。当报警类型为信号量报警时，值为 1 表示该报警端口输出

dwAlarmRelateChannel

触发的录像通道。当报警类型为信号量报警时，值为 1 表示该通道录像，如 dwAlarmRelateChannel[0] 表示触发第 1 个通道录像

dwChannel

发生报警的通道。当报警类型为 2、3、6 时有效，如 `dwChannel[0]` 值为 1 表示第 1 个通道报警

dwDiskNumber

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

5.8 NET_DVR_ALARMINFO_V30:上传的报警信息 V30

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

public static class `NET_DVR_ALARMINFO_V30` extends `Structure`

```
{
    public int      dwAlarmType;
    public int      dwAlarmInputNumber;
    public byte[]   byAlarmOutputNumber = new byte[MAX_ALARMOUT_V30];
    public byte[]   byAlarmRelateChannel = new byte[MAX_CHANNUM_V30];
    public byte[]   byChannel = new byte[MAX_CHANNUM_V30];
    public byte[]   byDiskNumber = new byte[MAX_DISKNUM_V30];

    public NET_DVR_ALARMINFO_V30(Pointer p) {
        super(p);
    }
    @Override
    protected List getFieldOrder() {
        //字段顺序必须跟C++结构体参数顺序一致
        return Arrays.asList("dwAlarmType", "dwAlarmInputNumber", "byAlarmOutputNumber",
            "byAlarmRelateChannel", "byChannel", "byDiskNumber");
    }
}
```

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

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 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

5.9 NET_DVR_ALARMINFO_V40:上传的报警信息 V40

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_DVR_ALARMINFO_V40 extends Structure
{
    public NET\_DVR\_ALRAM\_FIXED\_HEADER struAlarmFixedHeader = new
NET_DVR_ALRAM_FIXED_HEADER();
    public Pointer
                                pAlarmData;

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟C++结构体参数顺序一致
        return Arrays.asList("struAlarmFixedHeader", "pAlarmData");
    }
}
```

Members

struAlarmFixedHeader

报警固定部分

pAlarmData

报警可变部分内容，不同的报警类型对应不同的取值，详见表 5.1

Remarks

不同的报警信息类型（通过报警固定部分里面的参数进行判断），报警可变部分内容不同，详见表 5.1。

表 5.1 可变报警信息

struAlarmFixedHeader 中 dwAlarmType 取值	含义	pAlarmData 对应内容
0	信号量报警	dwTrigerAlarmOutNum*(DWORD)报警输出 口号+dwTrigerRecordChanNum*(DWORD)通 道号
2、3、6、9、10、11、13、 15、16	信号丢失、移动侦测、遮挡报警、视频 信号异常、录像异常、智能场景变化、 前端/录像分辨率不匹配、智能侦测、POE 供电异常	dwAlarmChanNum*(DWORD)通道号
1、4、5	硬盘满、硬盘未格式化、写硬盘出错	dwAlarmHardDiskNum*(DWORD)硬盘号
7、8、12、17、18	制式不匹配、非法访问、阵列异常、录 播主机报警、TME 语音对讲请求报警	无

5.10 NET_DVR_ALARMOUTCFG_V30:报警输出参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_ALARMOUTCFG_V30 extends NET_DVR_CONFIG
```

```

{
    public byte[]                sAlarmOutName = new byte[HCNetSDK.NAME_LEN];
    public int                   dwAlarmOutDelay;
    public NET_DVR_SCHEDTIME[][] struAlarmOutTime = new
NET_DVR_SCHEDTIME[HCNetSDK.MAX_DAYS][HCNetSDK.MAX_TIMESEGMENT_V30];
    public NET_DVR_ALARMOUTCFG_V30()
    {
        for(int i=0; i<HCNetSDK.MAX_DAYS; i++)
        {
            for(int j=0; j<HCNetSDK.MAX_TIMESEGMENT_V30; j++)
            {
                struAlarmOutTime[i][j] = new NET_DVR_SCHEDTIME();
            }
        }
    }
}

```

Members

sAlarmOutName

报警输出名称

dwAlarmOutDelay

输出信号状态维持时间，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，7-手动（需手动关闭），传入其他不正确的值设备默认为 5 分钟

struAlarmOutTime

报警输出激活时间段

5.11 NET_DVR_ALARMOUTSTATUS_V30:报警输出状态

In class `com.hikvision.netsdk.HCNetSDK`

```

public class NET_DVR_ALARMOUTSTATUS_V30 extends NET_DVR_CONFIG
{
    public byte[] Output = new byte[HCNetSDK.MAX_ALARMOUT_V30];
}

```

Members

Output

报警输出口的状态，数组下标表示输出口号，值：0-无效，1-有效

5.12 NET_DVR_ALRAM_FIXED_HEADER:报警设备信息

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_DVR_ALRAM_FIXED_HEADER extends Structure
{
    public int                dwAlarmType;
}

```



```
public NET\_DVR\_TIME\_EX    struAlarmTime = new NET_DVR_TIME_EX();
public uStruAlarm          ustruAlarm = new uStruAlarm();
```

```
@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("dwAlarmType", "struAlarmTime", "ustruAlarm");
}
}
```

Members

dwAlarmType

报警信息类型：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-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，15-智能侦测，16-POE 供电异常，17-录播主机报警，18-TME 语音对讲请求报警

struAlarmTime

报警时间

uStruAlarm

报警信息联合体，dwAlarmType 为 0 对应联合体中结构 struioAlarm，dwAlarmType 为 2、3、6、9、10、11、13、15、16 对应联合体中结构 sstrualarmChannel，dwAlarmType 为 1、4、5 对应联合体中结构 strualarmHardDisk，dwAlarmType 为 17 对应联合体中结构 strurecordingHost，其他取值时联合体参数无效

5.13 NET_DVR_AP_INFO:单个无线资源参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_AP_INFO
{
    public byte[]    sSsid = new byte[HCNetSDK.IW_ESSID_MAX_SIZE];
    public int       dwMode;
    public int       dwSecurity;
    public int       dwChannel;
    public int       dwSignalStrength;
    public int       dwSpeed;
}
```

Members

sSsid

SSID

dwMode

工作模式：0-mange 模式，1-ad-hoc 模式

dwSecurity

安全模式：0 不加密，1 wep 加密，2 wpa-psk，3 wpa-Enterprise

dwChannel

通道号

dwSignalStrength

0-100 信号由最弱变为最强

dwSpeed

速率，单位是 0.01mbps

5.14 NET_DVR_AP_INFO_LIST:无线网络资源列表

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_AP_INFO_LIST extends NET_DVR_CONFIG
{
    public int                dwCount;
    public NET\_DVR\_AP\_INFO[] struApInfo = new NET_DVR_AP_INFO[HCNetSDK.WIFI_MAX_AP_COUNT];
    public NET_DVR_AP_INFO_LIST()
    {
        for(int i = 0; i < HCNetSDK.WIFI_MAX_AP_COUNT; i++)
        {
            struApInfo[i] = new NET_DVR_AP_INFO();
        }
    }
}
```

Members

dwCount

无线 AP 数量，不超过 20

struApInfo

AP 参数信息

5.15 NET_DVR_AREAINFOCFG:籍贯参数结构体

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_AREAINFOCFG
{
    WORD    wNationalityID;
    WORD    wProvinceID;
    WORD    wCityID;
    WORD    wCountyID;
    DWORD   dwCode;
}
```

Members

wNationalityID

国籍

wProvinceID

省，详见枚举类型：PROVINCE_CITY_IDX

```
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

0- 安徽

AOMEN_PROVINCE

1- 澳门

BEIJING_PROVINCE

2- 北京
CHONGQING_PROVINCE
3- 重庆
FUJIAN_PROVINCE
4- 福建
GANSU_PROVINCE
5- 甘肃
GUANGDONG_PROVINCE
6- 广东
GUANGXI_PROVINCE
7- 广西
GUIZHOU_PROVINCE
8- 贵州
HAINAN_PROVINCE
9- 海南
HEBEI_PROVINCE
10- 河北
HENAN_PROVINCE
11- 河南
HEILONGJIANG_PROVINCE
12- 黑龙江
HUBEI_PROVINCE
13- 湖北
HUNAN_PROVINCE
14- 湖南
JILIN_PROVINCE
15- 吉林
JIANGSU_PROVINCE
16- 江苏
JIANGXI_PROVINCE
17- 江西
LIAONING_PROVINCE
18- 辽宁
NEIMENGGU_PROVINCE
19- 内蒙古
NINGXIA_PROVINCE
20- 宁夏
QINGHAI_PROVINCE
21- 青海
SHANDONG_PROVINCE
22- 山东
SHANXI_JIN_PROVINCE
23- 山西
SHANXI_SHAN_PROVINCE

24- 陕西
SHANGHAI_PROVINCE
 25- 上海
SICHUAN_PROVINCE
 26- 四川
TAIWAN_PROVINCE
 27- 台湾
TIANJIN_PROVINCE
 28- 天津
XIZANG_PROVINCE
 29- 西藏
XIANGGANG_PROVINCE
 30- 香港
XINJIANG_PROVINCE
 31- 新疆
YUNNAN_PROVINCE
 32- 云南
ZHEJIANG_PROVINCE
 33- 浙江

wCityID

市

wCountyID

县

dwCode

国家标准的省份、城市、县级代码，为 0 表示设备不支持

5.16 NET_DVR_CHANNELSTATE_V30:通道状态

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_CHANNELSTATE_V30
{
    public byte                byRecordStatic;
    public byte                bySignalStatic;
    public byte                byHardwareStatic;
    public int                 dwBitRate;
    public int                 dwLinkNum;
    public NET\_DVR\_IPADDR[] struClientIP = new NET_DVR_IPADDR[HCNetSDK.MAX_LINK];
    public int                 dwIPLinkNum;
    public byte                byExceedMaxLink;

    public NET_DVR_CHANNELSTATE_V30(){
        for(int i=0; i<HCNetSDK.MAX_LINK; i++){
            struClientIP[i] = new NET_DVR_IPADDR();
        }
    }
}
```

```

    }
}

```

Members

byRecordStatic

通道是否在录像：0- 不录像，1- 录像

bySignalStatic

连接的信号状态：0- 正常，1- 信号丢失

byHardwareStatic

通道硬件状态：0- 正常，1-异常（例如 DSP 异常）

dwBitRate

实际码率

dwLinkNum

连接该通道的客户端个数

struClientIP

连接该通道的客户端 IP 地址

dwIPLinkNum

如果该通道为 IP 接入，表示该 IP 接入当前的连接数

byExceedMaxLink

是否超出了单路 6 路连接数 0 - 未超出, 1-超出

5.17 NET_DVR_CLIENTINFO:预览参数

In class com.hikvision.netsdk.HCNetSDK

```

public class NET_DVR_CLIENTINFO
{
    public int      IChannel;
    public int      ILinkMode;
    public String   sMultiCastIP;
}

```

Members

IChannel

通道号。如果是零通道，目前设备只支持一个零通道，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

例如：子码流 TCP 连接，则 ILinkMode=0x80000000

sMultiCastIP

多播组地址

5.18 NET_DVR_COMPRESSION_AUDIO:语音对讲音频参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_COMPRESSION_AUDIO
{
    byte    byAudioEncType;
    byte[]  byres = new byte[7];
}
```

Members

byAudioEncType

音频编码类型：0- G722，1- G711_U，2- G711_A，5- MP2L2，6- G726，7- AAC，8- PCM

byres

保留，置为 0

Remarks

配置设备的语音对讲音频参数后需要重启设备才生效。如果 `NET_DVR_GetDVRConfig` 返回失败，获取错误号为 23（设备不支持），则表示设备音频编码类型为 G722。

5.19 NET_DVR_COMPRESSION_INFO_V30:码流压缩参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_DVR_COMPRESSION_INFO_V30 extends Structure
{
```

```
    public byte    byStreamType;
    public byte    byResolution;
    public byte    byBitrateType;
    public byte    byPicQuality;
    public int     dwVideoBitrate;
    public int     dwVideoFrameRate;
    public short   wIntervalFrameI;
    public byte    byIntervalBPFrame;
    public byte    byENumber;
    public byte    byVideoEncType;
    public byte    byAudioEncType;
    public byte[]  byRes = new byte[10];
```

@Override

```
protected List<String> getFieldOrder() {
    //字段顺序必须跟C++结构体参数顺序一致
    return Arrays.asList("byStreamType", "byResolution", "byBitrateType", "byPicQuality",
        "dwVideoBitrate", "dwVideoFrameRate", "wIntervalFrameI", "byIntervalBPFrame",
        "byENumber", "byVideoEncType", "byAudioEncType", "byres");
}
```

```
}
```

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_COMPRESSION_INFO_V30
{
    public byte    byStreamType;
    public byte    byResolution;
    public byte    byBitrateType;
    public byte    byPicQuality;
    public int     dwVideoBitrate;
    public int     dwVideoFrameRate;
    public short   wIntervalFrameI;
    public byte    byIntervalBPFrame;
    public byte    byENumber;
    public byte    byVideoEncType;
    public byte    byAudioEncType;
    public byte[]  byRes = new byte[10];
}
```

Members

byStreamType

码流类型：0-视频流，1-复合流，0xfe- 自动（和源一致）

如果是事件压缩参数(struEventRecordPara，需要设备支持)，最高位(byStreamType & 0x80)表示是否启用事件压缩参数，即配置 struEventRecordPara 时：

byStreamType&0x80 == 0 表示禁用事件压缩参数；

(byStreamType&0x80 == 1)&&(byStreamType&0x7f == 0) 表示启用事件压缩参数并且码流类型设置为视频流；

(byStreamType&0x80 == 1)&&(byStreamType&0x7f == 1) 表示启用事件压缩参数并且码流类型设置为复合流；

byStreamType==0xfe 表示启用事件压缩参数并且码流类型设置为和源一致。

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)，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)

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，0xfe- 自动（和源一致）

dwVideoBitrate

视频码率：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，0xffffffff- 自动（和源一致）。

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

视频帧率：0-全部，1-1/16，2-1/8，3-1/4，4-1/2，5-1，6-2，7-4，8-6，9-8，10-10，11-12，12-16，13-20，14-15，15-18，16-22，17-25，18-30，19-35，20-40，21-45，22-50，23-55，24-60，25-3，26-5，27-7，28-9，29-100，30-120，31-24，32-48，0xffffffff- 自动（和源一致）

IntervalFrameI

I 帧间隔，0xfffe- 自动（和源一致），0xffff-无效

byIntervalBPFrame

视频帧格式：0-BBP 帧，1-BP 帧，2-单 P 帧，0xff-无效

byENumber

E 帧个数

byVideoEncType

视频编码类型：0-私有 264，1-标准 h264，2-标准 mpeg4，7-M-JPEG，8-MPEG2，9-SVAC，10-标准 h265，0xfe- 自动（和源一致），0xff-无效

byAudioEncType

音频编码类型：0-G722，1-G711_U，2-G711_A，5-MP2L2，6-G726，7-AAC，8-PCM，0xfe- 自动（和源一致），0xff-无效

byRes

保留，置为 0

5.20 NET_DVR_COMPRESSIONCFG_V30:通道压缩参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_COMPRESSIONCFG_V30 extends NET_DVR_CONFIG
{
    public NET\_DVR\_COMPRESSION\_INFO\_V30 struNormHighRecordPara = new
NET_DVR_COMPRESSION_INFO_V30();
    public NET\_DVR\_COMPRESSION\_INFO\_V30 struEventRecordPara = new
NET_DVR_COMPRESSION_INFO_V30();
    public NET\_DVR\_COMPRESSION\_INFO\_V30 struNetPara = new NET_DVR_COMPRESSION_INFO_V30();
    public NET\_DVR\_COMPRESSION\_INFO\_V30 struRes = new NET_DVR_COMPRESSION_INFO_V30();
}
```

Members

struNormHighRecordPara

录像的码流压缩参数（即主码流的压缩参数）

struEventRecordPara

事件触发压缩参数，报警事件触发后，主码流切换成事件压缩参数配置

struNetPara

网传的码流压缩参数（即子码流的压缩参数）

struRes

保留，置为 0

5.21 NET_DVR_DDNSPARA_V30:DDNS 配置参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_DDNSPARA_V30 extends NET_DVR_CONFIG
{
    public byte                byEnableDDNS;
    public byte                byHostIndex;
    public byte[]              byRes1 = new byte[2];
    public NET\_DVR\_SINGLE\_DDNS[] struDDNS = new NET_DVR_SINGLE_DDNS[HCNetSDK.MAX_DDNS_NUMS];
    public byte[]              byRes2 = new byte[16];

    public NET_DVR_DDNSPARA_V30(){
        for(int i = 0; i < HCNetSDK.MAX_DDNS_NUMS; i++)
        {
            struDDNS[i] = new NET_DVR_SINGLE_DDNS();
        }
    }
}
```

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

struDDNS

DDNS 服务器参数

byRes

保留，置为 0

5.22 NET_DVR_DECODERCFG_V30:云台解码器参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_DECODERCFG_V30 extends NET_DVR_CONFIG
{
    public int         dwBaudRate;
    public byte        byDataBit;
    public byte        byStopBit;
    public byte        byParity;
    public byte        byFlowcontrol;
    public short       wDecoderType;
    public short       wDecoderAddress;
    public byte[]      bySetPreset = new byte[HCNetSDK.MAX_PRESET_V30];
    public byte[]      bySetCruise = new byte[HCNetSDK.MAX_CRUISE_V30];
    public byte[]      bySetTrack = new byte[HCNetSDK.MAX_TRACK_V30];
}
```

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

wDecoderType

解码器类型, 通过 NET_DVR_GetPTZProtocol 获取, 该值对应于结构 NET_DVR_PTZ_PROTOCOL 中的 dwType

wDecoderAddress

解码器地址: [0,255]

bySetPreset

预置点是否设置: 0-没有设置, 1-设置

bySetCruise

巡航是否设置: 0-没有设置, 1-设置

bySetTrack

花样扫描是否设置: 0-没有设置, 1-设置

5.23 NET_DVR_DEVICECFG_V40:设备参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_DEVICECFG_V40 extends NET_DVR_CONFIG{
    public byte[] byDevTypeName = new byte[24];
}
```

Members

byDevTypeName

（只读，不可修改）设备型号名称

5.24 NET_DVR_DEVICEINFO_V30:设备信息

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_DEVICEINFO_V30
{
    public byte[] sSerialNumber = new byte[SERIALNO_LEN];
    public byte byAlarmInPortNum;
    public byte byAlarmOutPortNum;
    public byte byDiskNum;
    public byte byDVRType;
    public byte byChanNum;
    public byte byStartChan;
    public byte byAudioChanNum;
    public byte byIPChanNum;
    public byte byZeroChanNum;
    public short wDevType;
    public byte byStartDChan
    public byte byHighDChanNum
}
```

Members

sSerialNumber

设备序列号

byAlarmInPortNum

报警输入个数

byAlarmOutPortNum

报警输出个数

byDiskNum

硬盘个数

byDVRType

设备类型

byChanNum

设备模拟通道个数

byStartChan

模拟通道起始通道号

byAudioChanNum

设备语音通道数

byIPChanNum

设备最大数字通道个数，低 8 位

byZeroChanNum

零通道个数

wDevType

设备类型

byStartDChan

起始数字通道号

byHighDChanNum

数字通道个数，高 8 位

Members

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
DVR	1	尚未定义的 DVR 类型
ATMDVR	2	ATM DVR
DVS	3	DVS
DEC	4	6001D
ENC_DEC	5	6001F
DVR_HC	6	8000HC
DVR_HT	7	8000HT
DVR_HF	8	8000HF
DVR_HS	9	8000HS DVR(no audio)
DVR_HTS	10	8016HTS DVR(no audio)
DVR_HB	11	HB DVR(SATA HD)
DVR_HCS	12	8000HCS DVR
DVS_A	13	带 ATA 硬盘的 DVS
DVR_HC_S	14	8000HC-S
DVR_HT_S	15	8000HT-S
DVR_HF_S	16	8000HF-S
DVR_HS_S	17	8000HS-S
ATMDVR_S	18	ATM-S
DVR_7000H	19	7000H 系列
DEC_MAT	20	多路解码器
DVR_MOBILE	21	mobile DVR
DVR_HD_S	22	8000HD-S

DVR_HD_SL	23	8000HD-SL
DVR_HC_SL	24	8000HC-SL
DVR_HS_ST	25	8000HS_ST
DVS_HW	26	6000HW
DS630X_D	27	多路解码器
DS640X_HD	28	640X 高清解码器
DS610X_D	29	610X 解码器
IPCAM	30	网络摄像机
MEGA_IPCAM	31	高清网络摄像机
IPCAM_X62MF	32	X62MF 系列摄像机
ITCCAM	35	智能交通摄像机
IVS_IPCAM	36	智能分析高清网络摄像机(人脸抓拍机)
ZOOMCAM	38	一体机
IPDOME	40	IP 标清快球
IPDOME_MEGA200	41	IP 200 万高清快球
IPDOME_MEGA130	42	IP 130 万高清快球
TII_IPCAM	44	红外热成像摄像机
IPMOD	50	IP 模块
IDS6501_HF_P	60	6501 车牌识别
IDS6101_HF_A	61	智能 ATM
IDS6002_HF_B	62	双摄像机跟踪: DS6002-HF/B
IDS6101_HF_B	63	异常行为识别: DS6101-HF/B
IDS52XX	64	智能分析仪
IDS90XX	65	9000 智能
IDS8104_AHL_S_HX	66	海鑫人脸识别 ATM
IDS8104_AHL_S_H	67	私有人脸识别 ATM
IDS91XX	68	9100 智能
IIP_CAM_B	69	智能行为 IP 摄像机
IIP_CAM_F	70	智能人脸 IP 摄像机
DS71XX_H	71	DS71XXH_S
DS72XX_H_S	72	DS72XXH_S
DS73XX_H_S	73	DS73XXH_S
DS72XX_HF_S	74	DS72XX_HF_S
DS73XX_HFI_S	75	DS73XX_HFI_S

DS76XX_H_S	76	DS76XX_H_S
DS76XX_N_S	77	DS76XX_N_S
DS81XX_HS_S	81	DS81XX_HS_S
DS81XX_HL_S	82	DS81XX_HL_S
DS81XX_HC_S	83	DS81XX_HC_S
DS81XX_HD_S	84	DS81XX_HD_S
DS81XX_HE_S	85	DS81XX_HE_S
DS81XX_HF_S	86	DS81XX_HF_S
DS81XX_AH_S	87	DS81XX_AH_S
DS81XX_AHF_S	88	DS81XX_AHF_S
DS90XX_HF_S	90	DS90XX_HF_S
DS91XX_HF_S	91	DS91XX_HF_S
DS91XX_HD_S	92	91XXHD-S(MD)
IDS90XX_A	93	9000 智能 ATM
IDS91XX_A	94	9100 智能 ATM
DS95XX_N_S	95	DS95XXN-S NVR
DS96XX_N_SH	96	DS96XXN-SH NVR
DS90XX_HF_SH	97	DS90XX_HF_SH
DS91XX_HF_SH	98	DS91XX_HF_SH
DS_65XXHC	105	65XXHC DVS
DS_65XXHC_S	106	65XXHC-SATA DVS
DS_65XXHF	107	65XXHF DVS
DS_65XXHF_S	108	65XXHF-SATA DVS
DS_6500HF_B	109	65 rack DVS
IVMS_6200_C	110	iVMS-6200(/C) 客流量统计
IVMS_6200_B	111	IVMS_6200_B 异常行为识别
DS_72XXHV_ST15	112	72XXHV_ST15 DVR
DS_72XXHV_ST20	113	72XXHV_ST20 DVR
IVMS_6200_T	114	IVMS-6200(/T)
IVMS_6200_BP	115	IVMS-6200(/BP)
DS_81XXHC_ST	116	DS_81XXHC_ST
DS_81XXHS_ST	117	DS_81XXHS_ST
DS_81XXAH_ST	118	DS_81XXAH_ST
DS_81XXAHF_ST	119	DS_81XXAHF_ST

DS_66XXDVS	120	DS_66XXDVS
DS_19AXX	142	通用报警主机类产品
DS_19CXX	144	自助银行报警主机
DS_19DXX	145	动环监控报警主机
DS_19XX	146	1900 系列报警主机
DS_19SXX	147	视频报警主机
DS_1HXX	148	ATM 防护舱控制器
DS_C10H	161	多屏控制器
DS_C10N_BI	162	BNC 处理器
DS_C10N_DI	163	RGB 处理器
DS_C10N_SI	164	码流处理器
DS_C10N_DO	165	显示处理器
DS_C10N_SERVER	166	分布式服务器
IDS_8104_AHFL_S_H	171	8104ATM
IDS_65XX_HF_A	172	65 ATM
IDS90XX_HF_RH	173	9000 智能 RH
IDS91XX_HF_RH	174	9100 智能 RH 设备
IDS_65XX_HF_B	175	65 异常行为识别
IDS_65XX_HF_P	176	65 车牌识别
IVMS_6200_F	177	IVMS-6200(/F)人脸分析仪
IVMS_6200_F_S	179	IVMS-6200(/F_S)人脸后检索分析仪
DS90XX_HF_RH	181	DS90XX_HF_RH
DS91XX_HF_RH	182	9100 RH 设备
DS78XX_S	183	78 系列设备
DS81XXHW_S	185	DVR_81XXHW_S
DS81XXHW_ST	186	DVR_81XXHW_ST
DS91XXHW_ST	187	DVR_91XXHW_ST
DS91XX_ST	188	DVR_91XX_ST
DS81XX_ST	189	DVR_81XX_ST
DS81XXH_ST	190	DS81XXHDI_ST,DS81XXHE_ST
DS73XXH_ST	191	DS73XXHI_ST
DS81XX_SH	192	审讯 81SH, 81SHF
DS81XX_SN	193	审讯 81SNL
DS96XXN_ST	194	NVR: DS96xxN_ST

DS86XXN_ST	195	NVR: DS86xxN_ST
DS80XXHF_ST	196	DS80xxHF_ST
DS90XXHF_ST	197	DS90xxHF_ST
DS76XXN_ST	198	NVR: DS76xxN_ST
DS_9664N_RX	199	NVR: DS-9664N-RH、DS-9664N-RT
ENCODER_SERVER	200	编码卡服务器
DECODER_SERVER	201	解码卡服务器
PCNVR_SERVER	202	PCNVR 存储服务器
CVR_SERVER	203	CVR
DS_91XXHFH_ST	204	高清 DVR: DS_91xxHFH_ST
DS_66XXHFH	205	66 高清编码器
TRAFFIC_TS_SERVER	210	终端服务器
TRAFFIC_VAR	211	视频分析记录仪
IPCALL	212	IP 可视对讲主机
DS64XXHD_T	701	64-T 高清解码器
DS_65XXD	703	65 万能解码器
DS63XXD_T	704	63-T 标清解码器
DS_64XXHD_S	706	DS-64xxHD-S 高清解码器
DS_68XXT	707	多功能视音频转码器
DS_65XXD_T	708	65D-T 万能解码器
IPCAM_FISHEYE	1002	鱼眼摄像机
TRAFFIC_ECT	1400	出入口终端服务器
TRAFFIC_PARKING_SERVER	1401	停车场服务器
DS90XXHW_ST	2001	混合 DVR: DS-90xxHW-ST
DS72XXHX_SH	2002	DS-72xxHV-SH, DS-72xxHF-SH
DS_92XX_HF_ST	2003	DS-92xxHF-ST
DS_91XX_HF_XT	2004	Netra DVR: DS-91xxHF-XT
DS_90XX_HF_XT	2005	Netra 混合 DVR: DS-90xxHF-XT
DS_73XXHX_SH	2006	Netra DVR: DS-73xxHX-SH
DS_72XXHFH_ST	2007	Netra DVR: DS-72xxHFH-ST
DS_67XXHF_SATA	2008	DVS: DS-67xxHF-SATA
DS_67XXHW	2009	DVS: DS-67xxHW
DS_67XXHW_SATA	2010	DVS: DS-67xxHW-SATA
DS_67XXHF	2011	DVS: DS-67xxHF

DS_72XXHF_SV	2012	DVR: DS-72xxHF-SV
DS_72XXHW_SV	2013	DVR: DS-72xxHW-SV
DS_81XXHX_SH	2014	DVR: DS-81xxHX-SH
DS_71XXHX_SL	2015	DVR: DS-71xxHX-SL
DS_77XXN_ST	2201	Netra NVR: DS-77xxN-ST
DS_95XX_N_ST	2202	Netra NVR: DS-95xxN-ST
DS_85XX_N_ST	2203	Netra NVR: DS-85xxN-ST
DS_96XX_N_XT	2204	Netra NVR: DS-96xxN-XT
DS_76XX_N_SE	2205	Netra NVR: DS-76xxN-SE、DS-78xxN-SH
DS_86XXSN_SX	2206	Netra 审讯 NVR, 包括 4 种类型: DS-8608SNL-SP、DS-8608SNL-ST、DS-8608SN-SP、DS-8608SN-ST, L 表示带 LCD, P 表 POE
DS_96XX_N_RX	2207	Netra NVR: DS-96xxN-RX
DS_96XXX_N_E	2213	高性能 NVR(256 路)
DS_76XXN_EX	2214	76/78N-EX(/N/P)系列 NVR, 包括 4/8/16 路的 E1 一盘位设备和 8/16/32 路的 E2 两盘位设备
DS_77XXN_E4	2215	77N-E4(/N/P)系列 NVR, 包括 8/16/32 路设备
DS_86XXN_E8	2216	86N-E8(/N/P)系列 NVR, 包括 8/16/32 路设备
PCNVR_IVMS_4200	2301	iVMS-4200 的存储服务器
IVMS_6200_TP	2401	IVMS-6200 交通诱导分析仪
IVMS_6200_TF	2402	IVMS-6200 交通取证分析仪

5.25 NET_DVR_DISKSTATE:硬盘状态

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_DISKSTATE
{
    public int    dwVolume;
    public int    dwFreeSpace;
    public int    dwHardDiskStatic;
}
```

Members

dwVolume

硬盘容量, 单位: MB

dwFreeSpace

硬盘剩余空间, 单位: MB

dwHardDiskStatic

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

5.26 NET_DVR_DIGITAL_CHANNEL_STATE:数字通道状态

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_DIGITAL_CHANNEL_STATE extends NET_DVR_CONFIG
{
    public byte[]    byDigitalAudioChanTalkState = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]    byDigitalChanState = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]    byDigitalAudioChanTalkStateEx = new byte[HCNetSDK.MAX_CHANNUM_V30*3];
    public byte[]    byDigitalChanStateEx = new byte[HCNetSDK.MAX_CHANNUM_V30*3];
    public byte[]    byAnalogChanState = new byte[HCNetSDK.MAX_ANALOG_CHANNUM];
    public byte[]    byRes = new byte[32];
}
```

Members

byDigitalAudioChanTalkState

表示数字语音通道的对讲状态，从第 1 个到第 MAX_CHANNUM_V30 个，数组元素值：0- 未使用，1- 使用中，0xff- 无效

byDigitalChanState

数字通道状态，从第 1 个到第 MAX_CHANNUM_V30 个，数组元素值表示状态码，0 表示无效，其他值定义见“Remarks”说明

byDigitalAudioChanTalkStateEx

表示数字语音通道的对讲状态，从第 MAX_CHANNUM_V30+1 个到第 MAX_CHANNUM_V30*4 个，数组元素值：0- 未使用，1- 使用中，0xff- 无效

byDigitalChanStateEx

数字通道状态，从第 MAX_CHANNUM_V30+1 个到第 MAX_CHANNUM_V30*4 个，数组元素值表示状态码，0 表示无效，其他值定义见“Remarks”说明

byRes

保留，置为

Remarks

数字通道状态（byDigitalChanState、byDigitalChanStateEx）取值定义如下所示：

宏定义	宏定义值	含义
NET_SDK_DC_STATUS_CONNECTED	1	已连接
NET_SDK_DC_STATUS_CONNECTING	2	正在连接
NET_SDK_DC_STATUS_BAND_WIDTH_EXCEED	3	超过系统带宽
NET_SDK_DC_STATUS_DOMAIN_ERROR	4	域名错误
NET_SDK_DC_STATUS_CHANNEL_ERROR	5	通道号错误
NET_SDK_DC_STATUS_ACCOUNT_ERROR	6	用户名或密码错误
NET_SDK_DC_STATUS_STREAM_TYPE_NOT_SUPPORT	7	流类型不支持
NET_SDK_DC_STATUS_CONFLICT_WITH_DVR	8	和设备 IP 地址冲突
NET_SDK_DC_STATUS_CONFLICT_WITH_IPC	9	和 IPC IP 地址冲突
NET_SDK_DC_STATUS_NETWORK_UNREACHABLE	10	网络不可达

NET_SDK_DC_STATUS_IPC_NOT_EXIST	11	IP 通道未接入
NET_SDK_DC_STATUS_IPC_EXCEPTION	12	IP 通道异常
NET_SDK_DC_STATUS_OTHER_ERROR	13	其他错误
NET_SDK_DC_STATUS_RESOLUTION_NOT_SUPPORT	14	IPC 分辨率不支持
NET_SDK_DC_STATUS_IPC_LAN_ERR	15	IPC 语言与 NVR 语言不匹配
NET_SDK_DC_STATUS_USER_LOCKED	16	用户被锁定
NET_SDK_DC_STATUS_NOT_ACTIVATED	17	设备未激活
NET_SDK_DC_STATUS_USER_NOT_EXIST	18	用户不存在
NET_SDK_DC_STATUS_IPC_UNREGISTERED	19	IP 通道对应设备未注册(GB28181 协议接入)

5.27 NET_DVR_ETHERNET_V30:以太网参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_ETHERNET_V30 {  
    public NET_DVR_IPADDR    struDVRIP = new NET_DVR_IPADDR();  
    public NET_DVR_IPADDR    struDVRIPMask = new NET_DVR_IPADDR();  
    public int                dwNetInterface;  
    public int                wDVRPort;  
    public int                wMTU;  
    public byte[]             byMACAddr = new byte[HCNetSDK.MACADDR_LEN];  
}
```

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

5.28 NET_DVR_FACE_DETECTION: 人脸侦测报警信息结构体

In class com.hikvision.netsdk.HCNetSDK

```
public static class NET_DVR_FACE_DETECTION extends Structure
{
    public int                dwSize;
    public int                dwRelativeTime;
    public int                dwAbsTime;
    public int                dwBackgroundPicLen;
    public NET\_VCA\_DEV\_INFO    struDevInfo;
    public NET\_VCA\_RECT        struFacePic[MAX_FACE_PIC_NUM];
    public byte               byFacePicNum;
    public byte               byRes1;
    public short              wDevInfolvmsChannelEx;
    public byte[]             byRes = new byte[252];
    public byte               *pBackgroundPicpBuffer;
}
```

Members

dwSize

结构体大小

dwRelativeTime

相对时标

dwAbsTime

绝对时标

dwBackgroundPicLen

背景图的长度，为 0 表示没有图片，大于 0 表示有图片

struDevInfo

前端设备信息，包括 IP、端口、通道号等

struFacePic

人脸子图区域，归一化值，相对于大图（背景图）的分辨率

byFacePicNum

子图数量

byRes1

保留

wDevInfolvmsChannelEx

与 NET_VCA_DEV_INFO 里的 bylvmsChannel 含义相同，能表示更大的值。老客户端用 bylvmsChannel 能继续兼容，但是最大到 255。新客户端版本请使用 wDevInfolvmsChannelEx

byRes

保留，置为 0

pBackgroundPicpBuffer

背景图的图片数据

5.29 NET_DVR_FACELIB_COND: 导入人脸数据(人脸图片+图片附件信息)到人脸库的条件

In class com.hikvision.netsdk.HCNetSDK

```
public static class NET_DVR_FACELIB_COND extends Structure
{
    public int dwSize;
    public byte[] szFDID = new byte[NET_SDK_MAX_FDID_LEN];
    public byte byConcurrent;
    public byte[] byRes = new byte[127];

    @Override
    protected List<String> getFieldOrder() {
        // TODO Auto-generated method stub
        return Arrays.asList("dwSize", "szFDID", "byConcurrent", "byRes");
    }
}
```

Members

dwSize

结构体大小

szFDID

人脸库 ID, 唯一标识

byConcurrent

设备并发处理: 0- 不开启, 1- 开始

byRes

保留, 置为 0

5.30 NET_DVR_FILECOND: 录像文件查找条件

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_FILECOND
{
    public int          lChannel;
    public int          dwFileType;
    public int          dwIsLocked;
    public int          dwUseCardNo;
    public byte[]       sCardNumber = new byte[32];
    public NET\_DVR\_TIME struStartTime = new NET_DVR_TIME();
    public NET\_DVR\_TIME struStopTime = new NET_DVR_TIME();
}
```

```
}
```

Members

IChannel

通道号，模拟通道从 1 开始，IP 通道一般从 33 开始

dwFileType

录像文件类型

不带卡号查找时类型:

0xff-全部, 0-定时录像, 1-移动侦测, 2-报警触发, 3-报警触发或移动侦测, 4-报警触发和移动侦测, 5-命令触发, 6-手动录像, 7-智能录像;

带卡号查找时类型:

0xff-全部, 0-定时录像, 1-移动侦测, 2-接近报警, 3-出钞报警, 4-进钞报警, 5-命令触发, 6-手动录像, 7-震动报警, 8-环境报警, 9-智能报警

dwIsLocked

是否锁定: 0-未锁定文件, 1-锁定文件, 0xff 表示所有文件 (包括锁定和未锁定)

dwUseCardNo

是否带卡号查找

sCardNumber

卡号

struStartTime

开始时间

struStopTime

停止时间

5.31 NET_DVR_FINDDATA_V30:录像文件查找结果

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_FINDDATA_V30
```

```
{
    public byte[]      sFileName = new byte[100];
    NET\_DVR\_TIME      struStartTime;
    NET\_DVR\_TIME      struStopTime;
    public int         dwFileSize;
    public byte[]      sCardNum = new byte[32];
    public byte        byLocked;
    public byte[]      byRes = new byte[3];
}
```

Members

sFileName

文件名

struStartTime

文件的开始时间

struStopTime

文件的结束时间

dwFileSize

文件大小

sCardNum

卡号

byLocked

文件是否被锁定：1- 文件已锁定；0- 文件未锁定

byRes

保留，置为 0

5.32 NET_DVR_FIND_PICTURE_PARAM: 图片查找条件结构体

In class `com.hikvision.netsdk.HCNetSDK`

```
public static class NET_DVR_FIND_PICTURE_PARAM extends Structure{
    public int          dwSize;
    public int          lChannel;
    public byte         byFileType;
    public byte         byNeedCard;
    public byte         byProvince;
    public byte         byRes1;
    public byte[]       sCardNum = new byte[CARDNUM_LEN_V30];
    public NET_DVR_TIME struStartTime = new NET_DVR_TIME();
    public NET_DVR_TIME struStopTime = new NET_DVR_TIME();
    public int          dwTrafficType;
    public int          dwVehicleType;
    public int          dwIllegalType;
    public byte         byLaneNo;
    public byte         bySubHvtType;
    public byte[]       byRes2 = new byte[2];
    public byte[]       sLicense = new byte[MAX_LICENSE_LEN];
    public byte         byRegion;
    public byte         byCountry;
    public byte[]       byRes3 = new byte[6];
    @Override
    protected List<String> getFieldOrder() {
        // TODO Auto-generated method stub
        return Arrays.asList("dwSize", "lChannel", "byFileType", "byNeedCard", "byProvince", "byRes1",
            "sCardNum", "struStartTime", "struStopTime", "dwTrafficType", "dwVehicleType", "dwIllegalType", "byLaneNo",
            "bySubHvtType", "byRes2", "sLicense", "byRegion", "byCountry", "byRes3");
    }
}
```

Members

dwSize

结构体大小

IChannel

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- 区域入侵侦测，0x10- 场景变更侦测，0x11- 设备本地回放时截图，0x12- 智能侦测，0x13- 进入区域侦测，0x14- 离开区域侦测，0x15- 徘徊侦测，0x16- 人员聚集侦测，0x17- 快速运动侦测，0x18- 停车侦测，0x19- 物品遗留侦测，0x1a- 物品拿取侦测，0x1b- 车牌侦测，0x1c- 混行检测，0x1d- 取证事件，0x1e- 火点检测，0x20- 船只检测，0x21- 测温预警，0x22- 测温报警，0x23- 温差报警，0x24- 违停检测，0x25- 人脸抓拍，0x26- 离线测温报警，0x2a- 起身检测，0x2b- 折线攀高，0x2c- 如厕超时，0xff- 全部类型

byNeedCard

是否需要卡号

byProvince

省份索引值（*byFileType* 为车牌侦测时有效）：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

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(仅智能交通设备支持)，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(仅智能交通设备支持)，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_OPERATE_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(仅智能交通设备支持)，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(仅智能交通设备支持), 取值范围: 1~99

bySubHvtType

混行检测查找子类型: 0- 全部, 1- 机动车(机动车子类型中支持车牌检索、省份检索), 2-非机动车, 3- 行人

byRes2

保留，置为 0

*sLicense*车牌号码（*byFileType* 为车牌侦测时有效）*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)，0xff- 所有

*byCountry*国家索引值，取值定义详见 **COUNTRY_INDEX***byRes3*

保留，置为 0

5.33 NET_DVR_HANDLEEXCEPTION_V30:异常参数

In class **com.hikvision.netsdk.HCNetSDK**

```
public class NET_DVR_HANDLEEXCEPTION_V30
{
    public int      dwHandleType;
    public byte[]   byRelAlarmOut = new byte[HCNetSDK.MAX_ALARMOUT_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

0x40: 联动电子地图(目前仅 PCNVR 支持)

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，依次类推

5.34 NET_DVR_HIDEALARM_V30:遮挡报警参数

In class **com.hikvision.netsdk.HCNetSDK**

```
public class NET_DVR_HIDEALARM_V30
```

```

{
    public int                                dwEnableHideAlarm;
    public short                             wHideAlarmAreaTopLeftX;
    public short                             wHideAlarmAreaTopLeftY;
    public short                             wHideAlarmAreaWidth;
    public short                             wHideAlarmAreaHeight;
    public NET\_DVR\_HANDLEEXCEPTION\_V30 struHideAlarmHandleType = new
NET_DVR_HANDLEEXCEPTION_V30();
    public NET\_DVR\_SCHEDTIME[][]            struAlarmTime = new
NET_DVR_SCHEDTIME[HCNetSDK.MAX_DAYS][HCNetSDK.MAX_TIMESEGMENT_V30];

    public NET_DVR_HIDEALARM_V30()
    {
        for(int i = 0; i < HCNetSDK.MAX_DAYS; i++)
        {
            for(int j = 0; j < HCNetSDK.MAX_TIMESEGMENT_V30; j++)
            {
                struAlarmTime[i][j] = new NET_DVR_SCHEDTIME();
            }
        }
    }
}

```

Members

dwEnableHideAlarm

是否启动遮挡报警：0-否，1-低灵敏度，2-中灵敏度，3-高灵敏度

wHideAlarmAreaTopLeftX

遮挡区域的 x 坐标

wHideAlarmAreaTopLeftY

遮挡区域的 y 坐标

wHideAlarmAreaWidth

遮挡区域的宽

wHideAlarmAreaHeight

遮挡区域的高

strHideAlarmHandleType

处理方法

struAlarmTime

布防时间

Remarks

SDK 设定整个图像的区域大小为 704*576，结构中遮挡区域的坐标以及宽和高需要转换成 704*576 大小区域下的坐标和宽高值。

5.35 **NET_DVR_IPADDR**:IP 地址

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_DVR_IPADDR extends Structure
{
    public byte[] slpV4 = new byte[16];
    public byte[] slpV6 = new byte[128];

    @Override
    protected List getFieldOrder() {
        //字段顺序必须跟C++结构体参数顺序一致
        return Arrays.asList("slpV4", "byRes");
    }
}
```

In class **com.hikvision.netsdk.HCNetSDK**

```
public class NET_DVR_IPADDR
{
    public byte[] slpV4 = new byte[16];
    public byte[] slpV6 = new byte[128];
}
```

Members

slpV4

设备 IPv4 地址

slpV6

设备 IPv6 地址

5.36 NET_DVR_IPALARMOUTCFG:IP 报警输出配置

In class **com.hikvision.netsdk.HCNetSDK**

```
public class NET_DVR_IPALARMOUTCFG
{
    public NET\_DVR\_IPALARMOUTINFO[] strulPAlarmOutInfo = new
    NET_DVR_IPALARMOUTINFO[HCNetSDK.MAX_IP_ALARMOUT];
}
```

Members

strulPAlarmOutInfo

IP 报警输出信息，每位数组表示一个 IP 报警输出

5.37 NET_DVR_IPALARMOUTINFO:IP 报警输出信息

In class **com.hikvision.netsdk.HCNetSDK**

```
public class NET_DVR_IPALARMOUTINFO
{
```

```

    public byte    byIPID;
    public byte    byAlarmOut;
    public byte[]  byRes = new byte[18];
}

```

Members

byIPID

IP 设备 ID，取值范围[1,MAX_IP_DEVICE]，其中#define MAX_IP_DEVICE 32

byAlarmOut

报警输出号

byRes

保留，置为 0

5.38 NET_DVR_IPCHANINFO:IP 通道参数

In class com.hikvision.netsdk.HCNetSDK

```

public class NET_DVR_IPCHANINFO
{
    public byte    byEnable;
    public byte    byIPID;
    public byte    byChannel;
}

```

Members

byEnable

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

byIPID

IP 设备 ID

byChannel

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

5.39 NET_DVR_IPDEVINFO_V31:IP 设备信息

In class com.hikvision.netsdk.HCNetSDK

```

public class NET_DVR_IPDEVINFO_V31
{
    public byte    byEnable;
    public byte    byProType;
    public byte[]  sUserName = new byte[HCNetSDK.NAME_LEN];
    public byte[]  sPassword = new byte[HCNetSDK.PASSWD_LEN];
    public byte[]  byDomain = new byte[HCNetSDK.MAX_DOMAIN_NAME];
}

```

```

    public NET\_DVR\_IPADDR    struIP = new NET_DVR_IPADDR();
    public int                wDVRPort;
}

```

Members

byEnable

该 IP 设备是否启用

byProType

协议类型(默认为私有协议), 0- 私有协议, 1- 松下协议, 2- 索尼

sUserName

用户名

sPassword

密码

byDomain

设备域名

struIP

IP 地址

wDVRPort

端口号

Remarks

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

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

5.40 [NET_DVR_IPPARACFG_V40](#):IP 接入参数

class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_IPPARACFG_V40 extends NET_DVR_CONFIG
```

```

{
    Public int                dwGroupNum;
    public int                dwAChanNum;
    public int                dwDChanNum;
    public int                dwStartDChan;
    public byte[]             byAnalogChanEnable = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public NET\_DVR\_IPDEVINFO\_V31[] struIPDevInfo = new
NET_DVR_IPDEVINFO_V31[HCNetSDK.MAX_IP_DEVICE_V40];
    public NET\_DVR\_IPCHANINFO[]    struIPChanInfo = new
NET_DVR_IPCHANINFO[HCNetSDK.MAX_CHANNUM_V30];

    public NET_DVR_IPPARACFG_V40()
    {
        for(int i=0; i<HCNetSDK.MAX_IP_DEVICE_V40; i++)
        {

```



```

        struIPDevInfo[i] = new NET_DVR_IPDEVINFO_V31();
    }
    for(int i=0; i<HCNetSDK.MAX_CHANNUM_V30; i++)
    {
        struIPChanInfo[i] = new NET_DVR_IPCHANINFO();
    }
}

```

Members

dwGroupNum

设备支持的总组数（只读）。若设备支持的组数大于 1，NET_DVR_GetDVRConfig（或者 NET_DVR_SetDVRConfig）获取（或设置）相关通道参数需要按照组数调用多次命令分别获取（或设置）各组通道参数，此时接口中 IChannel 对应组号

dwAChanNum

最大模拟通道个数（只读）

dwDChanNum

数字通道个数（只读）

dwStartDChan

起始数字通道（只读）

byAnalogChanEnable

模拟通道资源是否启用，数组下标与通道号一一对应，取值：0-禁用，1-启用。

例如：byAnalogChanEnable[i]=1 表示通道(i+1)启用

struIPDevInfo

IP 设备信息，下标 0 对应设备 IP ID 为 1

struIPChanInfo

IP 通道信息

5.41 NET_DVR_JPEGPARA:JPEG 图像参数

In class com.hikvision.netsdk.HCNetSDK

```

public class NET_DVR_JPEGPARA
{
    public short  wPicSize;
    public short  wPicQuality;
}

```

Members

wPicSize

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

wPicQuality

图片质量系数：0- 最好，1- 较好，2- 一般

5.42 NET_DVR_MOTION_V30:移动侦测参数

In class `com.hikvision.netsdk.HCNetSDK`

public class `NET_DVR_MOTION_V30`

```
{
    public byte[][] byMotionScope = new byte[64][96];
    public byte byMotionSensitive;
    public byte byEnableHandleMotion;
    public NET\_DVR\_HANDLEEXCEPTION\_V30 struMotionHandleType = new
NET_DVR_HANDLEEXCEPTION_V30();
    public NET\_DVR\_SCHEDTIME[][] struAlarmTime = new
NET_DVR_SCHEDTIME[HCNetSDK.MAX_DAYS][HCNetSDK.MAX_TIMESEGMENT_V30];
    public byte[] byRelRecordChan = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public NET_DVR_MOTION_V30(){
        for(int i = 0; i < HCNetSDK.MAX_DAYS; i++)
        {
            for(int j = 0; j < HCNetSDK.MAX_TIMESEGMENT_V30; j++)
            {
                struAlarmTime[i][j] = new NET_DVR_SCHEDTIME();
            }
        }
    }
}
```

Members

byMotionScope

移动侦测区域，96×64 的数组中，P 制只有 22×18 有效，N 制只有 22×15 有效，取值为 1 表示该宏块设定为移动侦测区域，0 表示不设定为移动侦测区域；其他保留，置为 0。

byMotionSensitive

移动侦测灵敏度，取值范围[0,5]，若取值为 0xff 表示关闭，值越大越灵敏

byEnableHandleMotion

是否处理移动侦测：0-不处理，1-处理

strMotionHandleType

处理方式

struAlarmTime

布防时间

byRelRecordChan

报警触发的录象通道，为 1 表示触发该通道

Remarks

整个图像的区域大小 P 制为 704×576，N 制为 704×480，即将 704×576（或者 704×480）的画面分割成 22×18（或者 22×15）个小宏块，然后可分别设置每个宏块是否为移动侦测区域。如果 P 制情况下，设备的

图像大小不为 704*576，比如高清设备 1280*720，设置移动侦测时需要将画面缩小成 704*576，然后设置移动侦测区域。

5.43 NET_DVR_MULTI_STREAM_COMPRESSIONCFG:多码流压缩参数配置

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_DVR_MULTI_STREAM_COMPRESSIONCFG extends Structure
{
    public int                dwSize;
    public int                dwStreamType;
    public NET\_DVR\_COMPRESSION\_INFO\_V30 struStreamPara = new NET_DVR_COMPRESSION_INFO_V30();
    public byte[]             byRes = new byte[80];

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("dwSize", "dwStreamType", "struStreamPara", "byRes");
    }
}
```

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

5.44 NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND:多码流压缩参数配置条件

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND extends Structure
{
    public int                dwSize;
    public NET\_DVR\_STREAM\_INFO struStreamInfo = new NET_DVR_STREAM_INFO();
    public int                dwStreamType;
    public byte[]             byRes[] = new byte[32];
}
```

```

@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("dwSize", "struStreamInfo", "dwStreamType", "byRes");
}
}

```

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

5.45 NET_DVR_NETCFG_V30:网络参数

In class com.hikvision.netsdk.HCNetSDK

```

public class NET_DVR_NETCFG_V30 extends NET_DVR_CONFIG{
    public NET\_DVR\_ETHERNET\_V30[] struetherNet = new NET_DVR_ETHERNET_V30[HCNetSDK.MAX_ETHERNET];
    public NET\_DVR\_IPADDR struAlarmHostIpAddr = new NET_DVR_IPADDR();
    public int wAlarmHostIpPort;
    public byte byUseDhcp;
    public NET\_DVR\_IPADDR struDnsServer1IpAddr = new NET_DVR_IPADDR();
    public NET\_DVR\_IPADDR struDnsServer2IpAddr = new NET_DVR_IPADDR();
    public byte [] byIpResolver = new byte[HCNetSDK.MAX_DOMAIN_NAME];
    public int wIpResolverPort;
    public int wHttpPortNo;
    public NET\_DVR\_IPADDR struMulticastIpAddr = new NET_DVR_IPADDR();
    public NET\_DVR\_IPADDR struGatewayIpAddr = new NET_DVR_IPADDR();
    public NET\_DVR\_PPPOECFG struPPPoE = new NET_DVR_PPPOECFG();
    public NET_DVR_NETCFG_V30(){
        for(int i=0; i<HCNetSDK.MAX_ETHERNET; i++){
            struetherNet[i] = new NET_DVR_ETHERNET_V30();
        }
    }
}

```

Members

struetherNet

以太网口

struAlarmHostIpAddr

报警主机 IP 地址

wAlarmHostIpPort

报警主机端口号

byUseDhcp

是否启用 DHCP: 0xff-无效; 0-不启用; 1-启用

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

5.46 NET_DVR_NTTPARA:NTP 参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_NTTPARA extends NET_DVR_CONFIG
{
    public byte []    sNTPServer = new byte[64];
    public short      wInterval;
    public byte       byEnableNTP;
    public char       cTimeDifferenceH;
    public char       cTimeDifferenceM;
    public short      wNtpPort;
}
```

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

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

wNtpPort

NTP 服务器端口，设备默认为 123

5.47 **NET_DVR_PLAYCOND**: 回放或者下载信息结构体

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_PLAYCOND {
    public int dwChannel;
    public NET_DVR_TIME struStartTime = new NET_DVR_TIME();
    public NET_DVR_TIME struStopTime = new NET_DVR_TIME();
    public byte byDrawFrame;
    public byte byStreamType;
    public byte[] byStreamID = new byte[HCNetSDK.STREAM_ID_LEN];
    public byte[] byRes = new byte[30];
}
```

Members

dwChannel

通道号

struStartTime

开始时间

struStopTime

结束时间

byDrawFrame

是否抽帧：0- 不抽帧，1- 抽帧

byStreamType

码流类型：0- 主码流，1- 子码流，2- 码流三

byStreamID

流 ID，使用流 ID 方式时 `dwChannel` 设为 0xffffffff

byRes

保留

5.48 **NET_DVR_PICCFG_V30**: 图像参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_PICCFG_V30 extends NET_DVR_CONFIG
{
    public byte[] sChanName = new byte[HCNetSDK.NAME_LEN];
    public int dwVideoFormat;
    public NET\_DVR\_VICOLOR struViColor = new NET_DVR_VICOLOR();
}
```

```

public int                dwShowChanName;
public short              wShowNameTopLeftX;
public short              wShowNameTopLeftY;
public NET\_DVR\_VILOST\_V30 struVILost = new NET_DVR_VILOST_V30();
public NET\_DVR\_MOTION\_V30 struMotion = new NET_DVR_MOTION_V30();
public NET\_DVR\_HIDEALARM\_V30 struHideAlarm = new NET_DVR_HIDEALARM_V30();
public int                dwEnableHide;
public NET\_DVR\_SHELTER\[\] struShelter = new NET_DVR_SHELTER[HCNetSDK.MAX_SHELTERNUM];
public int                dwShowOsd;
public short              wOSDTopLeftX;
public short              wOSDTopLeftY;
public byte               byOSDType;
public byte               byDispWeek;
public byte               byOSDAttrib;
public byte               byHourOsdType;
public byte               byFontSize;
public NET_DVR_PICCFG_V30(){
    for(int i = 0; i < HCNetSDK.MAX_SHELTERNUM; i++)
    {
        struShelter[i] = new NET_DVR_SHELTER();
    }
}
}

```

Members

sChanName

通道名称

dwVideoFormat

视频制式：0- 不支持，1- NTSC，2- PAL

struViColor

图像参数，保留

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

struMotion

移动侦测报警参数

struHideAlarm

遮挡报警参数

dwEnableHide

是否启动隐私遮蔽：0-否，1-是

struShelter

隐私遮蔽区域参数

dwShowOsd

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

wOSDTopLeftX

OSD 的 x 坐标

wOSDTopLeftY

OSD 的 y 坐标

byOSDType

OSD 类型(年月日格式):

0- XXXX-XX-XX 年月日

1- XX-XX-XXXX 月日年

2- XXXX 年 XX 月 XX 日

3- XX 月 XX 日 XXXX 年

4- XX-XX-XXXX 日月年

5- XX 日 XX 月 XXXX 年

byDispWeek

是否显示星期: 0-不显示, 1-显示

byOSDAtrib

OSD 属性 (透明/闪烁):

1- 透明, 闪烁

2- 透明, 不闪烁

3- 闪烁, 不透明

4- 不透明, 不闪烁

byHourOSDType

小时制: 0 表示 24 小时制, 1 表示 12 小时制或 am/pm

byFontSize

字体大小: 0- 16*16(中)/8*16(英), 1- 32*32(中)/16*32(英), 2- 64*64(中)/32*64(英) 3- 48*48(中)/24*48(英),
0xff-自适应(adaptive)

5.49 NET_DVR_FIND_PICTURE_V40: 图片查找结果信息结构体

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_DVR_FIND_PICTURE_V40 extends Structure{
    public byte[] sFileName = new byte[PICTURE_NAME_LEN];
    public NET_DVR_TIME struTime = new NET_DVR_TIME();
    public int dwFileSize;
    public byte[] sCardNum = new byte[CARDNUM_LEN_V30];
    public byte byPlateColor;
    public byte byVehicleLogo;
    public byte byFileType;
    public byte byRecogResult ;
}
```



```

        public byte[]                                sLicense = new byte[MAX_LICENSE_LEN];
        public byte                                byEventSearchStatus;
        public byte[]                                byRes = new byte[75];
        public NET_DVR_PIC_EXTRA_INFO_UNION uPicExtraInfo = new
NET_DVR_PIC_EXTRA_INFO_UNION();
        @Override
        protected List<String> getFieldOrder() {
            // TODO Auto-generated method stub
            return Arrays.asList("sFileName", "struTime", "dwFileSize", "sCardNum", "byPlateColor",
"byVehicleLogo", "byFileType", "byRecogResult", "sLicense", "byEventSearchStatus", "byRes", "uPicExtraInfo");
        }
    }

```

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(仅智能交通设备支持)，具体定义如下：

```

enum _VCA_PLATE_COLOR_{
    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(仅智能交通设备支持), 定义详见 VLR_VEHICLE_CLASS

byFileType

文件类型: 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, 0x13- 进入区域侦测, 0x14- 离开区域侦测, 0x15- 徘徊侦测, 0x16- 人员聚集侦测, 0x17- 快速运动侦测, 0x18- 停车侦测, 0x19- 物品遗留侦测, 0x1a- 物品拿取侦测, 0x1b- 车牌侦测, 0x1c- 混行检测, 0x1d- 取证事件, 0x1e- 火点检测, 0x20- 船只检测, 0x21- 测温预警, 0x22- 测温报警, 0x23- 温差报警, 0x24- 违停检测, 0x25- 人脸抓拍, 0x26- 离线测温报警, 0x2a- 起身检测, 0x2b- 折线攀高, 0x2c- 如厕超时, 0xff- 全部类型

byRecogResult

车辆识别结果: 0- 未知, 1- 客车, 2- 货车, 3- 轿车, 4- 面包车, 5- 小货车, 6- 行人, 7- 二轮车, 8- 三轮车, 9- SUV/MPV, 10- 中型客车, 11- 机动车, 12- 非机动车

sLicense

车牌号码

byEventSearchStatus

查找状态, 连续图片表示同一查找结果的时候, 0- 表示后面没有图片信息, 1- 表示后面还有图片信息。总的图片信息包括最后一张状态为 0 的图片。

byRes

保留

uPicExtraInfo

图片附件信息(文件类型为人脸侦测时有效)

5.50 NET_DVR_PLATE_INFO:车牌信息

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_DVR_PLATE_INFO extends Structure{
    public byte          byPlateType;
    public byte          byColor;
    public byte          byBright;
    public byte          byLicenseLen;
    public byte          byEntireBelieve;
    public byte          byRegion;
    public byte          byCountry;
    public byte[]        byRes = new byte[33];
    public NET\_VCA\_RECT struPlateRect = new NET_VCA_RECT();
    public byte[]        sLicense = new byte[MAX_LICENSE_LEN];
    public byte[]        byBelieve = new byte[MAX_LICENSE_LEN];

    @Override
```

```
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("byPlateType", "byColor", "byBright", "byLicenseLen", "byEntireBelieve",
        "byRegion", "byCountry", "byRes", "struPlateRect", "sLicense", "byBelieve");
}
}
```

Members

byPlateType

车牌类型：0-标准民用车与军车车牌，1-02 式民用车牌，2-武警车车牌，3-警车车牌，4-民用车双行尾牌，5-使馆车牌，6-农用车车牌，7-摩托车车牌

byColor

车牌颜色：0-蓝色车牌，1-黄色车牌，2-白色车牌，3-黑色车牌，4-绿色车牌，5-民航黑色车牌，0xff-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&俄罗斯(EU&CIS) ，0xff- 所有

byCountry

国家索引值，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%

5.51 NET_DVR_PLATE_RESULT:车辆识别结果

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_DVR_PLATE_RESULT extends Structure
{
    public int                dwSize;
    public byte               byResultType;
    public byte               byChanIndex;
    public short              wAlarmRecordID;
```

```

public int                dwRelativeTime;
public byte[]             byAbsTime = new byte[32];
public int                dwPicLen;
public int                dwPicPlateLen;
public int                dwVideoLen;
public byte               byTrafficLight;
public byte               byPicNum;
public byte               byDriveChan;
public byte               byVehicleType;
public int                dwBinPicLen;
public int                dwCarPicLen;
public int                dwFarCarPicLen;
public ByteByReference    pBuffer3;
public ByteByReference    pBuffer4;
public ByteByReference    pBuffer5;
public byte               byRelaLaneDirectionType;
public byte[]             byRes3 = new byte[7];
public NET\_DVR\_PLATE\_INFO struPlateInfo = new NET_DVR_PLATE_INFO();
public NET\_DVR\_VEHICLE\_INFO struVehicleInfo = new NET_DVR_VEHICLE_INFO();
public ByteByReference    pBuffer1;
public ByteByReference    pBuffer2;

```

```

public NET_DVR_PLATE_RESULT(Pointer p){
    super(p);
}
@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("dwSize", "byResultType", "byChanIndex", "wAlarmRecordID", "dwRelativeTime",
        "byAbsTime", "dwPicLen", "dwPicPlateLen", "dwVideoLen", "byTrafficLight", "byPicNum",
        "byDriveChan", "byVehicleType", "dwBinPicLen", "dwCarPicLen", "dwFarCarPicLen", "pBuffer3",
        "pBuffer4", "pBuffer5", "byRelaLaneDirectionType", "byRes3", "struPlateInfo", "struVehicleInfo",
        "pBuffer1", "pBuffer2");
}
}

```

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（与关联车道号对应，确保车道唯一性）：

0- 其它，1- 从东向西，2- 从西向东，3- 从南向北，4- 从北向南，5- 从东南向西北，6- 从西北向东南，7- 从东北向西南，8-从西南向东北

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。
- 当 `byResultType` 为 2 时, 可以用 `wAlarmRecordID` 作为查找条件, 搜索报警录像。

5.52 NET_DVR_POINT_FRAME:云台图像区域位置信息

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_POINT_FRAME
{
    public int    xTop;
    public int    yTop;
    public int    xBottom;
    public int    yBottom;
    public int    bCounter;
}
```

Members

xTop

方框起始点的 x 坐标

yTop

方框起始点的 y 坐标

xBottom

方框结束点的 x 坐标

yBottom

方框结束点的 y 坐标

bCounter

保留

Remarks

该结构体中的坐标值与当前预览显示框的大小有关, 现假设预览显示框为 352*288, 我们规定原点为预览显示框左上角的顶点, 前四个参数计算方法如下:

`xTop` = 鼠标当前所选区域的起始点坐标的值*255/352;

`xBottom` = 鼠标当前所选区域的结束点坐标的值*255/352;

`yTop` = 鼠标当前所选区域的起始点坐标的值*255/288;

`yBottom` = 鼠标当前所选区域的结束点坐标的值*255/288;

缩小条件: `xTop` 减去 `xBottom` 的值大于 2。放大条件: `xBottom` 大于 `xTop`。

5.53 NET_DVR_PPPOECFG:PPPoE 配置参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_PPPOECFG
```

```
{
    public int          dwPPPOE;
    public byte[]       sPPPoEUser = new byte[HCNetSDK.NAME_LEN];
    public byte[]       sPPPoEPassword = new byte[HCNetSDK.PASSWD_LEN];
    public NET\_DVR\_IPADDR struPPPoEIP = new NET_DVR_IPADDR();
}
```

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

5.54 [NET_DVR_PRESET_NAME](#):单个预置点名称

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_PRESET_NAME
{
    public int          wPresetNum;
    public byte[]       byName = new byte[HCNetSDK.NAME_LEN];
}
```

Members

wPresetNum

预置点编号

byName

预置点名称

5.55 [NET_DVR_PRESET_NAME_ARRAY](#):预置点名称参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_PRESET_NAME_ARRAY extends NET_DVR_CONFIG
{
    public NET\_DVR\_PRESET\_NAME[] struPresetName = new
    NET_DVR_PRESET_NAME[HCNetSDK.MAX_PRESET_NUM];

    public NET_DVR_PRESET_NAME_ARRAY()
    {
        for(int i = 0; i < HCNetSDK.MAX_PRESET_NUM; i++)
        {
```

```

        struPresetName[i] = new NET_DVR_PRESET_NAME();
    }
}
}

```

Members

struPresetName

预置点名称，每位数组表示一个预置点

5.56 NET_DVR_PREVIEWINFO:预览参数

In class com.hikvision.netsdk.HCNetSDK

```

public class NET_DVR_PREVIEWINFO
{
    public int    IChannel;
    public int    dwStreamType;
    public int    dwLinkMode
    public int    bBlocked
    public int    bPassbackRecord
    public byte   byPreviewMode
    public byte   byProtoType
    public int    nRTSPPort
}

```

Members

IChannel

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一般从 33 开始，具体取值在登录接口返回

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

bBlocked

0- 非阻塞取流，1- 阻塞取流

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

byProtoType

应用层取流协议：0- 私有协议，1- RTSP 协议

nRTSPPort

RTSP 端口

5.57 NET_DVR_PTZCFG:云台协议配置

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_PTZCFG
{
    public int                dwPtzNum;
    public NET\_DVR\_PTZ\_PROTOCOL[] struPtz = new
NET_DVR_PTZ_PROTOCOL[HCNetSDK.PTZ_PROTOCOL_NUM];

    public NET_DVR_PTZCFG(){
        for(int i=0; i<HCNetSDK.PTZ_PROTOCOL_NUM; i++){
            struPtz[i] = new NET_DVR_PTZ_PROTOCOL();
        }
    }
}
```

Members

dwPtzNum

有效的 PTZ 协议数目，从 0 开始（即总数为该值加 1）

struPtz

协议信息，最多 200 种

5.58 NET_DVR_PTZ_PROTOCOL:云台协议信息

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_PTZ_PROTOCOL
{
    public int        dwType;
    public byte[]     byDescribe = new byte[HCNetSDK.DESC_LEN];
}
```

Members

dwType

协议类型值

byDescribe

协议描述符

5.59 NET_DVR_QUERY_COUNTRYID_COND:按国家编号查询服务器信息

条件参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_QUERY_COUNTRYID_COND extends NET_DVR_ADDR_QUERY_COND
```

```
{
    public int      wCountryID;
    public byte[]   szSvrAddr      = new byte[HCNetSDK.MAX_DOMAIN_NAME];
    public byte[]   szClientVersion = new byte[HCNetSDK.CLIENT_VERSION_LEN];
}
```

Members

wCountryID

国家编号，取值详见 5.163 国家编号说明

szSvrAddr

服务器地址

szClientVersion

客户端版本信息，例如：iVMS4500 V4.0.0.0 build20150112

5.60 NET_DVR_QUERY_COUNTRYID_RET:按国家编号查询结果

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_QUERY_COUNTRYID_RET extends NET_DVR_ADDR_QUERY_RET
{
    public byte[]   szResolveSvrAddr = new byte[HCNetSDK.MAX_DOMAIN_NAME];
    public byte[]   szAlarmSvrAddr  = new byte[HCNetSDK.MAX_DOMAIN_NAME];
}
```

Members

szResolveSvrAddr

解析服务器地址

szAlarmSvrAddr

报警服务器地址

5.61 NET_DVR_QUERY_DDNS_COND:HIDDNS 查询或诊断条件

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_QUERY_DDNS_COND extends NET_DVR_ADDR_QUERY_COND
{
    public byte[]   szResolveSvrAddr = new byte[HCNetSDK.MAX_DOMAIN_NAME];
    public byte[]   szDevNickName = new byte[HCNetSDK.MAX_DOMAIN_NAME];
    public byte[]   szDevSerial = new byte[HCNetSDK.SERIALNO_LEN];
    public byte[]   szClientVersion = new byte[HCNetSDK.CLIENT_VERSION_LEN];
}
```

Members

szResolveSvrAddr

解析服务器地址

szDevNickName

设备域名，按设备域名查询时有效

szDevSerial

设备序列号，按序列号查询时有效

szClientVersion

客户端版本信息，例如：iVMS4500 V4.0.0.0 build20150112

5.62 NET_DVR_QUERY_DDNS_RET:HIDDNS 查询结果

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_QUERY_DDNS_RET extends NET_DVR_ADDR_QUERY_RET
{
    public byte[]    szDevIP = new byte[HCNetSDK.SDK_MAX_IP_LEN];
    public int       wCmdPort;
    public int       wHttpPort;
}
```

Members

szDevIP

设备 IP 地址

wCmdPort

设备 SDK 端口号

wHttpPort

设备 HTTP 端口

5.63 NET_DVR_CHECK_DDNS_RET:HIDDNS 诊断结果

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_CHECK_DDNS_RET extends NET_DVR_ADDR_QUERY_RET
{
    public byte                byDevStatus;
    public int                 wRegionID;
    public NET\_DVR\_QUERY\_DDNS\_RET struQueryRet = new NET_DVR_QUERY_DDNS_RET();
}
```

Members

byDevStatus

设备状态：0- 正常，1- 未查找到，2- 设备不在线，3- 设备不在该区域

wRegionID

国家区域 ID：1- USA，2- South America，3- Asia，4-China，5-Europe，6-其他

struQueryRet

设备 IP 和端口查询结果

5.64 NET_DVR_QUERY_IPSERVER_COND:IPServer 查询条件

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_QUERY_IPSERVER_COND extends NET_DVR_ADDR_QUERY_COND
{
    public byte[]    szResolveSvrAddr = new byte[HCNetSDK.MAX_DOMAIN_NAME];
    public int       wResolveSvrPort;
    public byte[]    szDevNickName = new byte[HCNetSDK.MAX_DOMAIN_NAME];
    public byte[]    szDevSerial = new byte[HCNetSDK.SERIALNO_LEN];
}
```

Members

szResolveSvrAddr

IPServer 服务器地址

wResolveSvrPort

IPServer 服务器端口，7071

szDevNickName

设备名称，按设备名称查询时有效

szDevSerial

设备序列号，按序列号查询时有效

5.65 NET_DVR_QUERY_IPSERVER_RET:IPServer 查询结果

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_QUERY_IPSERVER_RET extends NET_DVR_ADDR_QUERY_RET
{
    public byte[]    szDevIP = new byte[HCNetSDK.SDK_MAX_IP_LEN];
    public int       wCmdPort;
}
```

Members

szDevIP

设备 IP 地址

wCmdPort

设备 SDK 端口号

5.66 NET_DVR_RECORDDAY:全体录像参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_RECORDDAY
{
    public short    wAllDayRecord;
    public byte     byRecordType;
```

```
}
```

Members

wAllDayRecord

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

5.67 NET_DVR_RECORDSCHED:时间段录像参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_RECORDSCHED {
    public NET\_DVR\_SCHEDTIME struRecordTime = new NET_DVR_SCHEDTIME();
    public byte byRecordType;
}
```

Members

struRecordTime

录像时间

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

5.68 NET_DVR_RECORD_V30:录像参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_RECORD_V30 extends NET_DVR_CONFIG
{
    public int dwRecord;
    public NET\_DVR\_RECORDDAY[] struRecAllDay = new NET_DVR_RECORDDAY[HCNetSDK.MAX_DAYS];
    public NET\_DVR\_RECORDSCHED[][] struRecordSched = new
NET_DVR_RECORDSCHED[HCNetSDK.MAX_DAYS][HCNetSDK.MAX_TIMESEGMENT_V30];
    public int dwRecordTime;
    public int dwPreRecordTime;
    public int dwRecorderDuration;
    public byte byRedundancyRec;
    public byte byAudioRec;
```

```

public byte                byStreamType;
public byte                byPassbackRecord;
public short               wLockDuration;
public byte                byRecordBackup;
public byte                bySVCLLevel;
public byte[]              byReserve = new byte[4];
public NET_DVR_RECORD_V30(){
    for(int i=0; i<HCNetSDK.MAX_DAYS; i++){
        struRecAllDay[i] = new NET_DVR_RECORDDAY();
        for(int j=0; j<HCNetSDK.MAX_TIMESEGMENT_V30; j++)
        {
            struRecordSched[i][j] = new NET_DVR_RECORDSCHED();
        }
    }
}
}

```

Members

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，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

byReserve

保留，置为 0

5.69 NET_DVR_RESOLVE_DEVICEINFO:域名解析设备信息

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_RESOLVE_DEVICEINFO
{
    public byte[]  sGetIP = new byte[64];
    public int     dwPort;
}
```

Members

sGetIP

设备 IP 地址

dwPort

设备端口号

Remarks

通过域名解析出设备当前 IP 地址和端口号，然后调用 [NET_DVR_Login_V30](#) 登录设备。

5.70 NET_DVR_SCHEDTIME:时间段参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_SCHEDTIME
{
    Public byte  byStartHour;
    Public byte  byStartMin;
    Public byte  byStopHour;
    Public byte  byStopMin;
}
```

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

5.71 NET_DVR_SDKLOCAL_CFG:SDK 本地参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_SDKLOCAL_CFG
{
    public byte    byEnableAbilityParse;
    public byte[]  byProtectKey = new byte[128];
    public byte    byCompatibleType;
}
```

Members

byEnableAbilityParse

使用能力集解析库：0-不使用，1-使用，默认不使用

byProtectKey

保留，置为 0

byCompatibleType

保留，置为 0

5.72 NET_DVR_SEARCH_EVENT_PARAM:按事件搜索的条件参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_SEARCH_EVENT_PARAM
{
    public short      wMajorType;
    public short      wMinorType;
    public NET\_DVR\_TIME struStartTime = new NET_DVR_TIME();
    public NET\_DVR\_TIME struEndTime = new NET_DVR_TIME();
    public byte        byLockType;
    public int[]       wAlarmInNo = new int[128];
    public int[]       wMotDetChanNo = new int[64];
    public int[]       wBehaviorChanNo = new int[64];
    public int[]       dwVCAChanNo = new int[HCNetSDK.MAX_CHANNUM_V30-1];
}
```

Members

wMajorType

搜索主类型，具体定义见下表：

```
enum _MAIN_EVENT_TYPE_ {
    EVENT_MOT_DET      = 0,
    EVENT_ALARM_IN     = 1,
    EVENT_VCA_BEHAVIOR = 2,
    EVENT_INQUEST      = 3,
    EVENT_VCA_DETECTION = 4
}MAIN_EVENT_TYPE
```


EVENT_MOT_DET

移动侦测

EVENT_ALARM_IN

报警输入

EVENT_VCA_BEHAVIOR

异常行为识别

EVENT_INQUEST

审讯事件

EVENT_VCA_DETECTION

智能侦测

wMinorType

搜索次类型，目前只支持 0xffff，表示全部

struStartTime

搜索的开始时间

struEndTime

搜索的停止时间

byLockType

是否锁定：0xff- 全部，0- 未锁，1- 锁定

*wAlarmInNo*报警输入号，按值表示，采用紧凑型排列，例如 `wAlarmInNo[0]==1&&wAlarmInNo[1]==2` 表示查找由报警输入 1 和报警输入 2 触发的事件*wMotDetChanNo*设备通道号，按值表示，采用紧凑型排列，例如 `wMotDetChanNo[0]==1&&wMotDetChanNo[1]==2` 表示查找通道 1 和通道 2 发生移动侦测触发的事件*wBehaviorChanNo*异常行为识别对应的通道，按值表示，采用紧凑型排列，`dwChanNo[0]==1&&dwChanNo[1]==2` 表示查找通道 1 和通道 2 的智能侦测事件*dwVCAChanNo*智能侦测对应的通道，按值表示，采用紧凑型排列，`dwChanNo[0]==1&&dwChanNo[1]==2` 表示查找通道 1 和通道 2 的智能侦测事件

5.73 NET_DVR_SEARCH_EVENT_RET:按事件搜索结果信息

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_SEARCH_EVENT_RET
{
    public short          wMajorType;
    public short          wMinorType;
    public NET\_DVR\_TIME   struStartTime = new NET_DVR_TIME();
    public NET\_DVR\_TIME   struEndTime = new NET_DVR_TIME();
    public int            dwAlarmInNo;
    public int            dwMotDetNo;
    public int            dwBehaviorChanNo;
```

}

Members**wMajorType**

搜索主类型，具体定义见下表：

```
enum _MAIN_EVENT_TYPE_{
    EVENT_MOT_DET      = 0,
    EVENT_ALARM_IN     = 1,
    EVENT_VCA_BEHAVIOR = 2,
    EVENT_INQUEST      = 3,
    EVENT_VCA_DETECTION = 4
}MAIN_EVENT_TYPE
```

EVENT_MOT_DET

移动侦测

EVENT_ALARM_IN

报警输入

EVENT_VCA_BEHAVIOR

异常行为识别

EVENT_INQUEST

审讯事件

EVENT_VCA_DETECTION

智能侦测

wMinorType

搜索次类型，不同的主类型对应次类型变化，移动侦测和报警输入没有对应次类型，其他主类型对应的次类型见下表：

主类型的宏定义	主类型的宏定义值	含义
EVENT_VCA_BEHAVIOR	2	异常行为识别
次类型宏定义	宏定义值	含义
EVENT_TRAVERSE_PLANE	0	穿越警戒面
EVENT_ENTER_AREA	1	目标进入区域，支持区域规则
EVENT_EXIT_AREA	2	目标离开区域，支持区域规则
EVENT_INTRUSION	3	周界入侵，支持区域规则
EVENT_LOITER	4	徘徊，支持区域规则
EVENT_LEFT_TAKE	5	丢包捡包，支持区域规则
EVENT_PARKING	6	停车，支持区域规则
EVENT_RUN	7	奔跑，支持区域规则
EVENT_HIGH_DENSITY	8	区域内人员密度，支持区域规则
EVENT_STICK_UP	9	贴纸条，支持区域规则
EVENT_INSTALL_SCANNER	10	安装读卡器，支持区域规则
EVENT_OPERATE_OVER_TIME	11	操作超时
EVENT_FACE_DETECT	12	异常人脸

EVENT_LEFT	13	物品遗留
EVENT_TAKE	14	物品拿取
EVENT_LEAVE_POSITION	15	离岗事件
EVENT_TRAIL_INFO	16	尾随
EVENT_FALL_DOWN_INFO	19	倒地
EVENT_OBJECT_PASTE	20	异物粘贴区域
EVENT_FACE_CAPTURE_INFO	21	正常人脸
EVENT_MULTI_FACES_INFO	22	多张人脸
EVENT_AUDIO_ABNORMAL_INFO	23	声强突变

主类型的宏定义	主类型的宏定义值	含义
EVENT_INQUEST	3	审讯事件
次类型宏定义	宏定义值	含义
INQUEST_START_INFO	0x1001	审讯开始信息
INQUEST_STOP_INFO	0x1002	审讯停止信息
INQUEST_TAG_INFO	0x1003	重点标记信息
INQUEST_SEGMENT_INFO	0x1004	审讯片断状态信息

主类型的宏定义	主类型的宏定义值	含义
EVENT_VCA_DETECTION	4	智能侦测
次类型宏定义	宏定义值	含义
EVENT_VCA_TRAVERSE_PLANE	1	越界侦测
EVENT_FIELD_DETECTION	2	区域入侵侦测
EVENT_AUDIO_INPUT_ALARM	3	音频丢失侦测
EVENT_SOUND_INTENSITY_ALARM	4	声强突变侦测（声强陡升侦测、声强陡降侦测）
EVENT_FACE_DETECTION	5	人脸侦测
EVENT_VIRTUAL_FOCUS_ALARM	6	虚焦侦测
EVENT_SCENE_CHANGE_ALARM	7	场景变更侦测
EVENT_PIR_ALARM	8	PIR 报警

struStartTime

事件录像开始时间

struEndTime

事件录像停止时间

dwAlarmInNo

报警输入号

dwMotDetNo

移动侦测事件触发的设备通道号

dwBehaviorChanNo

异常行为识别事件触发的通道号

5.74 NET_DVR_SEND_PARAM_IN:数据发送输入参数结构体

```
public static class NET_DVR_SEND_PARAM_IN extends Structure
```

```
{
    public Pointer pSendData;
    public int dwSendDataLen;
    public NET_DVR_TIME_V30 struTime;
    public byte byPicType;
    public byte[] byRes1 = new byte[3];
    public int dwPicManageNo;
    public byte[] sPicName = new byte[NAME_LEN];
    public int dwPicDisplayTime;
    public Pointer pSendAppendData;
    public int dwSendAppendDataLen;
    public byte[] byRes = new byte[192];

    @Override
    protected List<String> getFieldOrder() {
        // TODO Auto-generated method stub
        return Arrays.asList("pSendData", "dwSendDataLen", "struTime", "byPicType", "byRes1",
            "dwPicManageNo",
            "sPicName", "dwPicDisplayTime", "pSendAppendData", "dwSendAppendDataLen",
            "byRes");
    }
}
```

Members

pSendData

发送的缓冲区

dwSendDataLen

发送数据长度

struTime

图片时间

byPicType

图片格式：1- jpg， 2- bmp， 3- png， 4- SWF， 5- GIF

byRes1

保留，置为 0

dwPicMangeNo

图片管理号

sPicName

图片名称

dwPicDisplayTime

图片播放时长，单位：秒

pSendAppendData

发送图片的附加信息缓冲区，不支持，设为 NULL

dwSendAppendDataLen

发送图片的附加信息数据长度，不支持，设为 0

byRes

保留，置为 0

5.75 NET_DVR_SERIALSTART_V40: 串口参数子类

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_SERIALSTART_V40 extends NET_DVR_SERIAL_COND
{
    public int    dwSerialPort;
    public int    wPort;
}
```

Members

dwSerialPort

串口类型：1- 232 串口，2- 485 串口

wPort

串口编号

5.76 NET_DVR_SERIAL_COND: 串口参数基类

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_SERIAL_COND{
}
```

Remarks

串口参数子类：[NET_DVR_SERIALSTART_V40](#)。

5.77 NET_DVR_SETUPALARM_PARAM: 报警布防参数

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_DVR_SETUPALARM_PARAM extends Structure
{
    public int    dwSize;
    public byte   byLevel;
    public byte   byAlarmInfoType;
    public byte   byRetAlarmTypeV40;
```

```

public byte    byRetDevInfoVersion;
public byte    byRetVQDAlarmType;
public byte    byFaceAlarmDetection;
public byte    bySupport;
public byte    byBrokenNetHttp;
public short   wTaskNo;
public byte[]  byRes1 = new byte[6];

protected List<String> getFieldOrder()
{
    return Arrays.asList("dwSize", "byLevel", "byAlarmInfoType", "byRetAlarmTypeV40",
        "byRetDevInfoVersion", "byRetVQDAlarmType", "byFaceAlarmDetection", "bySupport",
        "byBrokenNetHttp", "wTaskNo", "byRes1");
}
}

```

Remarks

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（COMM_UPLOAD_PLATE_RESULT），1- 新报警信息（COMM_ITS_PLATE_RESULT）

byRetAlarmTypeV40

0- 移动侦测、视频丢失、遮挡、IO 信号量等报警信息以普通方式上传（COMM_ALARM_V30），1- 报警信息以数据可变长方式上传（COMM_ALARM_V40，设备若不支持则仍以普通方式上传）

byRetDevInfoVersion

CVR 上传报警信息回调结构体版本：0- COMM_ALARM_DEVICE，1- COMM_ALARM_DEVICE_V40

byRetVQDAlarmType

VQD 报警上传类型类型：0- 上传 VQD 诊断信息（COMM_ALARM_VQD），1- VQD 诊断异常信息（COMM_ALARM_VQD_EX）

byFaceAlarmDetection

人脸侦测报警信息类型：1- 表示人脸侦测报警扩展([NET DVR FACE DETECTION](#))，0- 表示原先支持结构([NET VCA FACESNAP RESULT](#))

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

byBrokenNetHttp

断网续传类型（设备目前只支持一个断网续传布防连接），按位表示，值：0- 不续传，1- 续传

bit0- 车牌检测（IPC）

bit1- 客流统计

bit2- 热度图统计

例如：byBrokenNetHttp&0x1==0 表示车牌检测结果不续传

wTaskNo

任务处理号

byRes1

保留，置为 0

Remarks

- *byLevel* 和 *byAlarmInfoType* 针对智能交通设备（抓拍机）：一级布防最大连接数为 1 个，二级最大连接数为 3 个，三级最大连接数为 5 个，设备支持一级、二级、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。
- *byRetVQDAlarmType* 针对具有 VQD 诊断功能的设备。
- *wTaskNo* 针对车辆二次检测设备，用于区分不同布防链接，现在上传机制为：
 - 1) 如果 *wTaskNo* 为 0，所有的处理结果都需要从这个链接上传。
 - 2) 如果两个布防链接中 *wTaskNo* 的值相同（0 除外），返回布防链接错误。
 - 3) 布防链接后，Client 端下发任务单号 *wTaskNo*，和车辆二次识别任务和上传的结果中的 *wTaskNo* 都对应的。例如：布防链接中 *wTaskNo*==1，任务 A 中 *wTaskNo*==1，结果信息回调 *wTaskNo*==1（该信息回调只在布防中 *wTaskNo* == 1 的链接中回调）。

5.78 NET_DVR_SHELTER:隐私遮盖参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_SHELTER
{
    public short    wHideAreaTopLeftX;
    public short    wHideAreaTopLeftY;
    public short    wHideAreaWidth;
    public short    wHideAreaHeight;
}
```

Members

wHideAreaTopLeftX

遮挡区域的 x 坐标

wHideAreaTopLeftY

遮挡区域的 y 坐标

wHideAreaWidth

遮挡区域的宽

wHideAreaHeight

遮挡区域的高

Remarks

SDK 设定整个图像的区域大小为 704*576，结构中遮挡区域的坐标以及宽和高需要转换成 704*576 大小区域下的坐标和宽高值。

5.79 NET_DVR_SHOWSTRINGINFO:单个字符参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_SHOWSTRINGINFO
{
    public short      wShowString;
    public short      wStringSize;
    public short      wShowStringTopLeftX;
    public short      wShowStringTopLeftY;
    public byte[]     sString = new byte[44];
    @Override
    protected List<String> getFieldOrder() {
        // TODO Auto-generated method stub
        return Arrays.asList("wShowString", "wStringSize", "wShowStringTopLeftX", "wShowStringTopLeftY",
"sString");
    }
}
```

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

5.80 NET_DVR_SHOWSTRING_V30:字符叠加参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_SHOWSTRING_V30 extends NET_DVR_CONFIG
{
    public      int                dwSize;
    public      NET\_DVR\_SHOWSTRINGINFO[] struStringInfo = (NET_DVR_SHOWSTRINGINFO[])new
NET_DVR_SHOWSTRINGINFO().toArray(MAX_STRINGNUM_V30);

    @Override
    protected List<String> getFieldOrder() {
        // TODO Auto-generated method stub
        return Arrays.asList("dwSize", "struStringInfo");
    }
}
```



```
    }
}
```

Members

dwSize

结构体大小

struStringInfo

要显示的字符内容

5.81 NET_DVR_STREAM_INFO: 流 ID 信息

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_DVR_STREAM_INFO extends Structure
{
    public int      dwSize;
    public byte[]   byID = new byte[STREAM_ID_LEN];
    public int      dwChannel;
    public byte[]   byRes = new byte[32];

    @Override
    protected List<String> getFieldOrder() {
        return Arrays.asList("dwSize", "byID", "dwChannel", "byRes");
    }
}
```

Members

dwSize

结构体大小

byID

流 ID，为字母、数字和"_"的组合。全部为 0 时，无效

dwChannel

关联的设备通道

byRes

保留，置为 0

Remarks

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

5.82 NET_DVR_SINGLE_DDNS: DDNS 服务器信息

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_SINGLE_DDNS
{
    public byte[]   sUserName = new byte[HCNetSDK.NAME_LEN];
    public byte[]   sPassword = new byte[HCNetSDK.PASSWD_LEN];
}
```

```

    public byte[]    sDomainName = new byte[HCNetSDK.MAX_DOMAIN_NAME];
    public byte[]    sServerName = new byte[HCNetSDK.MAX_DOMAIN_NAME];
    public int       wDDNSPort;
    public byte[]    byRes = new byte[16];
}

```

Members

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

5.83 NET_DVR_TIME:时间参数

In class `com.hikvision.netsdk.HCNetSDK`

```

public class NET_DVR_TIME
{
    public int    dwYear;
    public int    dwMonth;
    public int    dwDay;
    public int    dwHour;
    public int    dwMinute;
    public int    dwSecond;
}

```

Members

dwYear

年

dwMonth

月

dwDay

日

dwHour

时

dwMinute

分

dwSecond

秒

5.84 NET_DVR_TIME_EX:时间参数

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_DVR_TIME_EX extends Structure
{
    public short wYear;
    public byte byMonth;
    public byte byDay;
    public byte byHour;
    public byte byMinute;
    public byte bySecond;
    public byte byRes;

    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("wYear","byMonth", "byDay", "byHour","byMinute", "bySecond", "byRes");
    }
}
```

Members

wYear
年
byMonth
月
byDay
日
byHour
时
byMinute
分
bySecond
秒
byRes
保留

5.85 NET_DVR_TIME_V30:时间参数

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_DVR_TIME_V30 extends Structure
{
    public short wYear;
```

```

public byte    byMonth;
public byte    byDay;
public byte    byHour;
public byte    byMinute;
public byte    bySecond;
public byte    byRes;
public short    wMilliSec;
public byte[]  byRes1 = new byte[2];

@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("wYear","byMonth","byDay", "byHour","byMinute", "bySecond","byRes",
        "wMilliSec", "byRes1");
}
}

```

Members

wYear
年

byMonth
月

byDay
日

byHour
时

byMinute
分

bySecond
秒

byRes
保留

wMilliSec
毫秒

byRes1
保留

5.86 NET_DVR_UPLOAD_FILE_RET:文件上传结果信息结构体

In class `com.hikvision.netsdk.HCNetSDK`

```

public static class NET_DVR_UPLOAD_FILE_RET extends Structure
{
    public byte[] sUrl = new byte[MAX_UPLOADFILE_URL_LEN];
    public byte[] byRes = new byte[260];
}

```

```

    @Override
    protected List<String> getFieldOrder() {
        // TODO Auto-generated method stub
        return Arrays.asList("sUrl", "byRes");
    }
}

```

Members

sUrl

文件上传返回的 URL 或者 ID

byRes

保留

5.87 NET_DVR_UPNP_NAT_STATE:Upnp 端口映射状态

In class com.hikvision.netsdk.HCNetSDK

```

public class NET_DVR_UPNP_NAT_STATE
{
    public NET\_DVR\_UPNP\_PORT\_STATE[] strUpnpPort = new NET_DVR_UPNP_PORT_STATE[12];
}

```

Members

strUpnpPort

端口映射状态：数组 0- web server 端口，数组 1- 管理端口，数组 2- rtsp 端口

5.88 NET_DVR_UPNP_PORT_STATE:Upnp 端口映射状态

In class com.hikvision.netsdk.HCNetSDK

```

public class NET_DVR_UPNP_PORT_STATE
{
    public int dwEnabled;
    public int wInternalPort;
    public int wExternalPort;
    public int dwStatus;
    public NET\_DVR\_IPADDR struNatExternalIp;
    public NET\_DVR\_IPADDR struNatInternalIp;
}

```

Members

dwEnabled

该端口是否被使能映射

wInternalPort

映射前的端口

wExternalPort

映射后的端口

dwStatus

端口映射状态：0- 未生效；1- 未生效：映射源端口与目的端口需一致；2- 未生效：映射端口号已被使用；3- 生效

struNatExternalIp

映射后的外部地址

struNatInternalIp

NAT 路由器 LAN IP 地址

5.89 NET_DVR_USER_INFO_V30:单个用户信息参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_USER_INFO_V30
```

```
{
    public byte[]          sUserName = new byte[HCNetSDK.NAME_LEN];
    public byte[]          sPassword = new byte[HCNetSDK.PASSWD_LEN];
    public byte[]          byLocalRight = new byte[HCNetSDK.MAX_RIGHT];
    public byte[]          byRemoteRight = new byte[HCNetSDK.MAX_RIGHT];
    public byte[]          byNetPreviewRight = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]          byLocalPlaybackRight = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]          byNetPlaybackRight = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]          byLocalRecordRight = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]          byNetRecordRight = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]          byLocalPTZRight = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]          byNetPTZRight = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]          byLocalBackupRight = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public NET_DVR_IPADDR  struUserIP = new NET_DVR_IPADDR();
    public byte[]          byMACAddr = new byte[HCNetSDK.MACADDR_LEN];
    public byte            byPriority;
    public byte[]          byRes = new byte[17];
}
```

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），重启，关机）

数组 6- 本地查看参数

数组 7- 本地管理模拟和 IP camera

数组 8- 本地备份

数组 9- 本地关机/重启

byRemoteRight

远程操作权限，参数取值为 1 表示使能：

数组 0- 远程控制云台

数组 1- 远程手动录象

数组 2- 远程回放

数组 3- 远程设置参数

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，重启，关机）

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)，设置之后只有该 IP 地址对应的客户端可以访问设备

byMACAddr

物理地址，设置之后只有 MAC 物理地址对应的客户端可以访问设备

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）

byRes

保留，置为 0

5.90 NET_DVR_USER_V30:用户参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_USER_V30 extends NET_DVR_CONFIG
{
    public NET\_DVR\_USER\_INFO\_V30\[\] struUser = new
NET_DVR_USER_INFO_V30[HCNetSDK.MAX_USERNUM_V30];
    public NET_DVR_USER_V30()
    {
        for(int i = 0; i < HCNetSDK.MAX_USERNUM_V30; i++)
        {
            struUser[i] = new NET_DVR_USER_INFO_V30();
        }
    }
}
```

Members

struUser

用户信息参数，每位数组表示一个用户，最大支持 32 个用户

5.91 NET_DVR_VEHICLE_INFO:车辆信息参数

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_DVR_VEHICLE_INFO extends Structure
{
    public int      dwIndex;
    public byte     byVehicleType;
    public byte     byColorDepth;
    public byte     byColor;
    public byte     byRes1;
    public short    wSpeed;
    public short    wLength;
```



```

public byte    byIllegalType;
public byte    byVehicleLogoRecog;
public byte    byVehicleSubLogoRecog;
public byte    byVehicleModel;
public byte[]  byCustomInfo = new byte[16];
public short   wVehicleLogoRecog;
public byte[]  byRes3 = new byte[14];

```

```
@Override
```

```
protected List<String> getFieldOrder() {
```

```
    //字段顺序必须跟 C++结构体参数顺序一致
```

```
    return Arrays.asList("dwIndex", "byVehicleType", "byColorDepth", "byColor", "byRes1", "wSpeed",
        "wLength", "byIllegalType", "byVehicleLogoRecog", "byVehicleSubLogoRecog", "byVehicleModel",
        "byCustomInfo", "wVehicleLogoRecog", "byRes3");
```

```
}
```

```
}
```

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，13-深灰，14-青色，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

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，22-行人闯红灯，23-加塞。对于 ITS 终端服务器，该参数无效，违法类型通过 [NET ITS PLATE RESULT](#) 中的 *wIllegalType* 进行判断。

byVehicleLogoRecog

车辆主品牌，请使用 *wVehicleLogoRecog* 字段

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*)

byRes3

保留

5.92 NET_DVR_VICOLOR:时间段图像参数

In class **com.hikvision.netsdk.HCNetSDK**

```
public class NET_DVR_VICOLOR
{
    public NET_DVR_COLOR[]    struColor = new NET_DVR_COLOR[HCNetSDK.MAX_TIMESEGMENT_V30];
    public NET_DVR_SCHETIME[] struHandleTime = new
NET_DVR_SCHETIME[HCNetSDK.MAX_TIMESEGMENT_V30];
    public NET_DVR_VICOLOR()
    {
        for(int i = 0; i < HCNetSDK.MAX_TIMESEGMENT_V30; i++)
        {
            struColor[i] = new NET_DVR_COLOR();
            struHandleTime[i] = new NET_DVR_SCHETIME();
        }
    }
}
```

Members

struColor

图象参数，保留

struHandleTime

处理时间段，目前不支持，保留

5.93 NET_DVR_VIDEOEFFECT:视频显示参数

In class **com.hikvision.netsdk.HCNetSDK**

```
public class NET_DVR_VIDEOEFFECT
{
    public int    iBrightValue;
    public int    iContrastValue;
    public int    iSaturationValue;
    public int    iHueValue;
}
```

Members

iBrightValue

亮度，取值范围：[1,10]

iContrastValue

对比度，取值范围：[1,10]

iSaturationValue

饱和度，取值范围：[1,10]

iHueValue

色度，取值范围：[1,10]

5.94 NET_DVR_VILOST_V30: 视频丢失参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_VILOST_V30
```

```
{
    public byte                                byEnableHandleVILost;
    public NET\_DVR\_HANDLEEXCEPTION\_V30 struVILostHandleType = new NET_DVR_HANDLEEXCEPTION_V30();
    public NET\_DVR\_SCHEDULETIME[][] struAlarmTime = new
NET_DVR_SCHEDULETIME[HCNetSDK.MAX_DAYS][HCNetSDK.MAX_TIMESEGMENT_V30];
    public NET_DVR_VILOST_V30(){
        for(int i = 0; i < HCNetSDK.MAX_DAYS; i++)
        {
            for(int j = 0; j < HCNetSDK.MAX_TIMESEGMENT_V30; j++)
            {
                struAlarmTime[i][j] = new NET_DVR_SCHEDULETIME();
            }
        }
    }
}
```

Members

byEnableHandleVILost

是否处理信号丢失报警：0-不处理，1-处理

strVILostHandleType

处理方式参数

struAlarmTime

布防时间参数

5.95 NET_DVR_VOD PARA: 录像回放参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_VOD_PARA {
    public int dwSize;
```

```

public NET\_DVR\_STREAM\_INFO struIDInfo = new NET_DVR_STREAM_INFO();
public NET\_DVR\_TIME struBeginTime = new NET_DVR_TIME();
public NET\_DVR\_TIME struEndTime = new NET_DVR_TIME();
public Surface hWnd;
public byte byDrawFrame;
public byte byVolumType;
public byte byVolumNum;
public byte byStreamType;
public int dwFileIndex;
public byte byAudioFile;
}

```

Members

dwSize

结构体大小

struIDInfo

流信息，流 ID 或者通道号

struBeginTime

开始时间

struEndTime

结束时间

hWnd

回放的窗口句柄，若置为空，SDK 仍能收到码流数据，但不解码显示

byDrawFrame

是否抽帧：0- 不抽帧，1- 抽帧

byVolumType

0-普通录像卷，1-存档卷，适用于 CVR 设备，普通卷用于通道录像，存档卷用于备份录像

byVolumeNum

存档卷号

byStreamType

码流类型：0- 主码流，1- 子码流，2- 码流三

dwFileIndex

存档卷上的录像文件索引，搜索存档卷录像时返回的值

byAudioFile

回放音频文件：0- 不回放音频文件，1- 回放音频文件，该功能需要设备支持，启动音频回放后只回放音频文件

byRes2

保留，置为 0

5.96 [NET_DVR_WIFIETHERNET](#):无线网口参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_DVR_WIFIETHERNET
{
    public byte[]    slpAddress = new byte[16];
    public byte[]    slpMask = new byte[16];
    public byte[]    byMACAddr = new byte[HCNetSDK.MACADDR_LEN];
    public int       dwEnableDhcp;
    public int       dwAutoDns;
    public byte[]    sFirstDns = new byte[16];
    public byte[]    sSecondDns = new byte[16];
    public byte[]    sGatewayIpAddr = new byte[16];
}
```

Members

slpAddress

设备 IP 地址

slpMask

掩码

byMACAddr

物理地址，仅获取不能设置

dwEnableDhcp

是否启动 DHCP: 0-不启动, 1-启动

dwAutoDns

如果启动 DHCP 是否自动获取 DNS: 0-不自动获取, 1-自动获取; 对于有线如果启动 DHCP, 目前自动获取 DNS

sFirstDns

第一个 DNS 域名

sSecondDns

第二个 DNS 域名

sGatewayIpAddr

网关地址

5.97 NET_DVR_WIFI_CFG:wifi 配置参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_DVR_WIFI_CFG extends NET_DVR_CONFIG
{
    NET\_DVR\_WIFIETHERNET struetherNet = new NET_DVR_WIFIETHERNET();
    public byte[]          sEssid = new byte[HCNetSDK.IW_ESSID_MAX_SIZE];
    public int             dwMode;
    public int             dwSecurity;
    public WEP             wep = new WEP();
    public WPA\_PSK          wpa_psk = new WPA_PSK();
    public WPA\_WPA2         wpa_wpa2 = new WPA_WPA2();
}
```

Members*struetherNet*

WIFI 网口参数

sEssid

SSID

dwMode

工作模式: 0-mange 模式, 1-ad-hoc 模式

dwSecurity

加密模式: 0- 不加密, 1- WEP 加密, 2- WPA-personal, 3- WPA-enterprise, 4- WPA2-personal, 5- WPA2-enterprise

wep

WEP 加密参数

wpa_psk

WPA-personal/WPA2-personal 加密参数

wpa_wpa2

WPA-enterprise/WPA2-enterpris 加密参数

5.98 NET_DVR_WIFI_CONNECT_STATUS:wifi 连接状态**In class com.hikvision.netsdk.HCNetSDK**

```
public class NET_DVR_WIFI_CONNECT_STATUS extends NET_DVR_CONFIG
{
    public byte    byCurStatus;
    public int     dwErrorCode;
}
```

Members*byCurStatus*

WIFI 连接状态: 1- 已连接, 2- 未连接, 3- 正在连接

dwErrorCode

错误号, byCurStatus==2 时有效: 1- 用户名或密码错误, 2- 无此路由器, 3- 未知错误

5.99 NET_DVR_WORKSTATE_V30:设备工作状态参数**In class com.hikvision.netsdk.HCNetSDK**

```
public class NET_DVR_WORKSTATE_V30
{
    public int                                dwDeviceStatic;
    public NET\_DVR\_DISKSTATE\[\]             struHardDiskStatic = new
NET_DVR_DISKSTATE[HCNetSDK.MAX_DISKNUM_V30];
    public NET\_DVR\_CHANNELSTATE\_V30\[\]        struChanStatic = new
NET_DVR_CHANNELSTATE_V30[HCNetSDK.MAX_CHANNUM_V30];
    public byte[]                             byAlarmInStatic = new byte[HCNetSDK.MAX_ALARM_IN_V30];
}
```

```

public byte[]                byAlarmOutStatic = new byte[HCNetSDK.MAX_ALARMOUT_V30];
public int                   dwLocalDisplay;
public byte[]                byAudioChanStatus = new byte[HCNetSDK.MAX_AUDIO_V30];

public NET_DVR_WORKSTATE_V30(){
    for(int i=0; i<HCNetSDK.MAX_DISKNUM_V30; i++){
        struHardDiskStatic[i] = new NET_DVR_DISKSTATE();
    }
    for(int i=0; i<HCNetSDK.MAX_CHANNUM_V30; i++){
        struChanStatic[i] = new NET_DVR_CHANNELSTATE_V30();
    }
}
}

```

Members

dwDeviceStatic

设备的状态：0- 正常；1- CPU 占用率太高，超过 85%；2- 硬件错误，例如串口异常

struHardDiskStatic

硬盘状态

struChanStatic

通道状态

byAlarmInStatic

报警输入口的状态：0-没有报警；1-有报警

byAlarmOutStatic

报警输出口的状态：0-没有输出，1-有报警输出

dwLocalDisplay

本地显示状态：0-正常，1-不正常

byAudioChanStatus

表示语音通道的状态：0-未使用，1-使用中，0xff 无效

5.100 NET_DVR_ZEROCHANCFG:零通道压缩参数

In class com.hikvision.netsdk.HCNetSDK

```

public class NET_DVR_ZEROCHANCFG extends NET_DVR_CONFIG
{
    public byte    byEnable;
    public int     dwVideoBitrate;
    public int     dwVideoFrameRate;
}

```

Members

byEnable

是否启用零通道编码：0-不启用，1-启用

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

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

5.101 NET_IPC_AUX_ALARMCFG:辅助报警参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_IPC_AUX_ALARMCFG extends NET_DVR_CONFIG
{
    public NET\_IPC\_SINGLE\_AUX\_ALARMCFG[] struAlarm = new
    NET_IPC_SINGLE_AUX_ALARMCFG[HCNetSDK.MAX_AUX_ALARM_NUM];

    public NET_IPC_AUX_ALARMCFG()
    {
        for(int i = 0; i < HCNetSDK.MAX_AUX_ALARM_NUM; i++)
        {
            struAlarm[i] = new NET_IPC_SINGLE_AUX_ALARMCFG();
        }
    }
}
```

Members

struAlarm

辅助报警参数, 每位数组表示一个辅助报警

5.102 NET_IPC_CALLHELP_ALARMCFG:呼救报警参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class NET_IPC_CALLHELP_ALARMCFG {
    public byte byAlarmHandle;
    public NET\_DVR\_HANDLEEXCEPTION\_V30 struAlarmHandleType = new NET_DVR_HANDLEEXCEPTION_V30();
    public byte[] byRelRecordChan = new byte[HCNetSDK.MAX_CHANNUM_V30];
}
```

Members

byAlarmHandle

是否处理: 0- 不处理, 1- 处理

struAlarmHandleType

处理方式

byRelRecordChan

报警触发的录像通道, 值为 1 表示触发该通道。例如: `byRelRecordChan[0]==1` 表示触发通道 1 录像, `byRelRecordChan[1]==1`, 表示触发通道 2 录像, 依次类推

5.103 NET_IPC_PIR_ALARMCFG:PIR 报警参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_IPC_PIR_ALARMCFG
{
    public byte[]                byAlarmName = new byte[HCNetSDK.NAME_LEN];
    public byte                  byAlarmHandle;
    public NET\_DVR\_HANDLEEXCEPTION\_V30 struAlarmHandleType = new
NET_DVR_HANDLEEXCEPTION_V30();
    public byte[]                byRelRecordChan = new byte[HCNetSDK.MAX_CHANNUM_V30];
    public NET\_DVR\_SCHEDTIME[][] struAlarmTime = new
NET_DVR_SCHEDTIME[HCNetSDK.MAX_DAYS][HCNetSDK.MAX_TIMESEGMENT_V30];

    public NET_IPC_PIR_ALARMCFG()
    {
        for(int i = 0; i < HCNetSDK.MAX_DAYS; i++)
        {
            for(int j = 0; j < HCNetSDK.MAX_TIMESEGMENT_V30; j++)
            {
                struAlarmTime[i][j] = new NET_DVR_SCHEDTIME();
            }
        }
    }
}
```

Members

byAlarmName

报警名称

byAlarmHandle

是否处理：0- 不处理，1- 处理

struAlarmHandleType

处理方式

byRelRecordChan

报警触发的录像通道，按位表示，为 1 表示触发该通道。例如：byRelRecordChan[0]==1，表示触发通道 1 录像；byRelRecordChan[1]==1，表示触发通道 2 录像，依次类推

struAlarmTime

布防时间，每周 7 天，每天最多 8 个时间段

5.104 NET_IPC_SINGLE_AUX_ALARMCFG:单个辅助报警参数

In class com.hikvision.netsdk.HCNetSDK

```
public class NET_IPC_SINGLE_AUX_ALARMCFG
{
```

```

public byte                                byAlarmType;
public NET\_IPC\_PIR\_ALARMCFG                struPIRArm = new NET_IPC_PIR_ALARMCFG();
public NET\_IPC\_SINGLE\_WIRELESS\_ALARMCFG[] struWirelessAlarm = new
NET_IPC_SINGLE_WIRELESS_ALARMCFG[HCNetSDK.MAX_WIRELESS_ALARM_NUM];
public NET\_IPC\_CALLHELP\_ALARMCFG            struCallHelpAlarm = new NET_IPC_CALLHELP_ALARMCFG();

public NET_IPC_SINGLE_AUX_ALARMCFG()
{
    for(int i = 0; i < HCNetSDK.MAX_WIRELESS_ALARM_NUM; i++)
    {
        struWirelessAlarm[i] = new NET_IPC_SINGLE_WIRELESS_ALARMCFG();
    }
}
}

```

Members

byAlarmType

报警器类型，定义详见下表

```

enum _IPC_AUX_ALARM_TYPE_ {
    IPC_AUXALARM_UNKNOW = 0,
    IPC_AUXALARM_PIR      = 1,
    IPC_AUXALARM_WIRELESS = 2,
    IPC_AUXALARM_CALLHELP = 3
}IPC_AUX_ALARM_TYPE

```

IPC_AUXALARM_UNKNOW

未知

IPC_AUXALARM_PIR

PIR 报警

IPC_AUXALARM_WIRELESS

无线报警

IPC_AUXALARM_CALLHELP

呼救报警

struPIRArm

PIR 报警参数

struWirelessAlarm

无线报警参数

struCallHelpAlarm

呼救报警参数

5.105 **NET_IPC_SINGLE_WIRELESS_ALARMCFG**:单个无线报警参数

In class **com.hikvision.netsdk.HCNetSDK**

```

public class NET_IPC_SINGLE_WIRELESS_ALARMCFG
{

```

```

    public byte[]
    public byte
    public byte
    public NET\_DVR\_HANDLEEXCEPTION\_V30 struAlarmHandleType = new
NET_DVR_HANDLEEXCEPTION_V30();
    public byte[]
    byRelRecordChan = new byte[HCNetSDK.MAX_CHANNUM_V30];
}

```

Members

byAlarmName

报警名称

byAlarmHandle

是否处理：0- 不处理，1- 处理

byID

无线报警 ID，取值范围：1~8

struAlarmHandleType

处理方式

byRelRecordChan

报警触发的录像通道，按位表示，为 1 表示触发该通道。例如：byRelRecordChan[0]==1，表示触发通道 1 录像；byRelRecordChan[1]==1，表示触发通道 2 录像，依次类推

5.106 **NET_ITS_PICTURE_INFO**:抓拍图片信息

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

public static class NET_ITS_PICTURE_INFO extends Structure

```

{
    public int
    public byte
    public byte
    public byte
    public byte
    public int
    public byte[]
    public NET\_VCA\_RECT
    public NET\_VCA\_RECT
    public ByteBuffer
    public int
    public byte
    public byte[]

    dwDataLen;
    byType;
    byDataType;
    byCloseUpType;
    byPicRecogMode;
    dwRedLightTime;
    byAbsTime = new byte[32];
    struPlateRect;
    struPlateRecgRect;
    pBuffer;
    dwUTCTime;
    byCompatibleAblity;
    byRes2 = new byte[7];

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("dwDataLen", "byType", "byDataType", "byCloseUpType", "byPicRecogMode",

```

```

        "dwRedLightTime", "byAbsTime", "struPlateRect", "struPlateRecgRect", "pBuffer", "dwUTCTime",
        "byCompatibleAblity", "byRes2");
    }
}

```

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 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（byType 为 1 即场景图时该参数有效，用户根据位置自己截取车牌特写图）

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。
- 图片云存储 URL 格式：
http://CVMIP:Port/pic?did=DevID&bid=BlkID&pid=PictureID&ptime=PicTime
CVMIP: CVM(云存储服务器)的 IP 地址
Port: CVM(云存储服务器)对外提供 http 服务的端口（固定 8009）

DevID: CVS(云存储服务器)中设备 ID 号

BlkID: CVS(云存储服务器)中设备的块号

PictureID: CVS(云存储服务器)为图片生成的编号

PicTime: 图片的时间戳

示例:

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

5.107 NET_ITS_PLATE_RESULT:车辆识别结果

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_ITS_PLATE_RESULT extends Structure
{
    public int                dwSize;
    public int                dwMatchNo;
    public byte               byGroupNum;
    public byte               byPicNo;
    public byte               bySecondCam;
    public byte               byFeaturePicNo;
    public byte               byDriveChan;
    public byte               byVehicleType;
    public byte               byDetSceneID;
    public byte               byVehicleAttribute;
    public short              wIllegalType;
    public byte[]             byIllegalSubType = new byte[8];
    public byte               byPostPicNo;
    public byte               byChanIndex;
    public short              wSpeedLimit;
    public byte[]             byRes2 = new byte[2];
    public NET\_DVR\_PLATE\_INFO struPlateInfo = new NET_DVR_PLATE_INFO();
    public NET\_DVR\_VEHICLE\_INFO struVehicleInfo = new NET_DVR_VEHICLE_INFO();
    public byte[]             byMonitoringSiteID = new byte[48];
    public byte[]             byDeviceID = new byte[48];
    public byte               byDir;
    public byte               byDetectType;
    public byte               byRelaLaneDirectionType;
    public byte               byRes3;
    public int                dwCustomIllegalType;
    public byte               byIllegalFromatType;
    public ByteBuffer          pIllegalInfoBuf;
    public byte[]             byRes4 = new byte[4];
    public byte               byDataAnalysis;
    public byte               byYellowLabelCar;
```

```

public byte                byDangerousVehicles;
public byte                byPilotSafebelt;
public byte                byCopilotSafebelt;
public byte                byPilotSunVisor;
public byte                byCopilotSunVisor;
public byte                byPilotCall;
public byte                byBarrierGateCtrlType;
public byte                byAlarmDataType;
public NET\_DVR\_TIME\_V30    struSnapFirstPicTime = new NET_DVR_TIME_V30();
public int                 dwIllegalTime;
public int                 dwPicNum;
public NET\_ITS\_PICTURE\_INFO[] struPicInfo = (NET_ITS_PICTURE_INFO[])new
NET_ITS_PICTURE_INFO().toArray(6);

public NET_ITS_PLATE_RESULT(Pointer p){
    super(p);
}
@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("dwSize", "dwMatchNo", "byGroupNum", "byPicNo", "bySecondCam",
        "byFeaturePicNo", "byDriveChan", "byVehicleType", "byDetSceneID", "byVehicleAttribute",
        "wIllegalType", "byIllegalSubType", "byPostPicNo", "byChanIndex", "wSpeedLimit", "byRes2",
        "struPlateInfo", "struVehicleInfo", "byMonitoringSiteID", "byDeviceID", "byDir", "byDetectType",
        "byRelaLaneDirectionType", "byRes3", "dwCustomIllegalType", "byIllegalFromatType",
        "pIllegalInfoBuf", "byRes4", "byDataAnalysis", "byYellowLabelCar", "byDangerousVehicles",
        "byPilotSafebelt", "byCopilotSafebelt", "byPilotSunVisor", "byCopilotSunVisor", "byPilotCall",
        "byBarrierGateCtrlType", "byAlarmDataType", "struSnapFirstPicTime", "dwIllegalTime", "dwPicNum",
        "struPicInfo");
}
}

```

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，11- 机动车，12- 非机动车，13- 小型轿车，14- 微型轿车，15- 皮卡车

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

willlegalType

违章类型，采用国标定义。willlegalType 为 0 时违法类型见 dwCustomIllegalType，willlegalType 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

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（反向），2-下行(正向)，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：0-车辆检测(不区分具体的车辆检测算法)，1-地感触发，2-视频触发，3-多帧识别，4-雷达触发，5-混行检测

byRelaLaneDirectionType

关联车道方向类型（与关联车道号对应，确保车道唯一性）：

0- 其它，1- 从东向西，2- 从西向东，3- 从南向北，4- 从北向南，5- 从东南向西北，6- 从西北向东南，7- 从东北向西南，8-从西南向东北

byRes3

保留

dwCustomIllegalType

违章类型定义(用户自定义): 当 *wIllegalType* 参数为 0 时, 使用该参数; 若 *wIllegalType* 参数不为 0 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

plIllegalInfoBuf

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: 0- 未知, 1- 系安全带, 2- 未系安全带

byPilotSunVisor

主驾驶是否打开遮阳板: 0- 未知, 1- 未打开遮阳板, 2- 打开遮阳板

byCopilotSunVisor

副驾驶是否打开遮阳板: 0- 未知, 1- 未打开遮阳板, 2- 打开遮阳板

byPilotCall

主驾驶员是否在打电话: 0- 未知, 1- 未打电话, 2- 打电话

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黑白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)

dwIllegalTime

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

一般情况下车辆类型以 *NET_ITS_PLATE_RESULT* 结构体中的 *byVehicleType* 字段为准; 若该字段为 0 时, 则以 *NET_DVR_VEHICLE_INFO* 结构体中的 *byVehicleType* 字段为准, 此时设备侧可以至少保证给出大车、小车类型。

5.108 NET_VCA_ADV_REACH_HEIGHT:折线攀高参数

In class *com.hcnetsdk.jna.HCNetSDKByJNA*

```
public static class NET_VCA_ADV_REACH_HEIGHT extends Structure
{
```



```

public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
public int                  dwCrossDirection;
public byte[]               byRes = new byte[4];

@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("struRegion", "dwCrossDirection", "byRes");
}
}

```

Members

struRegion

区域范围

dwCrossDirection

跨越方向：0- 双向，1- 由左至右，2- 由右至左

byRes

保留

5.109 [NET_VCA_ADV_TRAVERSE_PLANE](#):折线警戒面参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_ADV_TRAVERSE_PLANE extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public int                  dwCrossDirection;
    public byte                  bySensitivity;
    public byte[]               byRes = new byte[3];

    public NET_VCA_ADV_TRAVERSE_PLANE(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "dwCrossDirection", "bySensitivity", "byRes");
    }
}

```

Members

struRegion

折线警戒面

dwCrossDirection

跨越方向： 0- 双向， 1- 由左至右， 2- 由右至左

bySensitivity

灵敏度，取值范围： [1,5]

byRes

保留

5.110 NET_VCA_AREA:区域参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_AREA extends Structure {
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public byte                  byDetectionTarget;
    public byte[]                byRes = new byte[7];

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "byDetectionTarget", "byRes");
    }
}
```

Members

struRegion

区域范围

byDetectionTarget

检测目标： 0- 所有目标， 1- 人， 2- 车

byRes

保留，置为 0

5.111 NET_VCA_AUDIO_ABNORMAL:音频异常参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_AUDIO_ABNORMAL extends Structure
{
    public short    wDecibel;
    public byte     bySensitivity;
    public byte     byAudioMode;
    public byte     byEnable;
    public byte     byThreshold;
    public byte[]   byRes = new byte[54];

    public NET_VCA_AUDIO_ABNORMAL(Pointer p) {
        super(p);
    }
}
```

```

        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("wDecibel", "bySensitivity", "byAudioMode", "byEnable", "byThreshold", "byRes");
    }
}

```

Members

wDecibel

声音强度，保留

bySensitivity

灵敏度参数，取值范围：[1,100]

byAudioMode

声音检测模式：0- 启用灵敏度检测，1- 启用分贝阈值检测，2- 都启用

byEnable

是否开启声强陡升侦测功能：0- 否，1- 是

byThreshold

声音强度阈值，取值范围：[1,100]

byRes

保留，置为 0

5.112 NET_VCA_BLACKLIST_INFO:授权名单信息结构体

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_BLACKLIST_INFO extends Structure {
    public int                dwSize;
    public int                dwRegisterID;
    public int                dwGroupNo;
    public byte               byType;
    public byte               byLevel;
    public byte[]             byRes1= new byte[2];
    public NET\_VCA\_HUMAN\_ATTRIBUTE struAttribute;
    public byte[]             byRemark = new byte[NAME_LEN];
    public int                dwFDDescriptionLen;
    public byte               *pFDDescriptionBuffer;

    public byte[]             byRes2= new byte[20];
}NET_VCA_BLACKLIST_INFO_ALARM, *LPNET_VCA_BLACKLIST_INFO_ALARM;

```

Members

dwSize

结构体大小

dwRegisterID

名单注册 ID 号（只读）

dwGroupNo

分组号

byType

黑白名单标志：0- 全部，1- 白名单，2- 授权名单

byLevel

授权名单等级：0- 全部，1- 低，2- 中，3- 高

byRes1

保留

struAttribute

人员信息

byRemark

备注信息

dwFDDescriptionLen

人脸库描述数据长度

pFDDescriptionBuffer

人脸库描述数据指针

byRes2

保留

5.113 NET_VCA_BLACKLIST_INFO_ALARM:授权名单报警信息

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_BLACKLIST_INFO_ALARM extends Structure {
    public NET\_VCA\_BLACKLIST\_INFO    struBlackListInfo;
    public int                          dwBlackListPicLen;
    public int                          dwFDIDLen;
    public byte                         *pFDID;
    public int                          dwPIDLen;
    public byte                         *pPID;
    public short                        wThresholdValue;
    public byte[]                       byRes = new byte[4];
    public byte                         *pBuffer1;
}
```

Members

struBlackListInfo

授权名单基本信息

dwBlackListPicLen

授权名单人脸子图的长度，为 0 表示没有图片，大于 0 表示有图片

dwFDIDLen

人脸库 ID 长度

pFDID

人脸库 ID 数据缓冲区指针

dwPIDLen

人脸库图片 ID 长度

pPID

人脸库图片 ID 指针

wThresholdValue;

人脸库阈值[0,100]

byRes

保留

pBuffer1

授权名单人脸子图的图片数据

5.114 **NET_VCA_COMBINED_RULE**:组合规则参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_COMBINED_RULE extends Structure {
    public byte                byRuleSequence;
    public byte[]              byRes = new byte[7];
    public int                  dwMinInterval;
    public int                  dwMaxInterval;
    public NET\_VCA\_RELATE\_RULE\_PARAM struRule1Raram = new NET_VCA_RELATE_RULE_PARAM();
    public NET\_VCA\_RELATE\_RULE\_PARAM struRule2Raram = new NET_VCA_RELATE_RULE_PARAM();
    public byte[]              byRes1 = new byte[36];

    public NET_VCA_COMBINED_RULE(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("byRuleSequence", "byRes", "dwMinInterval", "dwMaxInterval", "struRule1Raram",
            "struRule2Raram", "byRes1");
    }
}
```

Members

byRuleSequence

规则触发顺序：0- 顺序或逆序触发，1- 顺序触发

byRes

保留，置为 0

dwMinInterval

最小时间间隔，单位：秒

dwMaxInterval

最大时间间隔，单位：秒

struRule1Raram

规则 1

struRule2Raram

规则 2

byRes1

保留，置为 0

5.115 **NET_VCA_DEV_INFO**:前端设备信息

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_DEV_INFO extends Structure{
    public NET\_DVR\_IPADDR struDevIP = new NET_DVR_IPADDR();
    public short          wPort;
    public byte           byChannel;
    public byte           byIvmsChannel;

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struDevIP", "wPort", "byChannel", "byIvmsChannel");
    }
}
```

Members

struDevIP

报警通道对应设备的 IP 地址

wPort

报警通道对应设备的端口号

byChannel

报警通道对应设备的通道号，参数值即表示通道号。比如，`byChannel=1`，表示通道 1。

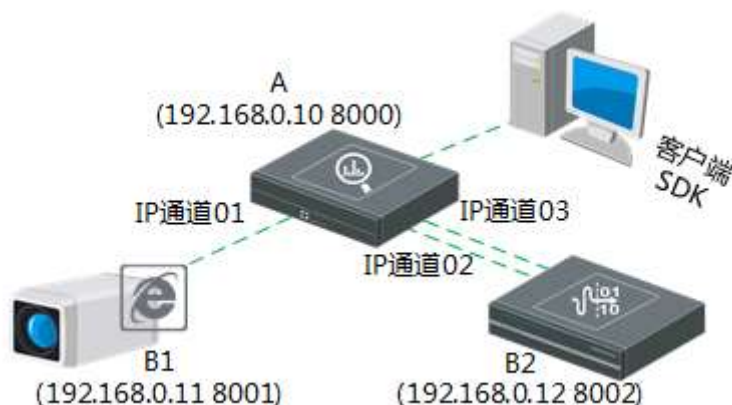
byIvmsChannel

SDK 接入设备的通道号

Remarks

如下图所示，我们 SDK 对接设备 A。

- 当接入设备(IPC/DVR/DVS/IVMS 等)的模拟通道报警时，`struDevIP`、`wPort` 为接入设备的 IP 地址和端口号，`byChannel`、`byIvmsChannel` 均为报警对应模拟通道的通道号。
- 当接入设备(HDVR/NVR/IVMS)的数字通道（IP 通道）报警时，`struDevIP`、`wPort`、`byChannel` 表示数字通道前端接入设备的 IP、端口和通道号，`byIvmsChannel` 为数字通道。如图，B1 设备通道 1 和 B2 设备的通道 1、2 分别接入 A 作为通道 1、2、3。则 `struDevIP`、`wPort`、`byChannel` 表示的是 B1 或者 B2 的 IP 地址、端口号和通道号，`byIvmsChannel` 表示 A 设备自己的数字通道号。比如 B2 设备的通道 2 发生了异常行为识别报警信息，则 SDK 接收到的报警信息中，则 `struDevIP=192.168.0.12`，`wPort=8002`，`byChannel=2`，`byIvmsChannel=3`。



5.116 NET_VCA_EVENT_UNION:事件参数联合体

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_EVENT_UNION extends Union
{
    public int[] uLen = new int[23];
}
```

Members

uLen

联合体长度，23*4 共 92 字节

Remarks

[NET_VCA_RULE_INFO](#) 里面 `wEventTypeEx` 表示事件类型，不同的事件类型对应不同的事件参数，联合体对应不同的结构体内容，详见表 5.2。由于事件类型众多，如果直接 JNA 联合体方式定义，列出所有结构体，数据转换较慢，会影响使用，因此此处我们仅给出联合体大小的参数，应用层需要自己根据事件类型转成对应结构体。

示例代码如下所示：

```
public class FMSGCallback implements HCNetSDKByJNA.FMSGCallback
{
    @Override
    public void invoke(int ICommand, NET_DVR_ALARMER pAlarmer, Pointer pAlarmInfo, int dwBufLen, Pointer pUser)
    {
        //字段顺序必须跟 C++结构体参数顺序一致
        System.out.println("alarm type:" + ICommand);
        if(ICommand == HCNetSDKByJNA.COMM_ALARM_V30)
        {
            HCNetSDKByJNA.NET_DVR_ALARMINFO_V30 struAlarmInfo = new
            HCNetSDKByJNA.NET_DVR_ALARMINFO_V30(pAlarmInfo);
            struAlarmInfo.read();
            System.out.println("COMM_ALARM_V30 alarm type:" + struAlarmInfo.dwAlarmType);
        }
    }
}
```

```
else if(lCommand == HCNetSDKByJNA.COMM_ALARM_V40)
{
    HCNetSDKByJNA.NET_DVR_ALARMINFO_V40 struAlarmInfo = new
HCNetSDKByJNA.NET_DVR_ALARMINFO_V40(pAlarmInfo);
    struAlarmInfo.read();
    System.out.println("COMM_ALARM_V40 alarm type:" +
struAlarmInfo.struAlarmFixedHeader.dwAlarmType);
}
else if(lCommand == HCNetSDKByJNA.COMM_UPLOAD_PLATE_RESULT)
{
    HCNetSDKByJNA.NET_DVR_PLATE_RESULT struAlarmInfo = new
HCNetSDKByJNA.NET_DVR_PLATE_RESULT(pAlarmInfo);
    struAlarmInfo.read();

    try {
        SimpleDateFormat sDateFormat = new SimpleDateFormat("yyyy-MM-dd-hh:mm:ss");
        String date = sDateFormat.format(new java.util.Date());
        FileOutputStream file = new FileOutputStream("/mnt/sdcard/" + date + ".bmp");
        file.write(struAlarmInfo.pBuffer1.getPointer().getByteArray(0, struAlarmInfo.dwPicLen), 0,
struAlarmInfo.dwPicLen);
        file.close();
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    System.out.println("COMM_UPLOAD_PLATE_RESULT license:" +
CommonMethod.toValidString(new String(struAlarmInfo.struPlateInfo.sLicense)));
}
else if(lCommand == HCNetSDKByJNA.COMM_ITS_PLATE_RESULT)
{
    HCNetSDKByJNA.NET_ITS_PLATE_RESULT struAlarmInfo = new
HCNetSDKByJNA.NET_ITS_PLATE_RESULT(pAlarmInfo);
    struAlarmInfo.read();
    System.out.println("COMM_ITS_PLATE_RESULT license:" + CommonMethod.toValidString(new
String(struAlarmInfo.struPlateInfo.sLicense)));
}
else if(lCommand == HCNetSDKByJNA.COMM_ALARM_RULE)
{
    HCNetSDKByJNA.NET_VCA_RULE_ALARM struAlarmInfo = new
HCNetSDKByJNA.NET_VCA_RULE_ALARM(pAlarmInfo);
    struAlarmInfo.read();
    if(struAlarmInfo.struRuleInfo.wEventTypeEx ==
HCNetSDKByJNA.ENUM_VCA_EVENT_EXIT_AREA)
```



```

        {
            //离开区域报警信息，联合体内容相应转成结构体 NET_VCA_AREA
            HCNetsdkByJNA.NET_VCA_AREA struExit = new
HCNetsdkByJNA.NET_VCA_AREA(struAlarmInfo.struRuleInfo.uEventParam.getPointer());
            struExit.read();
        }

        System.out.println("COMM_ALARM_RULE rule name:" + CommonMethod.toValidString(new
String(struAlarmInfo.struRuleInfo.byRuleName)));
    }
}
}
}

```

5.117 NET_VCA_FACESNAP_INFO_ALARM: 人脸抓拍信息

In class com.hikvision.netsdk.HCNetSDK

```

public static class NET_VCA_FACESNAP_INFO_ALARM extends Structure
{
    public int                dwRelativeTime;
    public int                dwAbsTime;
    public int                dwSnapFacePicID;
    public int                dwSnapFacePicLen;
    public NET\_VCA\_DEV\_INFO struDevInfo;
    public byte               byFaceScore;
    public byte               bySex;
    public byte               byGlasses;
    public byte               byAge;
    public byte               byAgeDeviation;
    public byte[]             byRes1= new byte[3];
    public int                dwUIDLen;
    public byte               *pUIDBuffer;
    public byte[]             byRes = new byte[4];
    public byte               *pBuffer1;
}NET_VCA_FACESNAP_INFO_ALARM, *LPNET_VCA_FACESNAP_INFO_ALARM;

```

Members

dwRelativeTime

相对时标

dwAbsTime

绝对时标

dwSnapFacePicID

抓拍人脸图 ID

dwSnapFacePicLen

抓拍人脸子图的长度，为 0 表示没有图片，大于 0 表示有图片

struDevInfo

前端设备信息

byFaceScore

人脸评分，指人脸子图的质量的评分，取值范围：0~100

bySex

性别：0- 未知，1- 男，2- 女

byGlasses

是否带眼镜：0- 未知，1- 是，2- 否

byAge

年龄

byAgeDeviation

年龄误差值，如 *byAge* 为 15 且 *byAgeDeviation* 为 1，则表示实际人脸图片年龄的为 14~16 之间

byRes1

保留

dwUIDLen

上传报警的标识长度

pUIDBuffer

缓冲区指针，存放上传报警的标识信息，信息相同表示同一次报警上传的结果

byRes

保留

pBuffer1

抓拍人脸子图的图片数据

5.118 NET_VCA_FACESNAP_RESULT:人脸抓拍结果

In class `com.hikvision.netsdk.HCNetSDK`

public static class `NET_VCA_FACESNAP_RESULT` extends `Structure`

```
{
    public int                dwSize;
    public int                dwRelativeTime;
    public int                dwAbsTime;
    public int                dwFacePicID;
    public int                dwFaceScore;
    public NET\_VCA\_TARGET\_INFO struTargetInfo;
    public NET\_VCA\_RECT       struRect;
    public NET\_VCA\_DEV\_INFO   struDevInfo;
    public int                dwFacePicLen;
    public int                dwBackgroundPicLen;
    public byte               bySmart;
    public byte               byAlarmEndMark;
    public byte               byRe= new byte[2];
}
```

```

public NET\_VCA\_HUMAN\_FEATURE    struFeature;
public int                      dwStayDuration;
public byte                     byRes2[18];
public short                    wDevInfolvmsChannelEx;
public byte[]                   byRes = new byte[16];
public byte                     *pBuffer1;
public byte                     *pBuffer2;
}

```

Members

dwSize

结构体大小

dwRelativeTime

相对时标

dwAbsTime

绝对时标

dwFacePicID

人脸图 ID

dwFaceScore

人脸评分，范围：0~100

struTargetInfo

报警目标信息

struRect

人脸子图区域

struDevInfo

前端设备信息

dwFacePicLen

人脸子图的长度，为 0 表示没有图片，大于 0 表示有图片

dwBackgroundPicLen

背景图的长度，为 0 表示没有图片，大于 0 表示有图片(保留)

bySmart

0- iDS 设备返回（默认值），1- SMART 设备返回

byAlarmEndMark

报警结束标记：0- 保留，1- 结束标记（该字段结合人脸 ID 字段使用，表示该 ID 对应的下报警结束，用于判断报警结束，提取识别图片数据中，清晰度最高的图片）

byRes

保留

struFeature

人体属性

dwStayDuration

停留画面中时间，单位：毫秒

byRes2

保留

wDevInfolvmsChannelEx

与 NET_VCA_DEV_INFO 里的 bylvmsChannel 含义相同，能表示更大的值。老客户端用 bylvmsChannel 能

继续兼容，但是最大到 255。新客户端版本请使用 `wDevInfolvmsChannelEx`

byRes1

保留

**pBuffer1*

人脸子图的图片数据

**pBuffer2*

背景图的图片数据

5.119 NET_VCA_FACESNAP_MATCH_ALARM:授权名单比对结果

In class `com.hikvision.netsdk.HCNetSDK`

public static class `NET_VCA_FACESNAP_MATCH_ALARM` extends `Structure`

```
{
    public int                dwSize;
    public float              fSimilarity;
    NET\_VCA\_FACESNAP\_INFO\_ALARM struSnapInfo;
    NET\_VCA\_BLACKLIST\_INFO\_ALARM struBlackListInfo;
    public char               sStorageIP[16];
    public short              wStoragePort;
    public byte               byMatchPicNum;
    public byte               byPicTransType;
    public int                dwSnapPicLen;
    public byte               *pSnapPicBuffer;
    public NET_VCA_RECT      struRegion;
    public int                dwModelDataLen;
    public byte               *pModelDataBuffer;
    public byte[]             byRes = new byte[8];
}
```

Members

dwSize

结构体大小

fSimilarity

相似度，取值范围：[0.001,1]

struSnapInfo

人脸抓拍上传信息

struBlackListInfo

授权名单报警信息

sStorageIP

存储服务 IP 地址

wStoragePort

存储服务端口号

byMatchPicNum

匹配图片的数量，0 是保留值（不支持该字段的设备，该值默认为 0；支持该字段的设备，该值为 0 时表示后续没有匹配的图片信息）

byPicTransType

图片数据传输方式: 0- 二进制，1- URL 路径

dwSnapPicLen

设备识别抓拍图片长度

pSnapPicBuffer

设备识别抓拍图片指针

struRegion

设备识别抓拍图片中人脸子图坐标，可以根据该坐标从抓拍图片上抠取人脸小图片

dwModelDataLen

建模数据长度

pModelDataBuffer

建模数据指针

byRes

保留

5.120 NET_VCA_FALL_DOWN:倒地参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_FALL_DOWN extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte                 bySensitivity;
    public byte                 byHeightThreshold;
    public byte[]               byRes = new byte[4];

    public NET_VCA_FALL_DOWN(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wDuration", "bySensitivity", "byHeightThreshold", "byRes");
    }
}
```

Members

struRegion

区域范围

wDuration

触发事件阈值，单位：秒，取值范围：[1,3600]

bySensitivity

灵敏度，取值范围：[1,5]

byHeightThreshold

高度阈值，取值范围：[0,250]，默认：90，单位：厘米

byRes

保留

5.121 NET_VCA_GET_UP:起身参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_GET_UP extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte                 byMode;
    public byte                 bySensitivity;
    public byte[]               byRes = new byte[4];

    public NET_VCA_GET_UP(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wDuration", "byMode", "bySensitivity", "byRes");
    }
}
```

Members

struRegion

区域范围

wDuration

触发事件阈值，取值范围：1~100 秒

byMode

起身检测模式：0-大床通铺模式，1-高低铺模式

bySensitivity

灵敏度参数，取值范围：[1,100]

byRes

保留

5.122 NET_VCA_HIGH_DENSITY: 人员聚集参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_HIGH_DENSITY extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public float                fDensity;
    public byte                 bySensitivity;
    public byte                 byRes;
    public short                wDuration;

    public NET_VCA_HIGH_DENSITY(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "fDensity", "bySensitivity", "byRes", "wDuration");
    }
}
```

Members

struRegion

区域范围

fDensity

密度比率，取值范围：[0.1,1.0]

bySensitivity

灵敏度参数，取值范围：[1,5]

byRes

保留，置为 0

wDuration

触发人员聚集参数报警阈值，20~360s

5.123 NET_VCA_HUMAN_ATTRIBUTE: 人员信息结构体

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_HUMAN_ATTRIBUTE extends Structure {
    public byte                bySex;
    public byte                byCertificateType;
    public byte[]              byBirthDate = new byte[MAX_HUMAN_BIRTHDATE_LEN];
    public byte[]              byName = new byte[NAME_LEN];
    public NET\_DVR\_AREAINFOCFG struNativePlace;
```

```

    public byte[]          byCertificateNumber = new byte[NAME_LEN];
    public int             dwPersonInfoExtendLen;
    public byte            *pPersonInfoExtend;
    public byte[]          byRes2= new byte[12];
}

```

Members

bySex

性别：0- 不启用，1- 男，2- 女，0xff- 未知

byCertificateType

证件类型：0- 不启用，1- 身份证，2- 警官证，0xff- 未知

byBirthDate

出生年月，如：201106

byName

姓名

struNativePlace

籍贯

byCertificateNumber

证件号

dwPersonInfoExtendLen

人员标签信息扩展长度

pPersonInfoExtend

人员标签信息扩展信息，对应 XML 数据结构：XML_PersonInfoExtendList

byRes2

保留

5.124 NET_VCA_HIGH_DENSITY_STATUS:人员聚集状态信息

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_HIGH_DENSITY_STATUS extends Structure {
    public NET\_VCA\_POLYGON struRegion = new NET_VCA_POLYGON();
    public float             fDensity;
    public byte              bySensitivity;
    public byte[]            byRes = new byte[3];

    public NET_VCA_HIGH_DENSITY_STATUS(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "fDensity", "bySensitivity", "byRes");
    }
}

```



```
    }
}
```

Members

struRegion

区域范围

fDensity

密度比率，取值范围[0.1,1.0]

bySensitivity

灵敏度参数，范围： [0,100]

byRes

保留，置为 0

5.125 NET_VCA_HUMAN_ENTER:人员进入参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

public static class NET_VCA_HUMAN_ENTER extends Structure

```
{
    public int[]    dwRes = new int[23];

    public NET_VCA_HUMAN_ENTER(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("dwRes");
    }
}
```

Members

dwRes

保留

5.126 NET_VCA_HUMAN_FEATURE:人体属性参数结构体

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

public static class NET_VCA_HUMAN_FEATURE extends Structure

```
{
    public byte                byAgeGroup;
    public byte                bySex;
    public byte                byEyeGlass;
    public byte                byAge;
```

```

    public byte                byAgeDeviation;
    public byte[] byRes        = new byte [11];
}

```

Members

byAgeGroup

年龄段，0xffffffff 表示全部（不关注年龄段），其他取值具体定义如下：

```

enum _HUMAN_AGE_GROUP_ENUM_{
    ENUM_AGE_GROUP_INFANT      = 1,
    ENUM_AGE_GROUP_CHILD       = 2,
    ENUM_AGE_GROUP_YOUNGSTER   = 3,
    ENUM_AGE_GROUP_ADOLESCENT  = 4,
    ENUM_AGE_GROUP_YOUTH       = 5,
    ENUM_AGE_GROUP_PRIME       = 6,
    ENUM_AGE_GROUP_MIDLIFE     = 7,
    ENUM_AGE_GROUP_MIDAGE      = 8,
    ENUM_AGE_GROUP_OLD         = 9
} HUMAN_AGE_GROUP_ENUM

```

ENUM_AGE_GROUP_INFANT

婴幼儿

ENUM_AGE_GROUP_CHILD

儿童

ENUM_AGE_GROUP_YOUNGSTER

少年

ENUM_AGE_GROUP_ADOLESCENT

青少年

ENUM_AGE_GROUP_YOUTH

青年

ENUM_AGE_GROUP_PRIME

壮年

ENUM_AGE_GROUP_MIDLIFE

中年

ENUM_AGE_GROUP_MIDAGE

中老年

ENUM_AGE_GROUP_OLD

老年

bySex

性别：1- 男，2- 女

byEyeGlass

是否戴眼镜：1- 不戴，2- 戴

byAge

年龄

byAgeDeviation

年龄误差值，如 *byAge* 为 15、*byAgeDeviation* 为 1，表示实际人脸图片年龄的为 14~16 之间

byRes

保留

5.127 NET_VCA_INTRUSION:区域入侵参数

In class com.hcnetsdk.jna.HCNetSDKByJNA

```
public static class NET_VCA_INTRUSION extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte                 bySensitivity;
    public byte                 byRate;
    public byte                 byDetectionTarget;
    public byte[]               byRes = new byte[3];

    public NET_VCA_INTRUSION(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wDuration", "bySensitivity", "byRate", "byDetectionTarget", "byRes");
    }
}
```

Members

struRegion

区域范围

wDuration

行为事件触发时间阈值，判断有效报警的时间。不同的设备取值范围不同，智能服务器为 1~120 秒（默认：5 秒），ATM 设备为 1~1200 秒，Smart IPC 为 1~100 秒

bySensitivity

灵敏度参数，取值范围：[1,100]

byRate

占比：区域内所有未报警目标尺寸目标占区域面积的比重，归一化为 1~100

byDetectionTarget

检测目标：0- 所有目标，1- 人，2- 车

byRes

保留，置为 0

5.128 NET_VCA_LEAVE_POSITION:离岗事件参数

In class com.hcnetsdk.jna.HCNetSDKByJNA

```

public static class NET_VCA_LEAVE_POSITION extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wLeaveDelay;
    public short                wStaticDelay;
    public byte                 byMode;
    public byte                 byPersonType;
    public byte[]               byRes = new byte[2];

    public NET_VCA_LEAVE_POSITION(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wLeaveDelay", "wStaticDelay", "byMode", "byPersonType", "byRes");
    }
}

```

Members

struRegion

区域范围

wLeaveDelay

无人报警时间，单位：秒，取值范围：1~1800

wStaticDelay

睡觉报警时间，单位：秒，取值范围：1~1800

byMode

模式：0- 离岗事件，1- 睡岗事件，2- 离岗睡岗事件

byPersonType

值岗人数类型：0-单人值岗，1-双人值岗

byRes

保留

5.129 **NET_VCA_LEFT**:丢包参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_LEFT extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte                 bySensitivity;
    public byte[]               byRes = new byte[5];
}

```

```

public NET_VCA_LEFT(Pointer p) {
    super(p);
    // TODO Auto-generated constructor stub
}
@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("struRegion", "wDuration", "bySensitivity", "byRes");
}
}

```

Members

struRegion

区域范围

wDuration

触发事件阈值，10~100 秒

bySensitivity

灵敏度，取值范围：[1,5]

byRes

保留

5.130 NET_VCA_LINE:线结构参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_LINE extends Structure
{
    public NET\_VCA\_POINT struStart = new NET_VCA_POINT();
    public NET\_VCA\_POINT struEnd = new NET_VCA_POINT();
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struStart", "struEnd");
    }
}

```

Members

struStart

起点

struEnd

终点

5.131 NET_VCA_LOITER:徘徊参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_LOITER extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte[]               byRes = new byte[6];

    public NET_VCA_LOITER(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wDuration", "byRes");
    }
}

```

Members

struRegion

区域范围

wDuration

触发徘徊报警的持续时间：1-120 秒，建议 10 秒

byRes

保留，置为 0

5.132 **NET_VCA_OVER_TIME**:操作超时参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_OVER_TIME extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte[]               byRes = new byte[6];

    public NET_VCA_OVER_TIME(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wDuration", "byRes");
    }
}

```

Members

struRegion

区域范围

wDuration

操作报警时间阈值

byRes

保留

5.133 NET_VCA_PARKING: 停车参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_PARKING extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte[]               byRes = new byte[6];

    public NET_VCA_PARKING(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wDuration", "byRes");
    }
}
```

Members

struRegion

区域范围

wDuration

触发停车报警的持续时间：1-120 秒，建议 10 秒

byRes

保留，置为 0

5.134 NET_VCA_PEOPLENUM_CHANGE: 人数变化参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_PEOPLENUM_CHANGE extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public byte                bySensitivity;
```

```

public byte                byPeopleNumThreshold;
public byte                byDetectMode;
public byte                byNoneStateEffective;
public short               wDuration;
public byte[]              byRes = new byte[2];

public NET_VCA_PEOPLENUM_CHANGE(Pointer p) {
    super(p);
    // TODO Auto-generated constructor stub
}

@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("struRegion", "bySensitivity", "byPeopleNumThreshold", "byDetectMode",
        "byNoneStateEffective", "wDuration", "byRes");
}
}

```

Members

struRegion

区域范围

bySensitivity

灵敏度，取值范围：[1,100]

byPeopleNumThreshold

人数阈值，取值范围：[0,5]，默认：1

byDetectMode

检测方式，与人数阈值相比较：1- 大于，2- 小于，3- 等于，4- 不等于

byNoneStateEffective

无人状态是否有效：0- 无效，1- 有效

wDuration

触发事件阈值，单位：秒，取值范围：1~3600，默认：2

byRes

保留

5.135 NET_VCA_POINT:点坐标参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_POINT extends Structure{
    public float  fX;
    public float  fY;

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
    }
}

```



```

        return Arrays.asList("fX", "fY");
    }
}

```

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]

5.136 NET_VCA_POLYGON:多边形参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_POLYGON extends Structure
{
    public int                dwPointNum;
    public NET\_VCA\_POINT[]  struPos = (NET_VCA_POINT[]) new NET_VCA_POINT().toArray(10);

    @Override
    protected List<String> getFieldOrder(){
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("dwPointNum", "struPos");
    }
}

```

Members

dwPointNum

有效点（大于等于 3），若是 3 点在一条线上认为是无效区域，线交叉认为是无效区域

struPos

多边形边界点，最大值为 10

5.137 NET_VCA_REACH_HIGHT:攀高参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_REACH_HIGHT extends Structure
{
    public NET\_VCA\_LINE    struVcaLine = new NET_VCA_LINE();
    public short           wDuration;
    public byte[]          byRes = new byte[6];

    public NET_VCA_REACH_HIGHT(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
}

```

```
@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("struVcaLine", "wDuration", "byRes");
}
}
```

Members

struVcaLine

攀高警戒面

wDuration

触发事件阈值，1-100 秒

byRes

保留

5.138 NET_VCA_RECT:区域框参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_RECT extends Structure {
    public float fX;
    public float fY;
    public float fWidth;
    public float fHeight;

    @Override
    protected List getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("fX", "fY", "fWidth", "fHeight");
    }
}
```

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]

5.139 NET_VCA_RELATE_RULE_PARAM:关联规则参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_RELATE_RULE_PARAM extends Structure {
    public byte    byRuleID;
    public byte    byRes;
    public short   wEventType;

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("byRuleID", "byRes", "wEventType");
    }
}

```

Members

byRuleID

规则序号，0 表示无

byRes

保留，置为 0

wEventType

行为事件类型，取值详见表 5.2

5.140 NET_VCA_RULE_ALARM:异常行为识别报警

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_RULE_ALARM extends Structure {
    public int                dwSize;
    public int                dwRelativeTime;
    public int                dwAbsTime;
    public NET\_VCA\_RULE\_INFO struRuleInfo = new NET_VCA_RULE_INFO();
    public NET\_VCA\_TARGET\_INFO struTargetInfo = new NET_VCA_TARGET_INFO();
    public NET\_VCA\_DEV\_INFO    struDevInfo = new NET_VCA_DEV_INFO();
    public int                dwPicDataLen;
    public byte               byPicType;
    public byte               byRelAlarmPicNum;
    public byte               bySmart;
    public byte               byRes;
    public int                dwAlarmID;
    public byte[]             byRes2 = new byte[8];
    public ByteByReference    pImage;

    public NET_VCA_RULE_ALARM(Pointer p){
        super(p);
    }

    @Override
    protected List<String> getFieldOrder() {

```

```

//字段顺序必须跟 C++结构体参数顺序一致
return Arrays.asList("dwSize", "dwRelativeTime", "dwAbsTime", "struRuleInfo", "struTargetInfo",
    "struDevInfo", "dwPicDataLen", "byPicType", "byRelAlarmPicNum", "bySmart", "byRes", "dwAlarmID",
    "byRes2", "pImage");
}
}

```

Members

dwSize

结构体大小

dwRelativeTime

相对时标，从开启智能到触发事件的时间

dwAbsTime

绝对时标

struRuleInfo

事件规则信息

struTargetInfo

报警目标信息

struDevInfo

前端设备信息

dwPicDataLen

返回图片的长度。为 0 表示没有图片，大于 0 表示该结构后面紧跟图片数据

byPicType

0- 普通图片，1- 对比图片

byRelAlarmPicNum

关联通道报警图片数量

bySmart

0- iDS 专业智能设备返回，1- Smart 设备返回

byRes

保留，置为 0

dwAlarmID

报警 ID，用以标识通道间关联产生的组合报警，0 表示无效

byRes2

保留，置为 0

pImage

指向图片的指针

Remarks

- 如当前报警通道存在关联通道时，针对同一报警事件，会对关联通道进行抓图并上传，这些图片数据分多次上传，其中报警事件信息一致。可通过字段 **dwAlarmID** 对这些报警进行区分组合，由 **byRelAlarmPicNum** 获取关联通道的抓拍图片数量，其中报警图片总数量为 **byRelAlarmPicNum + 1**，从而得到当前报警事件的多张图片。
- 相对时标暂未使用。从绝对时标 **dwAbsTime** 解析得到“年月日时分秒”的算法如下所示：

```

#define GET_YEAR(_time_)      (((_time_)>>26) + 2000)
#define GET_MONTH(_time_)    (((_time_)>>22) & 15)
#define GET_DAY(_time_)      (((_time_)>>17) & 31)

```

```
#define GET_HOUR(_time_)      (((_time_)>>12) & 31)
#define GET_MINUTE(_time_)    (((_time_)>>6) & 63)
#define GET_SECOND(_time_)    (((_time_)>>0) & 63)
```

5.141 NET_VCA_RULE_INFO:事件规则信息

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_RULE_INFO extends Structure {
    public byte                byRuleID;
    public byte                byRes;
    public short               wEventTypeEx;
    public byte[]              byRuleName = new byte[NAME_LEN];
    public int                 dwEventType;
    public NET\_VCA\_EVENT\_UNION uEventParam = new NET_VCA_EVENT_UNION();

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("byRuleID", "byRes", "wEventTypeEx", "byRuleName", "dwEventType",
            "uEventParam");
    }
}
```

Members

byRuleID

规则序号，为规则配置结构下标，0-7

byRes

保留，置为 0

wEventTypeEx

行为事件类型扩展，兼容字段 `dwEventType`，详见表 5.2

byRuleName

规则名称

dwEventType

警戒事件类型，请使用 `wEventTypeEx` 字段类型

uEventParam

事件参数联合体

Remarks

不同的事件类型，事件参数联合体对应不同的参数结构，如表 5.2 所示。

表 5.2 事件类型

wEventTypeEx 类型	类型值	描述	uEventParam 对应结构
ENUM_VCA_EVENT_TRAVERSE_PLANE	1	穿越警戒面（越界侦测）	NET_VCA_TRAVERSE_PLANE
ENUM_VCA_EVENT_ENTER_AREA	2	目标进入区域，支持区域规则	NET_VCA_AREA
ENUM_VCA_EVENT_EXIT_AREA	3	目标离开区域，支持区	NET_VCA_AREA

		域规则	
ENUM_VCA_EVENT_INTRUSION	4	周界入侵（区域入侵侦测），支持区域规则	NET_VCA_INTRUSION
ENUM_VCA_EVENT_LOITER	5	徘徊，支持区域规则	NET_VCA_LOITER
ENUM_VCA_EVENT_LEFT_TAKE	6	丢包捡包，支持区域规则	NET_VCA_TAKE_LEFT
ENUM_VCA_EVENT_PARKING	7	停车，支持区域规则	NET_VCA_PARKING
ENUM_VCA_EVENT_RUN	8	快速移动(奔跑)，支持区域规则	NET_VCA_RUN
ENUM_VCA_EVENT_HIGH_DENSITY	9	区域内人员密度，支持区域规则，人员聚集度超过设置的阈值时设备上传报警信息	NET_VCA_HIGH_DENSITY
ENUM_VCA_EVENT_VIOLENT_MOTION	10	剧烈运动检测	NET_VCA_VIOLENT_MOTION
ENUM_VCA_EVENT_REACH_HIGHT	11	攀高检测	NET_VCA_REACH_HIGHT
ENUM_VCA_EVENT_GET_UP	12	起身检测	NET_VCA_GET_UP
ENUM_VCA_EVENT_LEFT	13	物品遗留	NET_VCA_LEFT
ENUM_VCA_EVENT_TAKE	14	物品拿取	NET_VCA_TAKE
ENUM_VCA_EVENT_LEAVE_POSITION	15	离岗	NET_VCA_LEAVE_POSITION
ENUM_VCA_EVENT_TRAIL	16	尾随	NET_VCA_TRAIL
ENUM_VCA_EVENT_KEY_PERSON_GET_UP	17	重点人员起身检测	NET_VCA_GET_UP
ENUM_VCA_EVENT_STANDUP	18	起立检测	NET_VCA_STANDUP
ENUM_VCA_EVENT_FALL_DOWN	20	倒地检测	NET_VCA_FALL_DOWN
ENUM_VCA_EVENT_AUDIO_ABNORMAL	21	声强突变检测	NET_VCA_AUDIO_ABNORMAL
ENUM_VCA_EVENT_ADV_REACH_HEIGHT	22	折线攀高	NET_VCA_ADV_REACH_HEIGHT
ENUM_VCA_EVENT_TOILET_TARRY	23	如厕超时	NET_VCA_TOILET_TARRY
ENUM_VCA_EVENT_YARD_TARRY	24	放风场滞留	NET_VCA_YARD_TARRY
ENUM_VCA_EVENT_ADV_TRAVERSE_PLANE	25	折线警戒面	NET_VCA_ADV_TRAVERSE_PLANE
ENUM_VCA_EVENT_HUMAN_ENTER	29	人靠近 ATM（仅在 ATM_PANEL 模式下支持）	NET_VCA_HUMAN_ENTER
ENUM_VCA_EVENT_OVER_TIME	30	操作超时（仅在 ATM_PANEL 模式下支持）	NET_VCA_OVER_TIME
ENUM_VCA_EVENT_STICK_UP	31	贴纸条,支持区域规则	NET_VCA_STICK_UP
ENUM_VCA_EVENT_INSTALL_SCANNER	32	安装读卡器，支持区域规则	NET_VCA_SCANNER
ENUM_VCA_EVENT_PEOPLENUM_CHANGE	35	人数变化事件	NET_VCA_PEOPLENUM_CHANGE
ENUM_VCA_EVENT_SPACING_CHANGE	36	间距变化事件	NET_VCA_SPACING_CHANGE
ENUM_VCA_EVENT_COMBINED_RULE	37	组合规则事件	NET_VCA_COMBINED_RULE
ENUM_VCA_EVENT_HIGH_DENSITY_STATUS	39	区域内人员聚集状态，设备按照设置的时间间隔上传实时的人员聚集	NET_VCA_HIGH_DENSITY_STATUS

		状态信息，该时间间隔不支持通过 SDK 配置，需要通过服务器的配置文件来修改，默认：10s	
--	--	---	--

5.142 NET_VCA_RUN:快速移动参数

In class com.hcnetsdk.jna.HCNetSDKByJNA

public static class NET_VCA_RUN extends Structure

```
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public float                fRunDistance;
    public byte                 bySensitivity;
    public byte                 byMode;
    public byte                 byDetectionTarget;
    public byte                 byRes;

    public NET_VCA_RUN(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "fRunDistance", "bySensitivity", "byMode", "byDetectionTarget",
            "byRes");
    }
}
```

Members

struRegion

区域范围

fRunDistance

人奔跑最大距离，像素模式取值范围：[0.1,1.00]，实际模式取值范围：(1,20)m/s:

bySensitivity

灵敏度参数，取值范围：[1,5]

byMode

距离模式：0- 像素模式，1- 实际模式

byDetectionTarget

检测目标：0- 所有目标，1- 人，2- 车

byRes

保留，置为 0

5.143 NET_VCA_SCANNER:读卡器参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_SCANNER extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte                 bySensitivity;
    public byte[]               byRes = new byte[5];

    public NET_VCA_SCANNER(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wDuration", "bySensitivity", "byRes");
    }
}
```

Members

struRegion

区域范围

wDuration

读卡持续时间：4-60 秒，建议 10s 秒

bySensitivity

灵敏度参数，范围[1,5]

byRes

保留，置为 0

5.144 NET_VCA_SPACING_CHANGE:间距变化参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_SPACING_CHANGE extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public float                fSpacingThreshold;
    public byte                 bySensitivity;
    public byte                 byDetectMode;
    public short                wDuration;

    public NET_VCA_SPACING_CHANGE(Pointer p) {
```



```

        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "fSpacingThreshold", "bySensitivity", "byDetectMode", "wDuration");
    }
}

```

Members

struRegion

区域范围

fSpacingThreshold

间距阈值，取值范围：[0,10.0]，默认：1.0，单位：米

bySensitivity

灵敏度，取值范围：[1,100]

byDetectMode

检测方式，与人数阈值相比较：1- 大于，2- 小于

wDuration

触发事件阈值，单位：秒，取值范围：1~3600，默认：2

5.145 **NET_VCA_STANDUP**:起立参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_STANDUP extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public byte                bySensitivity;
    public byte                byHeightThreshold;
    public short               wDuration;
    public byte[]              byRes = new byte[4];

    public NET_VCA_STANDUP(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "bySensitivity", "byHeightThreshold", "wDuration", "byRes");
    }
}

```

Members

struRegion

区域范围

bySensitivity

灵敏度，取值范围：[1,100]

byHeightThreshold

高度阈值，取值范围：[0,250]，默认：130，单位：厘米

wDuration

触发事件阈值，单位：秒，取值范围：1~3600，默认：2

byRes

保留

5.146 NET_VCA_STICK_UP:贴纸条参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_STICK_UP extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte                 bySensitivity;
    public byte[]               byRes = new byte[5];

    public NET_VCA_STICK_UP(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wDuration", "bySensitivity", "byRes");
    }
}
```

Members

struRegion

区域范围

wDuration

触发时间阈值：4-60 秒，建议 10 秒

bySensitivity

灵敏度参数，范围[1,5]

byRes

保留，置为 0

5.147 **NET_VCA_TAKE**:捡包参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_TAKE extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte                 bySensitivity;
    public byte[]               byRes = new byte[5];

    public NET_VCA_TAKE_LEFT(Pointer p){
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder(){
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wDuration", "byRes");
    }
}
```

Members

struRegion

区域范围

wDuration

触发捡包报警阈值，10~100 秒

bySensitivity

灵敏度，取值范围：[1,5]

byRes

保留

5.148 **NET_VCA_TAKE_LEFT**:丢包/捡包参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_TAKE_LEFT extends Structure {
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte[]               byRes = new byte[6];

    public NET_VCA_TAKE_LEFT(Pointer p){
        super(p);
        // TODO Auto-generated constructor stub
    }
}
```

```

@Override
protected List<String> getFieldOrder(){
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("struRegion", "wDuration", "byRes");
}
}

```

Members

struRegion

区域范围

wDuration

触发丢包/捡包报警的持续时间：1-120 秒，建议 10 秒（如果 ATM 设备，时间为 4-60 秒）

byRes

保留，置为 0

5.149 **NET_VCA_TARGET_INFO**:报警目标信息

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_TARGET_INFO extends Structure
{
    public int          dwID;
    public NET\_VCA\_RECT struRect = new NET_VCA_RECT();
    public byte[]       byRes = new byte[4];

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("dwID", "struRect", "byRes");
    }
}

```

Members

dwID

目标 ID，人员密度过高报警时为 0

struRect

目标边界框

byRes

保留，置为 0

5.150 **NET_VCA_TOILET_TARRY**:如厕超时参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_TOILET_TARRY extends Structure
{

```

```

public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
public short                wDelay;
public byte[]               byRes = new byte[6];

public NET_VCA_TOILET_TARRY(Pointer p) {
    super(p);
    // TODO Auto-generated constructor stub
}
@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("struRegion", "wDelay", "byRes");
}
}

```

Members

struRegion

区域范围

wDelay

如厕超时时间，单位：秒，取值范围：[1,3600]

byRes

保留

5.151 [NET_VCA_TRAIL](#):尾随参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class NET_VCA_TRAIL extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wRes;
    public byte                 bySensitivity;
    public byte[]               byRes = new byte[5];

    public NET_VCA_TRAIL(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wRes", "bySensitivity", "byRes");
    }
}

```

Members

struRegion

区域范围

wRes

保留

bySensitivity

灵敏度，取值范围：[1,5]

byRes

保留

5.152 NET_VCA_TRAVERSE_PLANE: 穿越警戒面参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_TRAVERSE_PLANE extends Structure {
    public NET\_VCA\_LINE    struPlaneBottom = new NET_VCA_LINE();
    public int                dwCrossDirection;
    public byte               bySensitivity;
    public byte               byPlaneHeight;
    public byte               byDetectionTarget;
    public byte[]             byRes2 = new byte[37];

    public NET_VCA_TRAVERSE_PLANE(Pointer pointer){
        super(pointer);
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struPlaneBottom", "dwCrossDirection", "bySensitivity", "byPlaneHeight",
            "byDetectionTarget", "byRes2");
    }
}
```

Members

struPlaneBottom

警戒面底边

dwCrossDirection

穿越方向：0- 双向，1- 由左至右，2- 由右至左

bySensitivity

灵敏度，取值范围：[1,5] （对于 Smart IPC，取值范围为[1,100]）

byPlaneHeight

警戒面高度（网络摄像机不支持该参数设置）

byDetectionTarget

检测目标：0- 所有目标，1- 人，2- 车

byRes2

保留，置为 0

5.153 NET_VCA_VIOLENT_MOTION:剧烈运动参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_VIOLENT_MOTION extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDuration;
    public byte                 bySensitivity;
    public byte                 byMode;
    public byte[]               byRes = new byte[4];

    public NET_VCA_VIOLENT_MOTION(Pointer p) {
        super(p);
        // TODO Auto-generated constructor stub
    }
    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("struRegion", "wDuration", "bySensitivity", "byMode", "byRes");
    }
}
```

Members

struRegion

区域范围

wDuration

触发事件阈值，1-50 秒

bySensitivity

灵敏度参区域范围数，取值范围：[1,100]

byMode

0-纯视频模式，1-音视频联合模式，2-纯音频模式

byRes

保留

5.154 NET_VCA_YARD_TARRY:放风场滞留参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class NET_VCA_YARD_TARRY extends Structure
{
    public NET\_VCA\_POLYGON    struRegion = new NET_VCA_POLYGON();
    public short                wDelay;
    public byte[]               byRes = new byte[6];
}
```

```

public NET_VCA_YARD_TARRY(Pointer p) {
    super(p);
    // TODO Auto-generated constructor stub
}
@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("struRegion", "wDelay", "byRes");
}
}

```

Members

struRegion

区域范围

wDelay

放风场滞留时间，单位：秒，取值范围：[1,120]

byRes

保留

5.155 **struAlarmChannel**:报警通道参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class struAlarmChannel extends Structure{
    public int dwAlarmChanNum;

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("dwAlarmChanNum");
    }
}

```

Members

dwAlarmChanNum

发生报警通道数据个数，用于从 [NET DVR ALARMINFO V40](#) 的 `pAlarmData`(变长数据部分)计算出所有发生的报警通道号，四字节表示一个报警通道

5.156 **struAlarmHardDisk**:报警硬盘参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class struAlarmHardDisk extends Structure
{
    public int dwAlarmHardDiskNum;
}

```



```

@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("dwAlarmHardDiskNum");
}
}

```

Members

dwAlarmHardDiskNum

发生报警的硬盘个数，用于从 [NET DVR ALARMINFO V40](#) 的 pAlarmData(变长数据部分)计算出所有发生报警的硬盘号，四节表示一个硬盘

5.157 **struIOAlarm**:报警输入口信息

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class struIOAlarm extends Structure
```

```

{
    public int    dwAlarmInputNo;
    public int    dwTrigerAlarmOutNum;
    public int    dwTrigerRecordChanNum;

    @Override
    protected List<String> getFieldOrder() {
        //字段顺序必须跟 C++结构体参数顺序一致
        return Arrays.asList("dwAlarmInputNo","dwTrigerAlarmOutNum","dwTrigerRecordChanNum");
    }
}

```

Members

dwAlarmInputNo

发生报警的报警输入通道号，一次只有一个

dwTrigerAlarmOutNum

触发的报警输出个数，用于从 [NET DVR ALARMINFO V40](#) 的 pAlarmData(变长数据部分)计算出所有触发的报警输出通道号，四字节表示一个报警输出

dwTrigerRecordChanNum

触发的录像通道个数，用于从 [NET DVR ALARMINFO V40](#) 的 pAlarmData(变长数据部分)计算出所有触发的录像通道号，四字节表示一个通道

5.158 **struRecordingHost**:录播主机专用报警参数

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```
public static class struRecordingHost extends Structure
```

```

{
    public byte          bySubAlarmType;

```

```

public byte[]          byRes1 = new byte[3];
public NET\_DVR\_TIME\_EX struRecordEndTime = new NET_DVR_TIME_EX();
public byte[]          byRes = new byte[116];

@Override
protected List<String> getFieldOrder() {
    //字段顺序必须跟 C++结构体参数顺序一致
    return Arrays.asList("bySubAlarmType", "byRes1", "struRecordEndTime", "byRes");
}
}

```

Members

bySubAlarmType

报警子类型：1- 一键延迟录像

byRes1

保留，置为 0

struRecordEndTime

录播结束时间

byRes

保留，置为 0

5.159 **uStruAlarm**:报警信息联合体

In class `com.hcnetsdk.jna.HCNetSDKByJNA`

```

public static class uStruAlarm extends Union
{
    public byte[]          byUnionLen = new byte[128];
    public struIOAlarm      struioAlarm = new struIOAlarm();
    public struAlarmHardDisk strualarmHardDisk = new struAlarmHardDisk();
    public struAlarmChannel  strualarmChannel = new struAlarmChannel();
    public struRecordingHost strurecordingHost = new struRecordingHost();
}

```

Members

byUnionLen

联合体大小，128 字节

struioAlarm

报警输入/出口信息

strualarmHardDisk

报警硬盘参数

struAlarmChannel

报警通道参数

struRecordingHost

录播主机专用报警参数

5.160 WEP:wep 加密参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class WEP
{
    public int      dwAuthentication;
    public int      dwKeyLength;
    public int      dwKeyType;
    public int      dwActive;
    public byte[][] sKeyInfo = new
byte[HCNetSDK.WIFI_WEP_MAX_KEY_COUNT][HCNetSDK.WIFI_WEP_MAX_KEY_LENGTH];
}
```

Members

dwAuthentication

权限类型：0- 开放式，1- 共享式

dwKeyLength

密钥长度：0- 64 位，1- 128 位，2- 152 位

dwKeyType

密钥类型：0- 16 进制，1- ASCII

dwActive

激活哪个密钥（值有 0、1、2、3），0 表示激活第一个，以此类推

sKeyInfo

密钥信息

5.161 WPA_PSK:WPA_PSK 加密参数

In class `com.hikvision.netsdk.HCNetSDK`

```
public class WPA_PSK
{
    public int      dwKeyLength;
    public byte[]   sKeyInfo = new byte[HCNetSDK.WIFI_WPA_PSK_MAX_KEY_LENGTH];
    public byte     byEncryptType;
}
```

Members

dwKeyLength

字符加密的长度，允许 8-63 个 ASCII 字符

sKeyInfo

字符密钥的信息

byEncryptType

WPA-personal/WPA2-personal 模式下加密类型：0- AES，1- TKIP

5.162 WPA_WPA2:WPA_WPA2 加密参数

In class com.hikvision.netsdk.HCNetSDK

```
public class WPA_WPA2
{
    public byte        byEncryptType;
    public byte        byAuthType;
    public EAP TTLS   strueapTtls = new EAP_TTLS();
    public EAP PEAP   strueapPeap = new EAP_PEAP();
    public EAP TLS    strueapTls = new EAP_TLS();
}
```

Members

byEncryptType

加密类型：0- AES，1- TKIP

sKeyInfo

认证类型：0- EAP_TTLS，1- EAP_PEAP，2- EAP_TLS

EAP_TTLS

EAP_TTLS 认证参数

EAP_PEAP

EAP_PEAP 认证参数

EAP_TLS

EAP_TLS 认证参数

5.163 国家编号说明

中文	English	地区编号
欧洲	Europe	100
安道尔	Andorra	101
奥地利	Austria	102
阿尔巴尼亚	Albania	103
爱尔兰	Ireland	104
爱沙尼亚	Estonia	105
冰岛	Iceland	106
白俄罗斯	Belarus	107
保加利亚	Bulgaria	108
波兰	Poland	109
波斯尼亚和黑塞哥维那	Bosnia	110
比利时	Belgium	111
德国	Germany	112
丹麦	Denmark	113
俄罗斯联邦	Russia	114

法国	France	115
芬兰	Finland	116
荷兰	Holland	117
捷克	Czech	118
克罗地亚	Croatia	119
拉脱维亚	Latvia	120
立陶宛	Lithuania	121
列支敦士登	Liechtenstein	122
罗马尼亚	Romania	123
马其顿	Macedonia	124
马耳他	Malta	125
卢森堡	Luxembourg	126
摩纳哥	Monaco	127
摩尔多瓦	Moldova	128
挪威	Norway	129
塞尔维亚和黑山共和国	Serbia	130
葡萄牙	Portugal	131
瑞典	Sweden	132
瑞士	Switzerland	133
斯洛伐克	Slovak	134
斯洛文尼亚	Slovenia	135
圣马力诺	San marino	136
乌克兰	Ukraine	137
西班牙	Spain	138
希腊	Greece	139
匈牙利	Hungary	140
意大利	Italy	141
英国	United Kingdom	142
欧洲其他	Europe Other	143
亚洲国家列表	Asia	200
阿富汗	Afghanistan	201
阿拉伯联合酋长国	United Arab Emirates	202
阿曼	Oman	203
阿塞拜疆共和国	Azerbaijan	204
巴基斯坦	Pakistan	205
巴勒斯坦	Palestine	206
巴林	Bahrain	207
不丹	Bhutan	208
朝鲜	North Korea	209
东帝汶	Timor	210
菲律宾	Philippines	211
格鲁吉亚	Georgia	212

哈萨克斯坦	Kazakhstan	213
韩国	Korea	214
吉尔吉斯共和国	Kirgizstan	215
柬埔寨	Cambodia	216
卡塔尔	Qatar	217
科威特	Kuwait	218
老挝	Laos	219
黎巴嫩	Lebanon	220
马尔代夫	Maldives	221
马来西亚	Malaysia	222
蒙古	Mongolia	223
孟加拉国	Bangladesh	224
缅甸	Myanmar	225
尼泊尔	Nepal	226
日本	Japan	227
塞浦路斯	Cyprus	228
沙特阿拉伯	Saudi Arabia	229
斯里兰卡	Srilanka	230
塔吉克斯坦	Tajikistan	231
泰国	Thailand	232
土耳其	Turkey	233
土库曼斯坦	Turkmenistan	234
文莱	Brunei	235
乌兹别克斯坦	Uzbekistan	236
新加坡	Singapore	237
叙利亚	Syria	238
亚美尼亚共和国	Armenia	239
也门	Yemen	240
伊朗	Iran	241
伊拉克	Iraq	242
以色列	Israel	243
印度	India	244
印度尼西亚	Indonesia	245
约旦	Jordan	246
越南	Vietnam	247
中国	China	248
亚洲其他	Asia Other	249
美洲国家列表	America	300
阿根廷	Argentina	301
安提瓜和巴布达	Antigua and Barbuda	302
巴巴多斯	Barbados	303
玻利维亚	Bolivia	304

巴西	Brazil	305
多米尼克	Dominica	306
厄瓜多尔	Ecuador	307
古巴共和国	Cuba	308
哥伦比亚	Colombia	309
格林纳达	Grenada	310
圭亚那	Guyana	311
加拿大	Canada	312
秘鲁	Peru	313
美国	United States	314
墨西哥	Mexico	315
苏里南	Surinam	316
圣卢西亚	Saint-Lucia	317
特立尼达和多巴哥	Trinidad and Tobago	318
乌拉圭	Uruguay	319
委内瑞拉	Venezuela	320
牙买加	Jamaica	321
智利	Chile	322
巴哈马	Bahamas	323
美洲其他	America Other	324
非洲国家列表	Africa	400
阿尔及利亚	Algeria	401
埃及	Egypt	402
埃塞俄比亚	Ethiopia	403
安哥拉	Angola	404
贝宁	Benin	405
博茨瓦纳	Botswana	406
布基纳法索	Burkina Faso	407
布隆迪	Burundi	408
赤道几内亚	Equatorial Guinea	409
多哥	Togo	410
厄立特里亚	Eritrea	411
佛得角	Verde	412
冈比亚	Gambia	413
刚果（布）	Congo	414
刚果（金）	Congo-Kinshasa	415
吉布提	Djibouti	416
几内亚	Guinea	417
几内亚比绍	Guinea-Bissau	418
加蓬	Gabon	419
加纳	Ghana	420
津巴布韦	Zimbabwe	421

喀麦隆	Cameroon	422
科摩罗	Comoros	423
科特迪瓦	Cote d'Ivoire	424
肯尼亚	Kenya	425
莱索托	Lesotho	426
利比里亚	Liberia	427
利比亚	Libya	428
卢旺达	Rwanda	429
马达加斯加	Madagascar	430
马里	Mali	431
毛里求斯	Mauritius	432
毛里塔尼亚	Mauritania	433
摩洛哥	Morocco	434
莫桑比克	Mozambique	435
纳米比亚	Namibia	436
南非	South Africa	437
尼日尔	Niger	438
尼日利亚	Nigeria	439
塞拉利昂	Sierra Leone	440
塞内加尔	Senegal	441
塞舌尔	Seychelles	442
圣多美和普林西比	Sao Tome and Principe	443
苏丹	Sudan	444
索马里	Somali	445
坦桑尼亚	Tanzania	446
突尼斯	Tunisia	447
乌干达	Uganda	448
赞比亚	Zambia	449
乍得	Chad	450
中非共和国	Central African Republic	451
非洲其他	Africa Other	452
大洋洲国家列表	Oceania	500
澳大利亚	Australia	501
巴布亚新几内亚	Papua New Guinea	502
斐济	Fiji	503
库克群岛	Cook Islands	504
美属萨摩亚	Samoa	505
密克罗尼西亚联邦	Micronesia	506
瑙鲁	Nauru	507
汤加	Tonga	508
瓦努阿图	Vanuatu	509
新西兰	New Zealand	510

大洋洲其他	Oceania Other	511
-------	---------------	-----