

设备(热成像)

网络 SDK 编程指南

V5.2

声 明

非常感谢您购买我公司的产品，如果您有什么疑问或需要请随时联系我们。

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

目 录

声 明	I
目 录	II
1 SDK 简介	1
2 SDK 版本更新	4
3 函数调用顺序	6
3.1 实时预览模块流程	6
3.2 远程回放下载模块流程	7
3.3 语音对讲转发模块流程	10
3.4 动态火点搜索功能流程	12
3.5 报警监听模块流程	14
3.6 透明通道模块流程	15
4 函数调用实例	16
4.1 预览模块的示例代码	16
4.2 回放和下载模块的示例代码	21
4.3 语音对讲转发模块的示例代码	30
4.4 动态火点搜索功能的示例代码	31
4.5 报警监听模块的示例代码	34
4.6 透明通道模块的示例代码	37
5 函数说明	40
5.1 SDK 初始化	40
5.1.1 初始化 SDK NET_DVR_Init	40
5.1.2 释放 SDK 资源 NET_DVR_Cleanup	40
5.2 SDK 本地功能	40
SDK 本地参数配置	40
5.2.1 获取 SDK 本地参数 NET_DVR_GetSDKLocalCfg	40
5.2.2 设置 SDK 本地参数 NET_DVR_SetSDKLocalCfg	41
连接和接收超时时间及重连设置	42
5.2.3 设置网络连接超时时间和连接尝试次数 NET_DVR_SetConnectTime	42
5.2.4 设置重连功能 NET_DVR_SetReconnect	42
5.2.5 设置接收超时时间 NET_DVR_SetRecvTimeOut	42
多网卡绑定	42
5.2.6 获取所有 IP，用于支持多网卡接口 NET_DVR_GetLocalIP	42
5.2.7 设置 IP 绑定 NET_DVR_SetValidIP	43
SDK 版本、状态和能力	43
5.2.8 获取 SDK 的版本号和 build 信息 NET_DVR_GetSDKBuildVersion	43
5.2.9 获取当前 SDK 的状态信息 NET_DVR_GetSDKState	43
5.2.10 获取当前 SDK 的功能信息 NET_DVR_GetSDKAbility	43
SDK 启用写日志	44
5.2.11 启用写日志文件 NET_DVR_SetLogToFile	44
异常消息回调	44
5.2.12 注册异常消息回调函数 NET_DVR_SetExceptionCallBack_V30	44

获取错误信息	46
5.2.13 返回最后操作的错误码 NET_DVR_GetLastError.....	46
5.2.14 返回最后操作的错误码信息 NET_DVR_GetErrorMsg	47
5.3 用户注册	47
5.3.1 激活设备 NET_DVR_ActivateDevice	47
5.3.2 IPServer 或者 DDNS 域名解析, 获取动态 IP 地址和端口号 NET_DVR_GetDVRIPByResolveSvr_EX.....	47
5.3.3 用户注册设备 NET_DVR_Login_V40.....	48
5.3.4 用户注销 NET_DVR_Logout.....	48
5.4 获取设备能力集	48
5.4.1 获取设备能力集 NET_DVR_GetDeviceAbility	48
5.4.2 获取设备能力集 NET_DVR_GetSTDAbility.....	50
5.5 实时预览	50
5.5.1 实时预览 NET_DVR_RealPlay_V40	50
5.5.2 停止预览 NET_DVR_StopRealPlay	51
5.5.3 获取预览时用来解码和显示的播放库句柄 NET_DVR_GetRealPlayerIndex.....	51
5.6 强制 I 帧和刷新帧	52
5.6.1 强制 I 帧 NET_DVR_RemoteControl	52
5.7 预览显示视频参数配置	52
5.7.1 获取预览视频显示参数 NET_DVR_ClientGetVideoEffect	52
5.7.2 获取预览视频显示参数 NET_DVR_GetVideoEffect	53
5.7.3 设置预览视频显示参数 NET_DVR_ClientSetVideoEffect.....	53
5.7.4 设置预览视频显示参数 NET_DVR_SetVideoEffect	53
5.8 预览画面叠加字符和图像	54
5.8.1 预览画面叠加字符和图像, Linux 下无此接口 NET_DVR_RigisterDrawFun.....	54
5.9 预览时播放声音控制	54
5.9.1 设置声音播放模式 NET_DVR_SetAudioMode.....	54
5.9.2 独占声卡模式下开启声音 NET_DVR_OpenSound	54
5.9.3 独占声卡模式下开启声音 NET_DVR_CloseSound	55
5.9.4 共享声卡模式下开启声音 NET_DVR_OpenSoundShare.....	55
5.9.5 共享声卡模式下关闭声音 NET_DVR_CloseSoundShare	55
5.9.6 调节播放音量 NET_DVR_Volume	55
5.10 实时数据回调和录像	56
5.10.1 注册回调函数, 捕获实时码流数据 NET_DVR_SetRealDataCallBack	56
5.10.2 注册回调函数, 捕获实时码流数据 (标准码流) NET_DVR_SetStandardDataCallBack ...	56
5.10.3 捕获数据并保存到指定的文件中 NET_DVR_SaveRealData	57
5.10.4 停止数据捕获 NET_DVR_StopSaveRealData.....	57
5.11 预览抓图	58
5.11.1 设置抓图模式 NET_DVR_SetCapturePictureMode.....	58
5.11.2 预览时, 单帧数据捕获并保存成图片 NET_DVR_CapturePicture.....	58
5.12 设备抓图	58
5.12.1 单帧数据捕获并保存成 JPEG 图片 NET_DVR_CaptureJPEGPicture	58
5.12.2 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.....	59

5.13	云台控制	59
	云台控制操作	59
5.13.1	云台控制操作（需先启动图像预览）NET_DVR_PTZControl	59
5.13.2	云台控制操作（不用启动图像预览）NET_DVR_PTZControl_Other	61
5.13.3	带速度的云台控制操作（需先启动图像预览）NET_DVR_PTZControlWithSpeed	61
5.13.4	带速度的云台控制操作（不用启动图像预览）NET_DVR_PTZControlWithSpeed_Other	61
	云台预置点操作	62
5.13.5	云台预置点操作，需先启动预览 NET_DVR_PTZPreset	62
5.13.6	云台预置点操作 NET_DVR_PTZPreset_Other	62
	云台巡航操作	63
5.13.7	云台巡航操作，需先启动预览 NET_DVR_PTZPCruise	63
5.13.8	云台巡航操作 NET_DVR_PTZPCruise_Other	64
5.13.9	批量巡航路径 NET_DVR_GetDeviceConfig	64
	云台轨迹操作	65
5.13.10	云台轨迹操作，需先启动预览 NET_DVR_PTZTrack	65
5.13.11	云台轨迹操作 NET_DVR_PTZTrack_Other	65
	透明云台控制	66
5.13.12	透明云台操作，需先启动预览 NET_DVR_TransPTZ	66
5.13.13	透明云台操作 NET_DVR_TransPTZ_Other	66
	云台区域缩放控制	66
5.13.14	云台图象区域选择放大或缩小 NET_DVR_PTZSelZoomIn	66
5.13.15	云台图像区域选择放大或缩小 NET_DVR_PTZSelZoomIn_Ex	67
5.14	报警布撤防	67
	设置报警等信息上传的回调函数	67
5.14.1	注册报警消息回调函数 NET_DVR_SetDVRMessageCallBack_V31	67
	布防撤防	68
5.14.2	建立报警上传通道，获取报警等信息 NET_DVR_SetupAlarmChan_V41	68
5.14.3	撤销报警上传通道 NET_DVR_CloseAlarmChan_V30	68
5.15	监听报警	68
5.15.1	启动监听，接收设备主动上传的报警等信息 NET_DVR_StartListen_V30	68
5.15.2	停止监听（支持多线程）NET_DVR_StopListen_V30	69
5.16	远程参数配置	70
	系统参数配置	70
5.16.1	获取设备参数 NET_DVR_GetDVRConfig	70
5.16.2	设置设备参数 NET_DVR_SetDVRConfig	71
	通道参数配置	71
5.16.3	获取通道参数 NET_DVR_GetDVRConfig	71
5.16.4	设置通道参数 NET_DVR_SetDVRConfig	72
5.16.5	批量获取配置信息 NET_DVR_GetDeviceConfig	73
5.16.6	批量设置配置信息 NET_DVR_SetDeviceConfig	74
	网络参数配置	75
5.16.7	获取网络参数 NET_DVR_GetDVRConfig	75
5.16.8	设置网络参数 NET_DVR_SetDVRConfig	75
5.16.9	批量获取配置信息 NET_DVR_GetDeviceConfig	76

5.16.10	批量设置配置信息 NET_DVR_SetDeviceConfig.....	77
	Smart 参数配置.....	78
5.16.11	批量获取配置信息 NET_DVR_GetDeviceConfig.....	78
5.16.12	批量设置配置信息 NET_DVR_SetDeviceConfig.....	79
	报警输入输出配置	80
5.16.13	获取设备参数 NET_DVR_GetDVRConfig.....	80
5.16.14	设置设备参数 NET_DVR_SetDVRConfig.....	80
5.16.15	获取设备报警输出 NET_DVR_GetAlarmOut_V30	81
5.16.16	设置设备报警输出 NET_DVR_SetAlarmOut	81
	RTSP 协议参数配置	82
5.16.17	获取 RTSP 协议参数 NET_DVR_GetRtspConfig.....	82
5.16.18	设置 RTSP 协议参数 NET_DVR_SetRtspConfig	82
5.17	热成像相关功能	82
	热成像参数配置	82
5.17.1	获取设备的配置信息 NET_DVR_GetSTDConfig.....	82
5.17.2	获取设备的配置信息 NET_DVR_SetSTDConfig	83
	远程控制	84
5.17.3	远程控制 NET_DVR_STDControl.....	84
	手动快门补偿	84
5.17.4	热成像仪手动快门补偿 NET_DVR_ShutterCompensation.....	84
	坏点校正	84
5.17.5	热成像仪坏点校正 NET_DVR_CorrectDeadPixel	84
	辅助聚焦控制	85
5.17.6	控制一键聚焦 NET_DVR_FocusOnePush	85
	恢复镜头电机默认位置	85
5.17.7	恢复镜头电机默认位置 NET_DVR_ResetLens.....	85
5.18	录像文件回放、下载、锁定及备份.....	85
	录像文件的查找	85
5.18.1	根据文件类型、时间查找设备录像文件 NET_DVR_FindFile_V40.....	85
5.18.2	逐个获取查找到的文件信息 NET_DVR_FindNextFile_V40	86
5.18.3	关闭文件查找，释放资源 NET_DVR_FindClose_V30.....	86
	回放录像文件	86
5.18.4	按文件名回放录像文件 NET_DVR_PlayBackByName	86
5.18.5	按时间回放录像文件 NET_DVR_PlayBackByTime_V40	87
5.18.6	控制录像回放的状态 NET_DVR_PlayBackControl_V40	88
5.18.7	停止回放录像文件 NET_DVR_StopPlayBack	89
	回放录像文件时的数据捕获	90
5.18.8	捕获回放的录像数据，并保存成文件 NET_DVR_PlayBackSaveData	90
5.18.9	停止保存录像数据 NET_DVR_StopPlayBackSave	90
5.18.10	注册回调函数，捕获录像数据 NET_DVR_SetPlayDataCallBack_V40.....	90
	回放的其他操作	91
5.18.11	获取录像回放时显示的 OSD 时间 NET_DVR_GetPlayBackOsdTime.....	91
5.18.12	录像回放时抓图，并保存在文件中 NET_DVR_PlayBackCaptureFile	91
5.18.13	刷新显示回放窗口 NET_DVR_RefreshPlay	91

5.18.14	获取回放时用来解码显示的播放库句柄 NET_DVR_GetPlayBackPlayerIndex	92
	下载录像文件	92
5.18.15	按文件名下载录像文件 NET_DVR_GetFileByName.....	92
5.18.16	按时间下载录像文件 NET_DVR_GetFileByTime_V40.....	92
5.18.17	控制录像下载的状态 NET_DVR_PlayBackControl_V40	93
5.18.18	停止下载录像文件 NET_DVR_StopGetFile.....	94
5.18.19	获取当前下载录像文件的进度 NET_DVR_GetDownloadPos	94
	录像文件锁定和解锁	95
5.18.20	按文件名锁定录像文件 NET_DVR_LockFileByName	95
5.18.21	按文件名解锁录像文件 NET_DVR_UnlockFileByName	95
5.19	图片的查找、回放下载	95
	查找图片	95
5.19.1	根据类型和时间查找图片 NET_DVR_FindPicture	95
5.19.2	逐个获取查找到的图片 NET_DVR_FindNextPicture_V40.....	95
5.19.3	关闭图片查找，释放资源 NET_DVR_CloseFindPicture.....	96
	回放（下载）图片	96
5.19.4	图片回放 NET_DVR_GetPicture_V30	96
5.20	语音对讲、转发及广播	97
	语音对讲(Windows 32 位系统支持).....	97
5.20.1	启动语音对讲 NET_DVR_StartVoiceCom_V30.....	97
5.20.2	设置语音对讲客户端的音量 NET_DVR_SetVoiceComClientVolume	98
5.20.3	停止语音对讲或者语音转发 NET_DVR_StopVoiceCom	98
	语音转发	98
5.20.4	启动语音转发，获取编码后的音频数据 NET_DVR_StartVoiceCom_MR_V30	98
5.20.5	转发语音数据 NET_DVR_VoiceComSendData	99
5.20.6	停止语音对讲或语音转发 NET_DVR_StopVoiceCom	100
	语音广播(Windows 32 位系统支持).....	100
5.20.7	启动语音广播的 PC 端声音捕获 NET_DVR_ClientAudioStart_V30.....	100
5.20.8	添加设备的某个语音通道到可以接收 PC 端声音的广播组 NET_DVR_AddDVR_V30.....	100
5.20.9	从可接收 PC 机声音的广播组里删除该设备的语音通道 NET_DVR_DeIDVR_V30	101
5.20.10	停止语音广播的 PC 端声音捕获 NET_DVR_ClientAudioStop.....	101
	音频压缩参数	101
5.20.11	获取当前生效的对讲音频压缩参数 NET_DVR_GetCurrentAudioCompress	101
	音频编解码(Windows 32 位系统支持).....	102
	G722 音频编解码.....	102
5.20.12	初始化音频编码 NET_DVR_InitG722Encoder.....	102
5.20.13	G722 音频编码 NET_DVR_EncodeG722Frame.....	102
5.20.14	释放音频编码资源 NET_DVR_ReleaseG722Encoder.....	102
5.20.15	初始化音频解码 NET_DVR_InitG722Decoder	102
5.20.16	G722 音频解码 NET_DVR_DecodeG722Frame	103
5.20.17	释放音频解码资源 NET_DVR_ReleaseG722Decoder	103
	G711 音频编解码.....	103
5.20.18	G711 音频编码 NET_DVR_EncodeG711Frame	103
5.20.19	G711 音频解码 NET_DVR_DecodeG711Frame	104

G726 音频编解码	104
5.20.20 初始化音频编码 NET_DVR_InitG726Encoder	104
5.20.21 G726 音频编码 NET_DVR_EncodeG726Frame	104
5.20.22 释放音频编码资源 NET_DVR_ReleaseG726Encoder	105
5.20.23 初始化音频解码 NET_DVR_InitG726Decoder	105
5.20.24 G726 音频解码 NET_DVR_DecodeG726Frame	105
5.20.25 释放音频解码资源 NET_DVR_ReleaseG726Decoder	105
5.21 数据透传	106
透明通道	106
5.21.1 建立透明通道 NET_DVR_SerialStart	106
5.21.2 通过透明通道向设备串口发送数据 NET_DVR_SerialSend	106
5.21.3 断开透明通道 NET_DVR_SerialStop	106
向串口发送数据	107
5.21.4 直接向串口发送数据，不需要建立透明通道 NET_DVR_SendToSerialPort	107
5.21.5 直接向 232 串口发送数据，不需要建立透明通道 NET_DVR_SendTo232Port	107
5.22 存储管理	107
5.22.1 远程格式化设备硬盘 NET_DVR_FormatDisk	107
5.22.2 获取格式化硬盘的进度 NET_DVR_GetFormatProgress	107
5.22.3 关闭格式化硬盘句柄，释放资源 NET_DVR_CloseFormatHandle	108
5.23 设备手动录像	108
5.23.1 远程手动启动设备录像 NET_DVR_StartDVRRecord	108
5.23.2 远程手动停止设备录像 NET_DVR_StopDVRRecord	109
5.24 服务器测试	109
服务器测试	109
5.24.1 启动长连接远程配置 NET_DVR_StartRemoteConfig	109
5.24.2 获取长连接配置的状态 NET_DVR_GetRemoteConfigState	110
5.24.3 关闭长连接配置接口所创建的句柄，释放资源 NET_DVR_StopRemoteConfig	110
Email 测试	110
5.24.4 测试按已配置的 EMAIL 参数能否收发成功 NET_DVR_StartEmailTest	110
5.24.5 获取邮件测试的进度 NET_DVR_GetEmailTestProgress	111
5.24.6 停止邮件测试 NET_DVR_StopEmailTest	111
5.25 设备维护管理	111
获取设备工作状态	111
5.25.1 获取设备的工作状态 NET_DVR_GetDVRWorkState_V30	111
5.25.2 设备在线状态检测 NET_DVR_RemoteControl	112
5.25.3 启动设备状态巡检 NET_DVR_StartGetDevState	112
5.25.4 停止设备状态巡检 NET_DVR_StopGetDevState	112
远程升级	113
5.25.5 设置远程升级时网络环境 NET_DVR_SetNetworkEnvironment	113
5.25.6 远程升级 NET_DVR_Upgrade	113
5.25.7 获取远程升级的进度 NET_DVR_GetUpgradeProgress	113
5.25.8 获取远程升级的状态 NET_DVR_GetUpgradeState	113
5.25.9 获取远程升级的阶段信息 NET_DVR_GetUpgradeStep	114
5.25.10 关闭远程升级句柄，释放资源 NET_DVR_CloseUpgradeHandle	114

日志查找	114
5.25.11 查找设备的日志信息 NET_DVR_FindDVRLog_V30	114
5.25.12 逐条获取查找到的日志信息 NET_DVR_FindNextLog_V30	121
5.25.13 释放查找日志的资源 NET_DVR_FindLogClose_V30	122
恢复设备默认参数	122
5.25.14 恢复设备默认参数 NET_DVR_RestoreConfig	122
5.25.15 完全恢复出厂默认参数 NET_DVR_RemoteControl	122
导入/导出配置文件	123
5.25.16 导出配置文件 NET_DVR_GetConfigFile_V30	123
5.25.17 导出配置文件 NET_DVR_GetConfigFile	123
5.25.18 导入配置文件 NET_DVR_SetConfigFile_EX	123
5.25.19 导入配置文件 NET_DVR_SetConfigFile	124
关机和重启	124
5.25.20 重启设备 NET_DVR_RebootDVR	124
5.25.21 关闭设备 NET_DVR_ShutDownDVR	124
6 错误代码及说明	125
6.1 网络通讯库错误码	125
6.2 RTSP 通讯库错误码	128
6.3 软解码库错误码	129
6.4 语音对讲库错误码	130

1 SDK 简介

设备网络 SDK 是基于设备私有网络通信协议开发的，为嵌入式网络硬盘录像机、NVR、视频服务器、网络摄像机、网络球机、解码器、报警主机等网络产品服务的配套模块，用于远程访问和控制设备软件的二次开发。

本文档仅介绍热成像相关功能接口，更多其他功能接口请参考《设备网络 SDK 使用手册.chm》。

适用于但不仅限于以下产品型号：

DS-2TD2035T-XXX(HZXXX)、DS-2TD2035-XXX(HZXXX)、DS-2TD2235D-XXX、DS-2TD4035D-XXX、DS-2TD6035DT-25(40)、DS-2TD8035-XXX、DS-2TD8065T-XXX 等。

设备网络 SDK 主要功能：

主要用于实时码流预览、录像文件回放和下载、云台控制、布防/撤防、语音对讲、日志管理、远程升级、格式化硬盘（SD 卡）、参数配置（系统配置、通道配置、串口配置、报警配置、用户配置）和获取设备能力集等。

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

表 1.1 Windows SDK 组件

网络通讯库	外部接口	HCNetSDK.h	头文件	
		HCNetSDK.lib	LIB 库文件	
		HCNetSDK.dll	DLL 库文件	
	核心组件	HCCore.lib	LIB 库文件	
		HCCore.dll	DLL 库文件	
组件库	设备配置核心组件	HCCoreDevCfg.dll	DLL 库文件	HCNetSDKCom 文件夹
	预览组件	HCPreview.lib	LIB 库文件	HCNetSDKCom 文件夹
		HCPreview.dll	DLL 库文件	HCNetSDKCom 文件夹
	回放组件	HCPlayBack.dll	DLL 库文件	HCNetSDKCom 文件夹
	语音组件	HCVoiceTalk.dll	DLL 库文件	HCNetSDKCom 文件夹
	报警组件	HCArm.lib	LIB 库文件	HCNetSDKCom 文件夹
		HCArm.dll	DLL 库文件	HCNetSDKCom 文件夹
	显示组件	HCDisplay.dll	DLL 库文件	HCNetSDKCom 文件夹
	行业应用管理配置组件	HCIndustry.dll	DLL 库文件	HCNetSDKCom 文件夹
	维护管理配置组件	HCGeneralCfgMgr.lib	LIB 库文件	HCNetSDKCom 文件夹
		HCGeneralCfgMgr.dll	DLL 库文件	HCNetSDKCom 文件夹
RTSP 通讯库		StreamTransClient.dll	DLL 库文件	HCNetSDKCom 文件夹

转封装库		SystemTransform.dll	DLL 库文件	HCNetSDKCom 文件夹
语音对讲库		AudioIntercom.dll	DLL 库文件	HCNetSDKCom 文件夹
		OpenAL32.dll	DLL 库文件	HCNetSDKCom 文件夹
字符转码库		libiconv2.dll	DLL 库文件	HCNetSDKCom 文件夹
模拟能力集		LocalXml.zip	XML 文件包	
软解码库		plaympeg4.h	头文件	
		PlayCtrl.lib	LIB 库文件	
		PlayCtrl.dll	DLL 库文件	
		AudioRender.dll	DLL 库文件	
		SuperRender.dll	DLL 库文件	
硬解码库		DsSdk.dll	DLL 库文件	HCNetSDKCom 文件夹

表 1.2 Linux SDK 组件

网络通讯库	外部接口	HCNetSDK.h	头文件	
		libhcnetsdk.so	SO 库文件	
	核心组件	libHCCore.so	SO 库文件	
组件库	设备配置核心组件	libHCCoreDevCfg.so	SO 库文件	HCNetSDKCom 文件夹
	预览组件	libHCPreview.so	SO 库文件	HCNetSDKCom 文件夹
	回放组件	libHCPlayBack.so	SO 库文件	HCNetSDKCom 文件夹
	语音组件	libHCVoiceTalk.so	SO 库文件	HCNetSDKCom 文件夹
	报警组件	libHCAAlarm.so	SO 库文件	HCNetSDKCom 文件夹
	显示组件	libHCDisplay.so	SO 库文件	HCNetSDKCom 文件夹
	行业应用管理配置组件	libHCIndustry.so	SO 库文件	HCNetSDKCom 文件夹
	维护管理配置组件	libHCGeneralCfgMgr.so	SO 库文件	HCNetSDKCom 文件夹
hpr 库		libhpr.so	SO 库文件	
RTSP 通讯库		libStreamTransClient.so	SO 库文件	HCNetSDKCom 文件夹
转封装库		libSystemTransform.so	SO 库文件	HCNetSDKCom 文件夹
字符转码库		libiconv2.so	SO 库文件	HCNetSDKCom 文件夹
软解码库		LinuxPlayM4.h	头文件	
		PlayM4.h	头文件	
		libMPCtrl.so	SO 库文件	
		libPlayCtrl.so	SO 库文件	

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

组件在 SDK 中的作用和使用条件分别说明。

- **网络通讯库**：设备网络 SDK 的主体，主要用于网络客户端与各类产品之间的通讯交互，负责远程功能调控，远程参数配置及码流数据的获取和处理等。设备网络 SDK V5.0 针对产品应用业务进行细化，对之前版本的 SDK 的功能模块进行组件化，其中外部接口（HCNetSDK.dll）仍然保持和设备网络 SDK V4.x 版本保存一致(向下兼容)，其他单独的业务功能（预览、回放等）可以加载单独的模块组件，多个业务功能也可以组合使用。**更新 SDK 时，HCNetSDK.dll、HCCore.dll 以及 HCNetSDKCom 文件夹下的功能组件库文件都需要更新加载，且 HCNetSDKCom 文件夹名不能修改。**
- **hpr 库**：网络通讯库的依赖库，Linux SDK 使用时和网络通讯库同时加载。
- **RTSP 通讯库**：支持 RTSP 传输协议的网络库。当需要对支持 RTSP 协议的产品进行取流等操作时必须加载该项组件。
- **转封装库**：库的功能可以分为两种：一种是将标准码流转换成采用我们公司封装格式的码流。当用户需要对支持 RTSP 协议的产品捕获采用本公司封装格式的码流数据时（即当设置 [NET_DVR_RealPlay_V40](#) 接口中的回调函数捕获数据或者调用 [NET_DVR_SetRealDataCallBack](#) 接口捕获数据时）必须加载该组件。另一种功能是将标准码流转换成其他格式的封装，如 3GPP、PS 等。例如，当用户需要对支持 RTSP 协议的产品实时捕获指定封装格式的码流数据（对应的 SDK 接口为 [NET_DVR_SaveRealData](#)）时必须加载该项组件。
- **语音对讲库**：用于语音对讲时通过声卡采集数据并按照指定的编码格式编码码流或者解码播放音频码流数据（不带封装格式的码流数据）。V4.2.2.5 及以前版本 SDK 均采用 windows API 实现相关功能。之后版本默认使用语音对讲库的方式，通过接口 [NET_DVR_SetSDKLocalCfg](#) 可以选择之前的 windows API 模式。OpenAL32.dll 为依赖库，语音对讲库模式下必须加载。**Windows64 位或者 Linux 系统下无语音对讲功能。**
- **字符转换库**：电脑字符集和设备字符集不一致时，SDK 内部需要进行字符编码转换，SDK 默认使用 libiconv 库进行类型转换。如果用户不想使用 libiconv 编码库，可以调用 [NET_DVR_SetSDKLocalCfg](#) (类型: [NET_SDK_LOCAL_CFG_TYPE_BYTE_ENCODE](#))设置字符转码回调函数，将用户自己的字符编码接口告知 SDK，然后 SDK 将使用用户提供的字符编码接口进行字符串处理。
- **模拟能力集**：如果需要获取设备能力集（[NET_DVR_GetDeviceAbility](#)），建议调用 [NET_DVR_SetSDKLocalCfg](#) 启用模拟能力集，此时需要加载 LocalXml.zip（要求和网络通讯库放在同一个目录下）。
- **软解码库**：主要用于对实时码流数据进行解码显示（实现预览功能）和对录像文件进行回放解码等。用户如果需要在 SDK 内部进行对实时流和录像码流播放显示时（即 [NET_DVR_RealPlay_V40](#) 接口的第二个结构体参数的播放句柄设置成有效句柄时）必须加载该组件，而如果用户仅需要用网络通讯库捕获到数据后再外部自行处理就不需要加载该组件，这种情况下用户在外部分自行解码将更灵活，可参见软解码库函数说明《播放器 SDK 编程指南》。**Linux 64 位系统不支持软解码功能，预览、回放等窗口句柄传空，仅支持只取流不解码。**
- **硬解码库**：需在配备硬解码卡（MD 卡）的前提下使用，通过解码卡的解码与输出功能实现实时流的解码显示及向监视器上矩阵输出的功能。**Windows64 位或者 Linux 系统下无该硬解码库。对于热成像设备设备，不需要使用该库文件。**

2 SDK 版本更新

Version 5.1.6.15 (build20160302)

- 热成像双光谱重载云台(DS-2TD8035T 等) V1.0
- 新增低功耗配置功能（对应接口：[NET_DVR_GetSTDConfig](#) 和 [NET_DVR_SetSTDConfig](#)）：
命令：NET_DVR_GET_LOWPOWER、NET_DVR_SET_LOWPOWER。
- 新增变倍联动配置功能（对应接口：[NET_DVR_GetSTDConfig](#) 和 [NET_DVR_SetSTDConfig](#)）：
命令：NET_DVR_GET_ZOOMLINKAGE、NET_DVR_SET_ZOOMLINKAGE。
- 新增手动测距功能（对应接口：[NET_DVR_SetSTDConfig](#)）：
命令：NET_DVR_SET_MANUALRANGING。
- 新增火点可见光镜头聚焦变倍功能（对应接口：[NET_DVR_STDCtrl](#)）：
命令：NET_DVR_FIRE_FOCUSZOOM_CTRL。
- 新增低功耗配置能力集（对应接口：[NET_DVR_GetSTDAbility](#)）：
能力集类型：NET_DVR_GET_LOWPOWER_CAPABILITIES。
- 新增变倍联动配置能力集（对应接口：[NET_DVR_GetSTDAbility](#)）：
能力集类型：NET_DVR_GET_ZOOMLINKAGE_CAPABILITIES。
- 新增手动测距配置能力集（对应接口：[NET_DVR_GetSTDAbility](#)）：
能力集类型：NET_DVR_GET_MANUALRANGING_CAPABILITIES。
- NET_DVR_FIND_PICTURE_PARAM(图片查找条件)和 NET_DVR_FIND_PICTURE_V40(图片查找结果)中
byFileType(图片类型)分别新增取值：0x1e-火点检测。
- NET_DVR_FILECOND_V40(录像文件查找条件)和 NET_DVR_FINDDATA_V40(录像文件查找结果)中
dwFileType(文件类型)分别新增取值：34-火点检测。
- NET_DVR_RECORDSCHD_V40(时间段录像参数)和 NET_DVR_RECORDDAY_V40(全天录像参数)中
byRecordType(录像类型)分别新增取值：30-火点检测。
- NET_DVR_CAMERAPARAMCFG_EX(前端参数配置)使用 8 个保留字节新增参数：struOpticalDehaze(光学透
雾参数)。
- NET_DVR_LASER_PARAM_CFG(激光参数配置)使用 3 个保留字节新增参数：byEnabled(手动控制激光灯使
能)、byIllumination(激光灯强度配置)、byLightAngle(补光角度)。
- NET_DVR_DEVSERVER_CFG(模块服务配置)使用 1 个保留字节新增参数：byEnableDeicing(除冰功能)。
- 火点检测配置能力集(FireDetection)新增节点：<fireFrameDis>(火点框显示)、<fireMaxTemp>(最高温度)、
<fireMaxTempPosition>(最高温度位置显示)、<fireDistance>(最高温度距离)。
- 热成像能力集(ThermalCap)新增节点：<isFireFocusZoomSupport>(是否支持火点可见光镜头聚焦变倍)。
- JPEG 抓图能力集(JpegCaptureAbility)扩展，<supportFileType>(支持图片类型)新增取值：fireDetection(火
点检测)。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中设备通道输入能力(ChannelInputAbility)扩展，<RecordPlan>
和<supportRecordType>中子节点<supportRecordType>(支持录像类型)分别新增取值：fireDetection(火点
检测)。
- 设备前端参数能力(IPC_FRONT_PARAMETER_V20)扩展，新增节点：<OpticalDehaze>(光学透雾能力)，
<LaserConfig>(激光参数配置能力)中新增子节点：<enable>(使能)、<illumination>(激光灯强度配置)、
<lightAngle>(补光角度)。
- 设备软硬件能力集(BasicCapability)扩展，节点<DevModuleServerCfg>(模块服务配置能力)中新增子节点：
<deicing>(除冰能力)。

Version 5.0.2.15 (build20141212)

- 火点检测 DS-2TD4035D-25、DS-2TD4035D-50
- 新增火点检测配置功能（对应接口：[NET_DVR_GetSTDConfig](#) 和 [NET_DVR_SetSTDConfig](#)）：
命令：NET_DVR_GET_FIREDETECTION、NET_DVR_SET_FIREDETECTION。
- 新增火点检测联动配置功能（对应接口：[NET_DVR_GetSTDConfig](#) 和 [NET_DVR_SetSTDConfig](#)）：
命令：NET_DVR_GET_FIREDETECTION_TRIGGER、NET_DVR_SET_FIREDETECTION_TRIGGER。
- 新增火点检测报警功能（对应接口：[NET_DVR_SetDVRMessageCallBack_V31](#)）：
COMM_FIREDETECTION_ALARM。
- 新增设备能力集（对应接口：[NET_DVR_GetSTDAbility](#)）：
能力集类型：NET_DVR_GET_THERMAL_CAPABILITIES(热成像能力)、
NET_DVR_GET_FIREDETECTION_CAPABILITIES(火点检测配置能力)。
- 新增日志类型（对应接口：[NET_DVR_FindDVRLog_V30](#)）：
报警日志次类型：0x48、0x49。

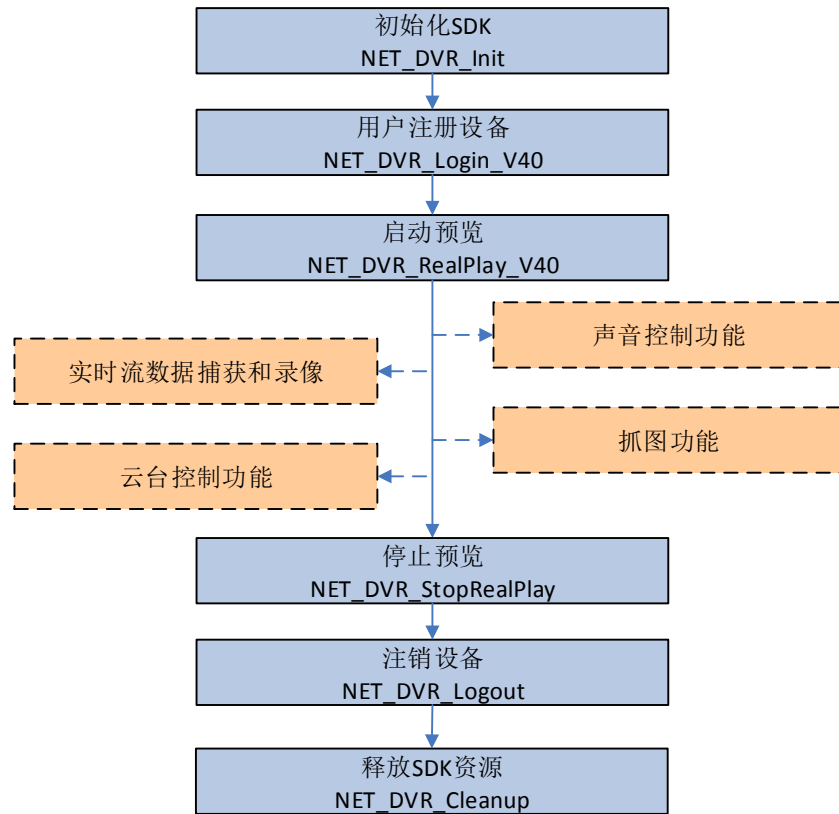
Version 4.1.5.3 (build20120528)

- TII_IPCAM（红外热成像摄像机）
- 新增热成像仪手动快门补偿接口：[NET_DVR_ShutterCompensation](#)。
- 新增热成像仪坏点校正接口：[NET_DVR_CorrectDeadPixel](#)。
- 新增 IPC 辅助聚焦控制接口：[NET_DVR_FocusOnePush](#)。
- 新增 IPC 恢复镜头电机默认位置接口：[NET_DVR_ResetLens](#)。
- 新增配置功能：NET_DVR_AUDIO_INPUT_PARAM、NET_DVR_CAMERA_DEHAZE_CFG、
NET_IPC_AUX_ALARMCFG。
- 新增遥控器控制接口：[NET_DVR_RemoteControl](#)。
- 新增报警上传消息类型：NET_IPC_AUXALARM_RESULT。
- 新增设备报警能力集：设备报警能力 XML 描述。
- 新增报警日志的次类型：MINOR_WIRELESS_ALARM_START、MINOR_WIRELESS_ALARM_STOP、
MINOR_PIR_ALARM_START、MINOR_PIR_ALARM_STOP、MINOR_CALLHELP_ALARM_START、
MINOR_CALLHELP_ALARM_STOP。
- NET_DVR_CAMERAPARAMCFG 使用 7 个保留字节增加参数 byDimmerMode、byPaletteMode、
byEnhancedMode、byFilterSwitch、byFocusSpeed、byAutoCompensationInterval 和 bySceneMode。
- NET_DVR_EXPOSURE 使用 1 个保留字节增加参数 byAutoApertureLevel。
- 设备前端参数能力集增加自动光圈灵敏度能力项。
- NET_DVR_COMPRESSION_INFO_V30 使用 1 个保留字节增加参数 byVideoEncComplexity。
- NET_DVR_HANDLEEXCEPTION_V30 参数 dwHandleType 增加定义 0x20(无线声光报警器联动)。
- NET_DVR_RECORDDAY 和 NET_DVR_RECORDSCHED 的参数 byRecordType，或者 NET_DVR_FILECOND 和
NET_DVR_FILECOND_V40 的参数 dwFileType 增加定义：10-PIR 报警，11-无线报警，12-呼救报警，13-PIR|
无线报警|呼救报警。
- NET_DVR_FINDDATA_V30 和 NET_DVR_FIND_PICTURE_PARAM 的参数 byFileType 增加定义：10-PIR 报警，
11-无线报警，12-呼救报警。
- NET_DVR_EVENT_CAPTURE 的参数 struRelCaptureChan 增加定义：数组 3 表示 PIR 报警抓图，数组 4 表
示无线报警抓图，数组 5 表示呼救报警抓图。

3 函数调用顺序

3.1 实时预览模块流程

图 3.1 实时预览模块流程



图中虚线框部分的模块是与预览模块相关，在启动预览后调用，这些模块之间是并列的关系，各自完成相应的功能。

热成像设备一般有两个通道，第一个通道为可见光通道，第二个通道为热成像通道，如图所示。

图 3.2 预览通道



实时流解码方式:

- 方式一：在预览接口 [NET DVR RealPlay V40](#) 中预览参数的播放窗口句柄赋成有效句柄，则由 SDK 实现解码功能。在初始化 SDK 和注册设备两步骤后，直接调用启动预览和停止预览接口。
- 方式二：用户可以通过设置预览接口 [NET DVR RealPlay V40](#) 中预览参数的播放窗口句柄为空值，并通过调用捕获数据的接口（即设置 [NET DVR RealPlay V40](#) 接口中的回调函数或调用 [NET DVR SetRealDataCallBack](#)、[NET DVR SetStandardDataCallBack](#) 接口），获取码流数据进行后续解码播放处理。

相关功能:

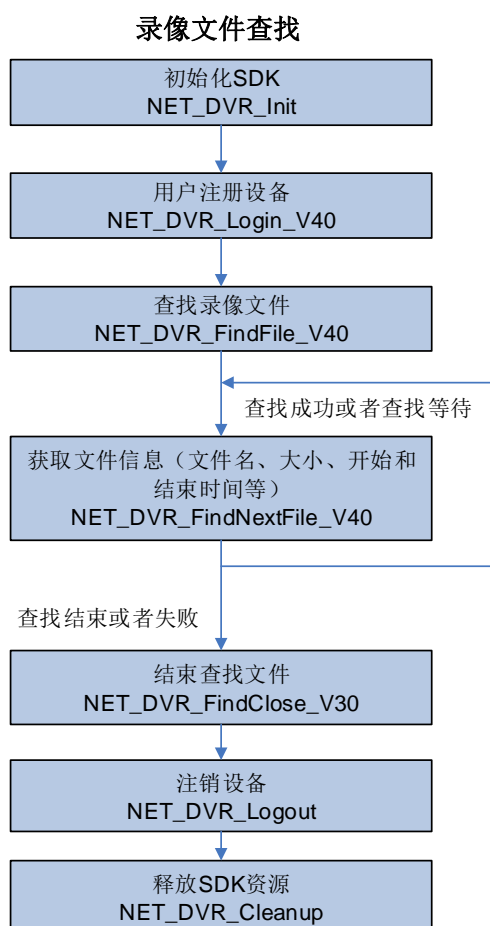
- 声音控制功能主要实现独占、共享声音的打开和关闭；音量的控制。相关接口有：[NET DVR OpenSound](#)、[NET DVR CloseSound](#)、[NET DVR OpenSoundShare](#)、[NET DVR CloseSoundShare](#)、[NET DVR Volume](#) 等。
- 实时流数据捕获和录像模块主要实现数据回调和本地录像的功能，可以供用户后续处理。相关接口有：[NET DVR SetRealDataCallBack](#)、[NET DVR SetStandardDataCallBack](#)、[NET DVR SaveRealData](#) 等。
- 抓图功能主要实现对当前解码图像的捕获，保存格式为 BMP。相关接口有：[NET DVR CapturePicture](#)。[NET DVR CaptureJPEGPicture](#) 支持登录后直接从设备抓取 JPEG 图片，需要设备支持。
- 云台控制模块主要是在开启预览的前提下实现对云台控制的操作功能，包括云台预置点、巡航、轨迹和透明云台等。相关接口有：[NET DVR PTZControl](#)、[NET DVR PTZPreset](#)、[NET DVR PTZCruise](#)、[NET DVR PTZTrack](#)、[NET DVR TransPTZ](#) 等。

[调用实例代码](#)

3.2 远程回放下载模块流程

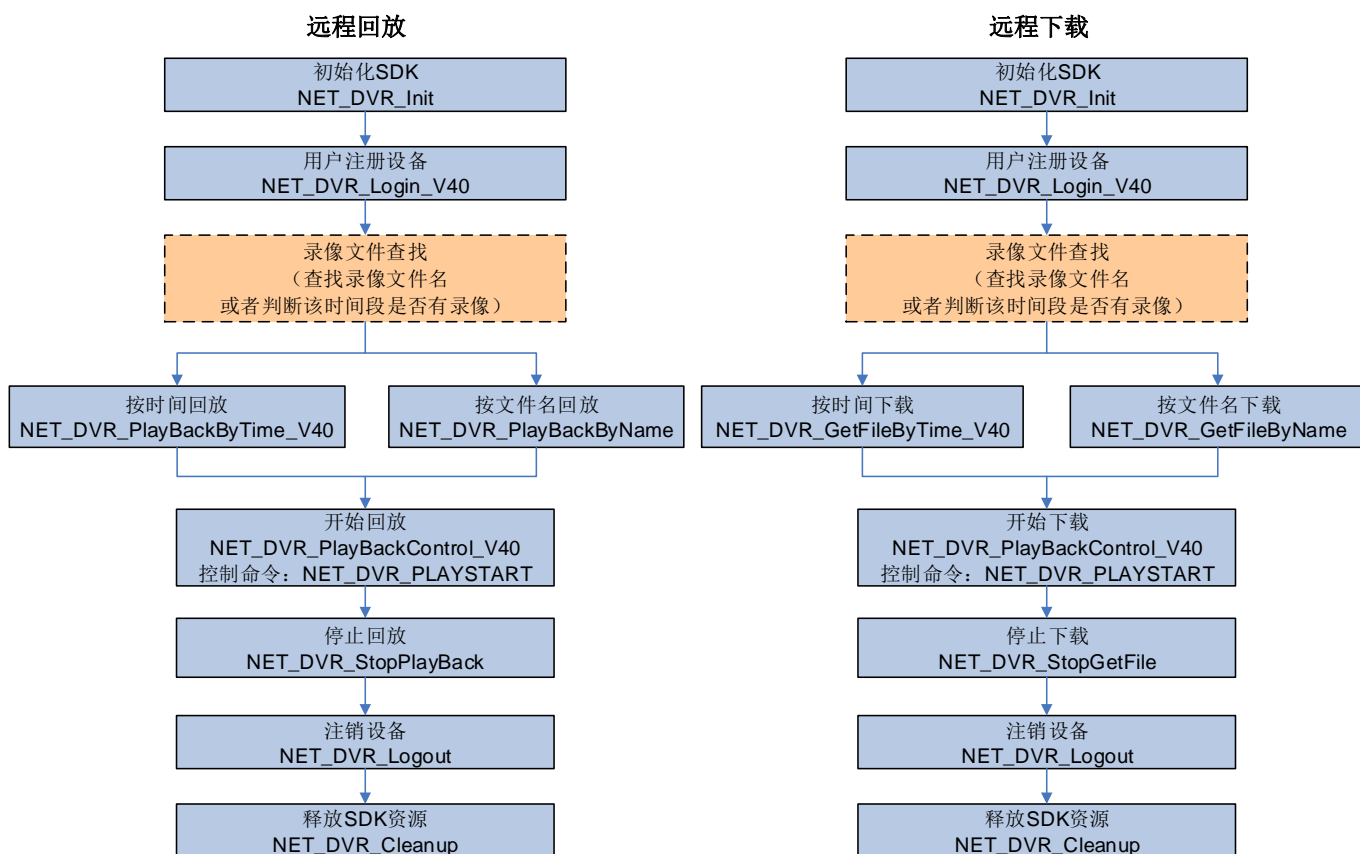
该流程介绍查找、回放、下载网络摄像机或者网络球机 SD 卡中的录像文件，不支持查找 PC 上存储的录像文件。

图 3.3 录像文件查找流程



- 按文件回放或下载需要通过查找录像文件功能先获取文件名等信息，相关接口：[NET_DVR_FindFile_V40](#)、[NET_DVR_FindNextFile_V40](#)。

图 3.4 回放下载流程

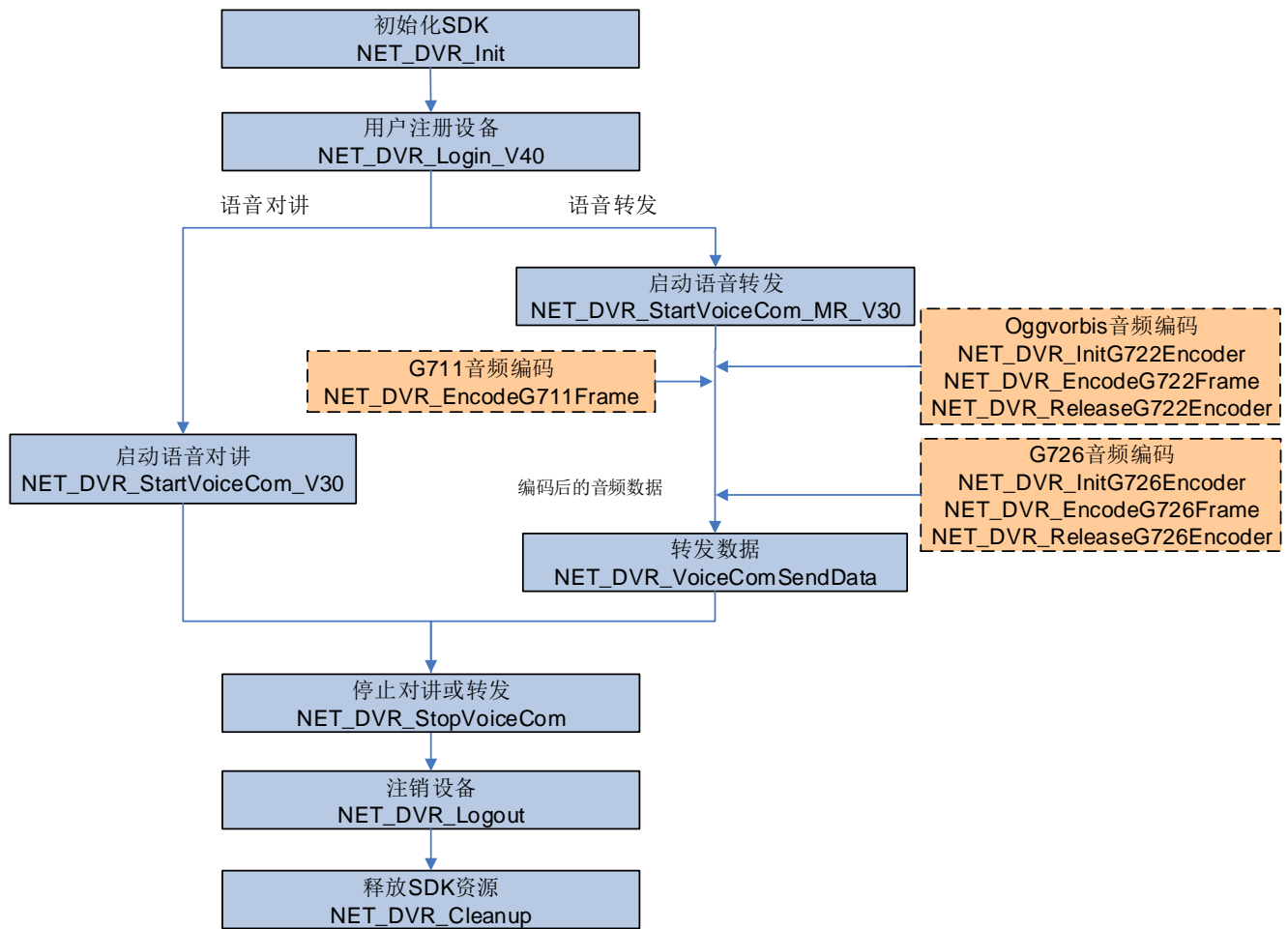


- 按文件名回放和下载：根据获取到的文件名进行回放或下载，相关接口：[NET_DVR_PlayBackByName](#)、[NET_DVR_GetFileByName](#)。在调用了回放或下载的接口后，还必须调用控制接口（[NET_DVR_PlayBackControl_V40](#)）的开始播放命令（NET_DVR_PLAYSTART）。
- 按时间回放或下载文件：用户可以无需调用查找录像文件的相关接口，只需要指定开始和结束时间即可，相关接口：[NET_DVR_PlayBackByTime_V40](#)、[NET_DVR_GetFileByTime_V40](#)，将按照指定时间范围内最近的有录像的时间段开始回放或下载。如果指定时间段内没有录像文件，按时间回放或下载接口不会返回失败，回放数据或下载文件为 0KB，因此用户可以在回放和下载之前通过查找录像文件的相关接口获取文件的开始和结束时间后，再按这个时间范围指定回放或下载接口中的时间参数，避免没有录像数据。注意，调用回放和下载接口之后，必须调用控制接口 [NET_DVR_PlayBackControl_V40](#)（命令：NET_DVR_PLAYSTART）控制开始回放或下载。

[调用实例代码](#)

3.3 语音对讲转发模块流程

图 3.5 语音对讲和转发模块流程



- 语音对讲功能实现 PC 机与设备间音频的发送和接收。在成功注册设备后调用 [NET_DVR_StartVoiceCom_V30](#) 接口完成，同时在该接口中用户可以通过设置回调函数获取当前设备发送或者 PC 机采集的数据（按需要选择回调编码后或者 PCM 数据）。
- 语音转发功能实现将待发送的音频数据（编码后）转发给设备。
 首先，调用 [NET_DVR_StartVoiceCom_MR_V30](#) 接口启动与某台设备的语音转发（此时已建立与设备之间的连接，等待发送数据）。
 第二，准备好待发送的数据（需要经过编码），相对于上图中虚线框部分，如果数据已按本公司的压缩格式处理这部分就可以省略。数据源可以从 PC 声卡中采集或是从文件中读取，但是需要经过本公司提供的压缩算法进行压缩处理，SDK 提供一套编码接口，方法如下：
 - 设备语音对讲音频编码格式为 G711：
调用 [NET_DVR_EncodeG711Frame](#) 进行音频编码。
 - 设备语音对讲音频编码格式为 G722：
 1) [NET_DVR_InitG722Encoder](#) 初始化音频编码；
 2) [NET_DVR_EncodeG722Frame](#) 进行 G722 音频编码；
 3) 当结束所有的编码过程，需要调用 [NET_DVR_ReleaseG722Encoder](#) 释放编码音频资源。
 - 设备语音对讲音频编码格式为 G726：

- 1) [NET_DVR_InitG726Encoder](#) 初始化音频编码;
- 2) [NET_DVR_EncodeG726Frame](#) 进行 G726 音频编码;
- 3) 当结束所有的编码过程, 需要调用 [NET_DVR_ReleaseG726Encoder](#) 释放编码音频资源。

第三, 经过第二部的编码操作, 我们可以每次得到固定大小的且经过编码后的数据, 调用 [NET_DVR_VoiceComSendData](#) 接口发送这些数据给设备。

第四, 从设备发送给客户端的数据可以通过 [NET_DVR_StartVoiceCom MR V30](#) 接口设置的回调函数获取, 获取到的数据可以通过语音解码接口进行解码, 获取 PCM 音频数据, 输出到声卡输出或者做其他处理。

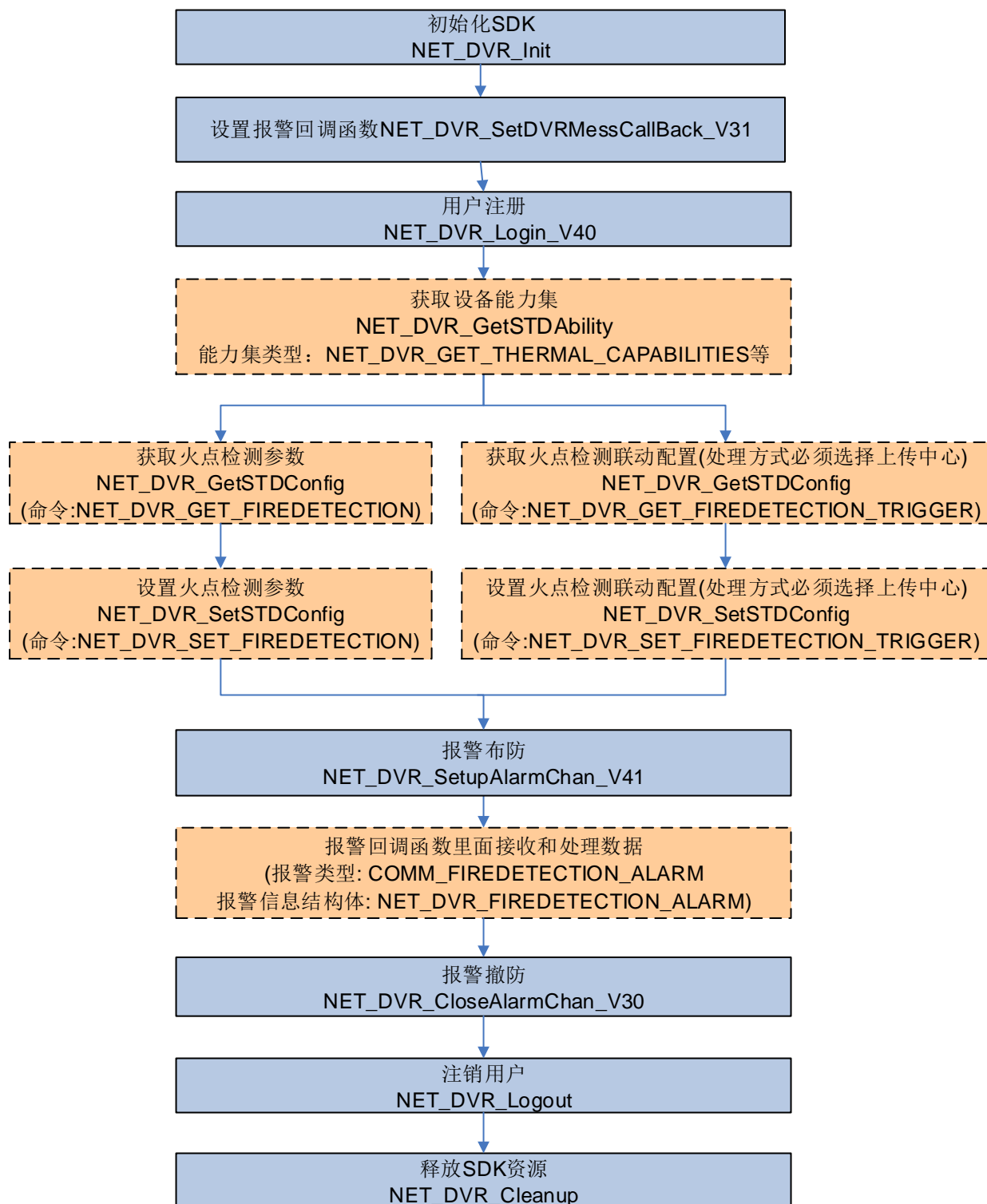
最后, 等所有的转发操作完成后, 调用 [NET_DVR_StopVoiceCom](#) 接口结束与设备的语音转发连接。

- Windows 64 位或者 Linux 系统下只支持语音转发功能, 语音对讲、语音广播、音频编解码接口均不支持

[调用实例代码](#)

3.4 动态火点搜索功能流程

图 3.6 动态火点搜索



热成像设备一般有两个通道，第一个通道为可见光通道，第二个通道为热成像通道。热成像相关功能配置只支持在热成通道上实现。

- 登录设备之后，调用 [NET_DVR_GetSTDAbility](#) 可以获取设备是否支持热成像功能、支持配置的火点检测参数能力和联动能力，分别对应能力集类型：NET_DVR_GET_THERMAL_CAPABILITIES、NET_DVR_GET_FIREDETECTION_CAPABILITIES、NET_DVR_GET_EVENT_TRIGGERS_CAPABILITIES。
- 火点检测功能对应的参数配置和联动方式可以通过 WEB 网页直接访问设备配置，也可以通过 SDK 接口进行配置，如果事先已经配置则可省略该过程。相关接口：[NET_DVR_GetSTDConfig](#)(命令:NET_DVR_GET_FIREDETECTION)、[NET_DVR_SetSTDConfig](#)(命令:NET_DVR_SET_FIREDETECTION); [NET_DVR_GetSTDConfig](#)(命令:NET_DVR_GET_FIREDETECTION_TRIGGER)、[NET_DVR_SetSTDConfig](#)(命令:NET_DVR_SET_FIREDETECTION_TRIGGER)。
- 配置检测参数和联动方式之后，设备将自动进行扫描检测，检测到火点时设备可按照配置的参数联动云台、录像或者报警。客户端可以通过报警布防方式获取设备上传的报警信息，此时联动参数中的处理方式必须选择“上传中心”类型。报警布防方式实现方法：1) 先调用 [NET_DVR_SetDVRMessageCallBack V31](#) 设置报警回调函数，在 SDK 初始化之后即可以调用，多台设备对接时也只需要调用一次设置一个回调函数；2) 每台设备分别登录，分别调用 [NET_DVR_SetupAlarmChan V41](#) 进行布防，布防即建立设备跟客户端之间报警上传的连接通道，这样设备发生报警之后通过该连接上传报警信息，SDK 在报警回调函数中接收和处理报警信息数据即可；3) 程序退出前或者不需要接收报警信息时调用 [NET_DVR_CloseAlarmChan V30](#) 进行撤防，释放资源，此时连接断开，设备将不再上传报警信息。
- 设备火点检测功能，检测到火点时，除了联动云台、录像和报警，在预览画面中也会以红色框将火点位置标识出来（如图 3.7），该信息直接叠加到码流里面，客户端预览取流后使用我司播放库解码显示时将直接显示该信息。调用 [NET_DVR_RealPlay V40](#) 传窗口句柄直接预览默认显示该叠加信息，此时调用 [NET_DVR_GetRealPlayerIndex](#) 获取播放库句柄，然后调用播放库接口 [PlayM4_RenderPrivateData](#)（该接口调用请参考播放库 SDK 编程指南）可以控制显示或者取消显示。

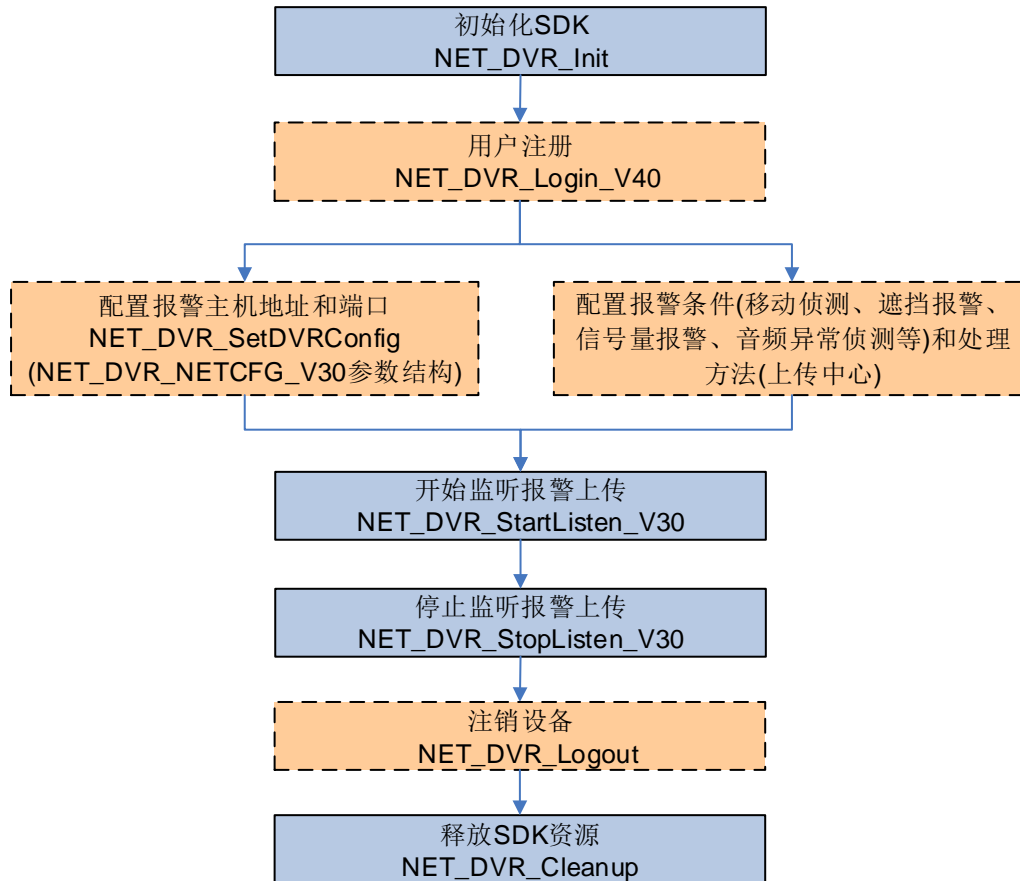
图 3.7 动态火点检测



[调用实例代码](#)

3.5 报警监听模块流程

图 3.8 报警监听流程



设备除了支持动态火点搜索功能，还支持移动侦测、遮挡报警、报警输入信号量报警和音频异常侦测等报警功能，这些报警信息可以通过报警布防方式获取，具体实现方法同动态火点检测报警信息获取，可以参考 3.4 章节；也可以通过报警监听方式获取，本章节即介绍报警监听方式接口调用流程。

- 报警监听方式：指 SDK 不主动发起连接设备，只是在设定的端口上启动监听，接收设备主动上传的报警信息。设备发生报警时，根据配置的报警主机 IP 和端口，主动连接该 PC，然后上传相关报警信息。
- 报警相关参数配置：首先，需要配置设备端网络参数中的报警主机地址（即启动监听的 PC 端 IP 地址）和报警主机端口（即启动监听的 PC 端监听端口），PC 在该端口上监听接收设备主动上传的报警信息，相关接口：[NET_DVR_GetDVRConfig](#)(命令: NET_DVR_GET_NETCFG_V30)、[NET_DVR_SetDVRConfig](#)(命令: NET_DVR_SET_NETCFG_V30)，对应结构体 NET_DVR_NETCFG_V30 中的参数 struAlarmHostIpAddr、wAlarmHostIpPort。其次，需要配置移动侦测、遮挡报警、报警输入、音频异常侦测相关参数，包括灵敏度、布防时间、联动方式（处理方式必须选择“上传中心”类型）等，相关接口：
 - 移动侦测、遮挡报警等参数配置：[NET_DVR_GetDVRConfig](#)(命令: NET_DVR_GET_PICCFG_V40)、[NET_DVR_SetDVRConfig](#)(命令: NET_DVR_SET_PICCFG_V40)；
 - 信号量报警输入参数配置：[NET_DVR_GetDVRConfig](#)(命令: NET_DVR_GET_ALARMINGCFG_V40)、[NET_DVR_SetDVRConfig](#)(命令: NET_DVR_SET_ALARMINGCFG_V40)；

- 音频异常侦测参数配置 [NET_DVR_GetDeviceConfig](#)(命令: NET_DVR_GET_AUDIOEXCEPTIONPARAM)、[NET_DVR_SetDeviceConfig](#)(命令: NET_DVR_SET_AUDIOEXCEPTIONPARAM)。

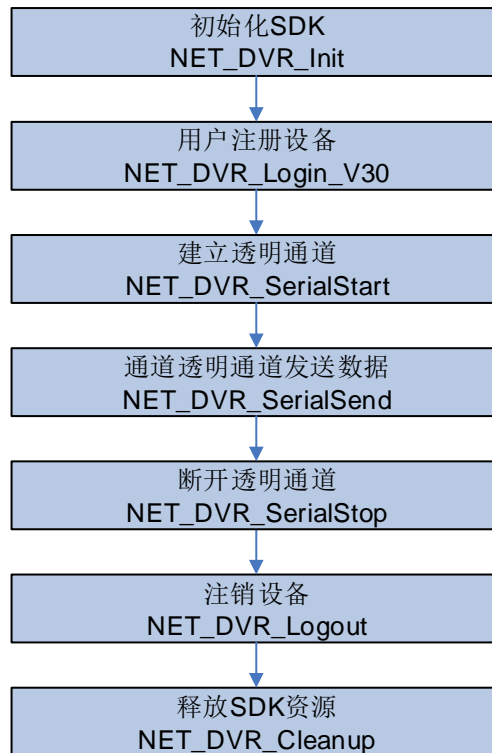
如果相关参数事先已经配置完成，那么该配置部分可以省略。

- 需要配置的参数都设置完后，调用 [NET_DVR_StartListen V30](#) 函数，开启 SDK 的监听端口，设置报警回调函数，准备接收设备上传的报警信息。启动监听的 IP 和端口必须跟设备端配置的报警主机地址和端口一致。
- 报警监听方式适用于多个设备向一台客户端上传报警，而且不需要设备登录即可完成，设备重启后不影响报警上传；缺点是设备只支持一个报警主机地址和端口号的配置。

[调用实例代码](#)

3.6 透明通道模块流程

图 3.9 透明通道模块流程



- SDK 提供将 485 和 232 串口作为透明通道，要将 232 串口作为透明通道使用，首先必须在 232 串口的配置信息中将工作模式选为透明通道，具体方法是调用接口 [NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#) 获取和设置参数 NET_DVR_RS232CFG 中的 dwWorkMode 值为透明通道。如果是 485 串口作为透明通道，这个步骤可以省略，调用 [NET_DVR_SerialStart](#) 建立透明通道和 [NET_DVR_SerialSend](#) 发送数据。整个过程结束还需要断开透明通道（[NET_DVR_SerialStop](#)）等操作。

[调用实例代码](#)

4 函数调用实例

4.1 预览模块的示例代码

[相关模块流程图](#)

方式一 SDK 直接解码显示

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

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

void CALLBACK g_ExceptionCallBack(DWORD dwType, LONG IUserID, LONG IHandle, void *pUser)
{
    char tempbuf[256] = {0};
    switch(dwType)
    {
        case EXCEPTION_RECONNECT:    //预览时重连
            printf("-----reconnect-----%d\n", time(NULL));
            break;
        default:
            break;
    }
}

void main() {
    //-----
    //初始化
    NET_DVR_Init();
    //设置连接时间与重连时间
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    //设置异常消息回调函数
    NET_DVR_SetExceptionCallBack_V30(0, NULL, g_ExceptionCallBack, NULL);

    //-----
}
```

```
// 获取控制台窗口句柄
```

```
HMODULE hKernel32 = GetModuleHandle("kernel32");
```

```
GetConsoleWindowHandle = (PROCGETCONSOLEWINDOW)GetProcAddress(hKernel32, "GetConsoleWindow");
```

```
HWND hWnd = GetConsoleWindowHandle(); //获取窗口句柄
```

```
//-----
```

```
// 注册设备
```

```
LONG IUserID;
```

```
//登录参数，包括设备地址、登录用户、密码等
```

```
NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
```

```
struLoginInfo.bUseAsyncLogin = 0; //同步登录方式
```

```
strcpy(struLoginInfo.sDeviceAddress, "192.0.0.64"); //设备 IP 地址
```

```
struLoginInfo.wPort = 8000; //设备服务端口
```

```
strcpy(struLoginInfo.sUserName, "admin"); //设备登录用户名
```

```
strcpy(struLoginInfo.sPassword, "12345"); //设备登录密码
```

```
//设备信息，输出参数
```

```
NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};
```

```
IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
```

```
if (IUserID < 0)
```

```
{
```

```
    printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
```

```
    NET_DVR_Cleanup();
```

```
    return;
```

```
}
```

```
//-----
```

```
//启动预览并设置回调数据流
```

```
LONG IRealPlayHandle;
```

```
NET_DVR_PREVIEWINFO struPlayInfo = {0};
```

```
struPlayInfo.hPlayWnd = hWnd; //需要 SDK 解码时句柄设为有效值，仅取流不解码时可设为空
```

```
struPlayInfo.lChannel = 1; //预览通道号
```

```
struPlayInfo.dwStreamType = 0; //0-主码流，1-子码流，2-码流 3，3-码流 4，以此类推
```

```
struPlayInfo.dwLinkMode = 0; //0- TCP 方式，1- UDP 方式，2- 多播方式，3- RTP 方式，4-RTP/RTSP，5-RSTP/HTTP
```

```
struPlayInfo.bBlocked = 1; //0- 非阻塞取流，1- 阻塞取流
```

```
struPlayInfo.byProtoType = 0; //应用层取流协议：0- 私有协议，1- RTSP 协议
```

```
IRealPlayHandle = NET_DVR_RealPlay_V40(IUserID, &struPlayInfo, NULL, NULL);
```

```
if (IRealPlayHandle < 0)
```

```
{
```

```
    printf("NET_DVR_RealPlay_V40 error\n");
```

```
    NET_DVR_Logout(IUserID);
```

```

    NET_DVR_Cleanup();
    return;
}
Sleep(10000);
//-----
//关闭预览
NET_DVR_StopRealPlay(IRealPlayHandle);
//注销用户
NET_DVR_Logout(IUserID);
//释放 SDK 资源
NET_DVR_Cleanup();
return;
}

```

方式二 实时流数据回调，用户自行处理码流数据（此处以软解显示为例）

```

#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
#include "plaympeg4.h"
#include <time.h>
using namespace std;

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

LONG lPort = -1; //全局的播放库 port 号

void CALLBACK g_RealDataCallBack_V30(LONG lRealHandle, DWORD dwDataType, BYTE *pBuffer, DWORD dwBufSize, void* dwUser)
{
    HWND hWnd= GetConsoleWindowHandle();

    switch (dwDataType)
    {
    {
    case NET_DVR_SYSHEAD: //系统头
        if(lPort > -1)
        {
            break; //同一路码流只需要调用一次 PlayM4_OpenStream、PlayM4_Play 等接口
        }
        if (!PlayM4_GetPort(&lPort)) //获取播放库未使用的通道号
        {
            break;
        }
    }
    }
}

```

```
//m_iPort = IPort;
/*第一次回调的是系统头，将获取的播放库 port 号赋值给全局 port，
下次回调数据时即使用此 port 号播放*/
if (dwBufSize > 0)
{
    if (!PlayM4_SetStreamOpenMode(IPort, STREAME_REALTIME)) //设置实时流播放模式
    {
        break;
    }

    if (!PlayM4_OpenStream(IPort, pBuffer, dwBufSize, 1024*1024)) //打开流接口
    {
        break;
    }

    if (!PlayM4_Play(IPort, hWnd)) //播放开始
    {
        break;
    }
}
break;
case NET_DVR_STREAMDATA: //码流数据
    if (dwBufSize > 0 && IPort != -1)
    {
        if (!PlayM4_InputData(IPort, pBuffer, dwBufSize))
        {
            break;
        }
    }
    break;
default: //其他数据
    if (dwBufSize > 0 && IPort != -1)
    {
        if (!PlayM4_InputData(IPort, pBuffer, dwBufSize))
        {
            break;
        }
    }
    break;
}

}

void CALLBACK g_ExceptionCallBack(DWORD dwType, LONG IUserID, LONG IHandle, void *pUser)
{

```

```

    char tempbuf[256] = {0};
    switch(dwType)
    {
    case EXCEPTION_RECONNECT:    //预览时重连
        printf("-----reconnect-----%d\n", time(NULL));
        break;
        default:
        break;
    }
}

void main() {

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

    //-----
    //设置异常消息回调函数
    NET_DVR_SetExceptionCallBack_V30(0, NULL, g_ExceptionCallBack, NULL);

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

    //-----
    // 注册设备
    LONG IUserID;

    //登录参数，包括设备地址、登录用户、密码等
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    struLoginInfo.bUseAsynLogin = 0; //同步登录方式
    strcpy(struLoginInfo.sDeviceAddress, "192.0.0.64"); //设备 IP 地址
    struLoginInfo.wPort = 8000; //设备服务端口
    strcpy(struLoginInfo.sUserName, "admin"); //设备登录用户名
    strcpy(struLoginInfo.sPassword, "12345"); //设备登录密码

    //设备信息，输出参数
    NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};

```

```

IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
if (IUserID < 0)
{
    printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

//-----
//启动预览并设置回调数据流
LONG IRealPlayHandle;

NET_DVR_PREVIEWINFO struPlayInfo = {0};
struPlayInfo.hPlayWnd = NULL;           //需要 SDK 解码时句柄设为有效值，仅取流不解码时可设为空
struPlayInfo.IChannel = 1;              //预览通道号
struPlayInfo.dwStreamType = 0;          //0-主码流，1-子码流，2-码流 3，3-码流 4，以此类推
struPlayInfo.dwLinkMode = 0;            //0- TCP 方式，1- UDP 方式，2- 多播方式，3- RTP 方式，4-RTP/RTSP，5-RSTP/HTTP
struPlayInfo.bBlocked = 1;              //0- 非阻塞取流，1- 阻塞取流

IRealPlayHandle = NET_DVR_RealPlay_V40(IUserID, &struPlayInfo, g_RealDataCallBack_V30, NULL);
if (IRealPlayHandle < 0)
{
    printf("NET_DVR_RealPlay_V40 error\n");
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

Sleep(10000);
//-----
//关闭预览
NET_DVR_StopRealPlay(IRealPlayHandle);
IPort = -1;

//注销用户
NET_DVR_Logout(IUserID);
NET_DVR_Cleanup();

return;
}

```

4.2 回放和下载模块的示例代码

[相关模块流程图](#)

示例一：查找录像文件并下载

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

int saveRecordFile(int userId,char * srcfile,char * destfile)
{
    int bRes = 1;
    int hPlayback = 0;

    //按文件名下载录像
    if( (hPlayback = NET_DVR_GetFileByName(userId, srcfile, destfile)) < 0 )
    {
        printf( "GetFileByName failed. error[%d]\n", NET_DVR_GetLastError());
        bRes= -1;
        return bRes;
    }

    //开始下载
    if(!NET_DVR_PlayBackControl_V40(hPlayback, NET_DVR_PLAYSTART, NULL,0,NULL,NULL))
    {
        printf("play back control failed [%d]\n",NET_DVR_GetLastError());
        bRes=-1;
        return bRes;
    }

    int nPos = 0;
    for(nPos = 0; nPos < 100&& nPos>=0; nPos = NET_DVR_GetDownloadPos(hPlayback))
    {
        printf("Be downloading...%d %%\n", nPos);    //下载进度
        Sleep(5000);    //millisecond
    }
    printf("have got %d\n", nPos);
    //停止下载
    if(!NET_DVR_StopGetFile(hPlayback))
    {
        printf("failed to stop get file [%d]\n",NET_DVR_GetLastError());
        bRes = -1;
        return bRes;
    }
    printf("%s\n",srcfile);
}
```

```

    if(nPos<0||nPos>100)
    {
        printf("download err [%d]\n",NET_DVR_GetLastError());
        bRes=-1;
        return bRes;
    }
    else
    {
        return 0;
    }
}

void main() {
    //-----
    // 初始化
    NET_DVR_Init();
    //设置连接时间与重连时间
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    // 注册设备
    LONG IUserID;

    //登录参数，包括设备地址、登录用户、密码等
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    struLoginInfo.bUseAsyncLogin = 0; //同步登录方式
    strcpy(struLoginInfo.sDeviceAddress, "192.0.0.64"); //设备 IP 地址
    struLoginInfo.wPort = 8000; //设备服务端口
    strcpy(struLoginInfo.sUserName, "admin"); //设备登录用户名
    strcpy(struLoginInfo.sPassword, "12345"); //设备登录密码

    //设备信息，输出参数
    NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};

    IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
    if (IUserID < 0)
    {
        printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }

    NET_DVR_FILECOND_V40 struFileCond={0};
    struFileCond.dwFileType = 0xFF;

```



```

struFileCond.IChannel = 1; //通道号
struFileCond.dwIsLocked = 0xFF;
struFileCond.dwUseCardNo = 0;
struFileCond.struStartTime.dwYear    = 2011; //开始时间
struFileCond.struStartTime.dwMonth   = 3;
struFileCond.struStartTime.dwDay     = 1;
struFileCond.struStartTime.dwHour    = 10;
struFileCond.struStartTime.dwMinute  = 6;
struFileCond.struStartTime.dwSecond  = 50;
struFileCond.struStopTime.dwYear     = 2011; //结束时间
struFileCond.struStopTime.dwMonth    = 3;
struFileCond.struStopTime.dwDay      = 1;
struFileCond.struStopTime.dwHour     = 11;
struFileCond.struStopTime.dwMinute   = 7;
struFileCond.struStopTime.dwSecond   = 0;

//-----
//查找录像文件
int IFindHandle = NET_DVR_FindFile_V40(IUserID, &struFileCond);
if(IFindHandle < 0)
{
    printf("find file fail,last error %d\n",NET_DVR_GetLastError());
    return;
}
NET_DVR_FINDDATA_V40 struFileData;
while(true)
{
    //逐个获取查找到的文件信息
    int result = NET_DVR_FindNextFile_V40(IFindHandle, &struFileData);
    if(result == NET_DVR_ISFINDING)
    {
        continue;
    }
    else if(result == NET_DVR_FILE_SUCCESS) //获取文件信息成功
    {
        char strFileName[256] = {0};
        sprintf(strFileName, "%s", struFileData.sFileName);
        saveRecordFile(IUserID, struFileData.sFileName, strFileName);
        break;
    }
    else if(result == NET_DVR_FILE_NOFIND || result == NET_DVR_NOMOREFILE) //未查找到文件或者查找结束
    {
        break;
    }
}

```

```
    else
    {
        printf("find file fail for illegal get file state");
        break;
    }
}

//停止查找
if(IFindHandle >= 0)
{
    NET_DVR_FindClose_V30(IFindHandle);
}

//注销用户
NET_DVR_Logout(IUserID);
//释放 SDK 资源
NET_DVR_Cleanup();
return;
}
```

示例二：按时间播放录像文件

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

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

void main() {
    //-----
    // 初始化
    NET_DVR_Init();
    //设置连接时间与重连时间
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

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

```

//-----
// 注册设备
LONG IUserID;

//登录参数, 包括设备地址、登录用户、密码等
NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
struLoginInfo.bUseAsyncLogin = 0; //同步登录方式
strcpy(struLoginInfo.sDeviceAddress, "192.0.0.64"); //设备 IP 地址
struLoginInfo.wPort = 8000; //设备服务端口
strcpy(struLoginInfo.sUserName, "admin"); //设备登录用户名
strcpy(struLoginInfo.sPassword, "12345"); //设备登录密码

//设备信息, 输出参数
NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};

IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
if (IUserID < 0)
{
    printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

HWND hWnd = GetConsoleWindowHandle(); //获取窗口句柄

NET_DVR_VOD_PARA struVodPara={0};
struVodPara.dwSize=sizeof(struVodPara);
struVodPara.struIDInfo.dwChannel=1; //通道号
struVodPara.hWnd=hWnd; //回放窗口
struVodPara.struBeginTime.dwYear   = 2013; //开始时间
struVodPara.struBeginTime.dwMonth  = 6;
struVodPara.struBeginTime.dwDay    = 14;
struVodPara.struBeginTime.dwHour   = 9;
struVodPara.struBeginTime.dwMinute = 0;
struVodPara.struBeginTime.dwSecond =0;
struVodPara.struEndTime.dwYear     = 2013; //结束时间
struVodPara.struEndTime.dwMonth    = 6;
struVodPara.struEndTime.dwDay      = 14;
struVodPara.struEndTime.dwHour     = 10;
struVodPara.struEndTime.dwMinute   = 7;
struVodPara.struEndTime.dwSecond   = 0;

//-----
//按时间回放

```

```

int hPlayback;
hPlayback = NET_DVR_PlayBackByTime_V40(IUserID, &struVodPara);
if(hPlayback < 0)
{
    printf("NET_DVR_PlayBackByTime_V40 fail,last error %d\n",NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}
//-----
//开始
if(!NET_DVR_PlayBackControl_V40(hPlayback, NET_DVR_PLAYSTART,NULL, 0, NULL,NULL))
{
    printf("play back control failed [%d]\n",NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

Sleep(15000); //millisecond
if(!NET_DVR_StopPlayBack(hPlayback))
{
    printf("failed to stop file [%d]\n",NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//注销用户
NET_DVR_Logout(IUserID);
//释放 SDK 资源
NET_DVR_Cleanup();
return;
}

```

示例三：按时间下载录像文件

```

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

void main() {

```

```

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

//-----
// 注册设备
LONG IUserID;

//登录参数, 包括设备地址、登录用户、密码等
NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
struLoginInfo.bUseAsynLogin = 0; //同步登录方式
strcpy(struLoginInfo.sDeviceAddress, "192.0.0.64"); //设备 IP 地址
struLoginInfo.wPort = 8000; //设备服务端口
strcpy(struLoginInfo.sUserName, "admin"); //设备登录用户名
strcpy(struLoginInfo.sPassword, "12345"); //设备登录密码

//设备信息, 输出参数
NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};

IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
if (IUserID < 0)
{
    printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

NET_DVR_PLAYCOND struDownloadCond={0};
struDownloadCond.dwChannel=1; //通道号
struDownloadCond.struStartTime.dwYear    = 2013; //开始时间
struDownloadCond.struStartTime.dwMonth   = 6;
struDownloadCond.struStartTime.dwDay     = 14;
struDownloadCond.struStartTime.dwHour    = 9;
struDownloadCond.struStartTime.dwMinute  = 50;
struDownloadCond.struStartTime.dwSecond  = 0;
struDownloadCond.struStopTime.dwYear     = 2013; //结束时间
struDownloadCond.struStopTime.dwMonth    = 6;
struDownloadCond.struStopTime.dwDay      = 14;
struDownloadCond.struStopTime.dwHour     = 10;
struDownloadCond.struStopTime.dwMinute   = 7;
struDownloadCond.struStopTime.dwSecond   = 0;

```

```
//-----  
//按时间下载  
int hPlayback;  
hPlayback = NET_DVR_GetFileByTime_V40(IUserID, "/test.mp4",&struDownloadCond);  
if(hPlayback < 0)  
{  
    printf("NET_DVR_GetFileByTime_V40 fail,last error %d\n",NET_DVR_GetLastError());  
    NET_DVR_Logout(IUserID);  
    NET_DVR_Cleanup();  
    return;  
}  
  
//-----  
//开始下载  
if(!NET_DVR_PlayBackControl_V40(hPlayback, NET_DVR_PLAYSTART, NULL, 0, NULL,NULL))  
{  
    printf("Play back control failed [%d]\n",NET_DVR_GetLastError());  
    NET_DVR_Logout(IUserID);  
    NET_DVR_Cleanup();  
    return;  
}  
  
int nPos = 0;  
for(nPos = 0; nPos < 100&& nPos>=0; nPos = NET_DVR_GetDownloadPos(hPlayback))  
{  
    printf("Be downloading... %d %%\n",nPos); //下载进度  
    Sleep(5000); //millisecond  
}  
if(!NET_DVR_StopGetFile(hPlayback))  
{  
    printf("failed to stop get file [%d]\n",NET_DVR_GetLastError());  
    NET_DVR_Logout(IUserID);  
    NET_DVR_Cleanup();  
    return;  
}  
if(nPos<0 | nPos>100)  
{  
    printf("download err [%d]\n",NET_DVR_GetLastError());  
    NET_DVR_Logout(IUserID);  
    NET_DVR_Cleanup();  
    return;  
}  
printf("Be downloading... %d %%\n",nPos);
```

```
//注销用户
NET_DVR_Logout(IUserID);
//释放 SDK 资源
NET_DVR_Cleanup();
return;
}
```

4.3 语音对讲转发模块的示例代码

[相关模块流程图](#)

示例一：语音对讲

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

void CALLBACK fVoiceDataCallBack(LONG lVoiceComHandle, char *pRecvDataBuffer, DWORD dwBufSize, BYTE byAudioFlag,
void*pUser)
{
    printf("receive voice data, %d\n", dwBufSize);
}

void main() {
    //-----
    // 初始化
    NET_DVR_Init();
    //设置连接时间与重连时间
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    // 注册设备
    LONG lUserID;

    //登录参数，包括设备地址、登录用户、密码等
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    struLoginInfo.bUseAsynLogin = 0; //同步登录方式
    strcpy(struLoginInfo.sDeviceAddress, "192.0.0.64"); //设备 IP 地址
    struLoginInfo.wPort = 8000; //设备服务端口
    strcpy(struLoginInfo.sUserName, "admin"); //设备登录用户名
}
```

```
strcpy(struLoginInfo.sPassword, "12345"); //设备登录密码

//设备信息, 输出参数
NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};

IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
if (IUserID < 0)
{
    printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

//语音对讲
LONG IVoiceHanle;
IVoiceHanle = NET_DVR_StartVoiceCom_V30(IUserID, 1, 0, fVoiceDataCallBack, NULL);
if (IVoiceHanle < 0)
{
    printf("NET_DVR_StartVoiceCom_V30 error, %d!\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

Sleep(5000); //millisecond
//关闭语音对讲
if (!NET_DVR_StopVoiceCom(IVoiceHanle))
{
    printf("NET_DVR_StopVoiceCom error, %d!\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//注销用户
NET_DVR_Logout(IUserID);
//释放 SDK 资源
NET_DVR_Cleanup();
return;
}
```

4.4 动态火点搜索功能的示例代码

[相关模块流程图](#)


```

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

BOOL CALLBACK MessageCallback(LONG ICommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD dwBufLen, void*
pUser)
{
    int i;

    switch(ICommand)
    {
    case COMM_ALARM_V30:
        {
            NET_DVR_ALARMINFO_V30 struAlarmInfo = {0};
            memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO_V30));
            switch (struAlarmInfo.dwAlarmType)
            {
            case 3: //移动侦测报警
                for (i=0; i<MAX_CHANNUM_V30; i++)
                {
                    if (struAlarmInfo.byChannel[i] == 1)
                    {
                        printf("发移动侦测报警, 报警通道号:%d\n", i+1);
                    }
                }
                break;
            default:
                break;
            }
        }
        break;
    case COMM_FIREDETECTION_ALARM: //火点检测报警
        {
            NET_DVR_FIREDETECTION_ALARM struFireDetection = {0};
            memcpy(&struFireDetection, pAlarmInfo, sizeof(NET_DVR_FIREDETECTION_ALARM));
            printf("火点检测报警: RelativeTime:%d, AbsTime:%d, PTZ{PanPos:%d, TiltPos:%d, ZoomPos:%d}, \
PicDataLen:%d, DevInfo{DevIP:%s, Port:%d, Channel:%d, IvmsChannel:%d}, \
FireMaxTemperature:%d, TargetDistance:%d, fireRectInfo{fX:%f,fY:%f,fWidth%f,fHeight%f}, \
fireMaxTemperaturePoint{fX:%f,fY:%f}\n", struFireDetection.dwRelativeTime, \
struFireDetection.dwAbsTime, struFireDetection.wPanPos, struFireDetection.wTiltPos, \
struFireDetection.wZoomPos, struFireDetection.dwPicDataLen, \
struFireDetection.struDevInfo.struDevIP.slpV4, struFireDetection.struDevInfo.wPort, \

```

```
        struFireDetection.struDevInfo.byChannel, struFireDetection.struDevInfo.byIvmsChannel, \
        struFireDetection.wFireMaxTemperature, struFireDetection.wTargetDistance, \
        struFireDetection.struRect.fx, struFireDetection.struRect.fy, struFireDetection.struRect.fWidth, \
        struFireDetection.struRect.fHeight, struFireDetection.struPoint.fx, struFireDetection.struPoint.fy);
    }
    break;
default:
    printf("其他报警，报警信息类型: %d\n", ICommand);
    break;
}

return TRUE;
}

void main() {
    //-----
    // 初始化
    NET_DVR_Init();
    //设置连接时间与重连时间
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //-----
    // 注册设备
    LONG IUserID;

    //登录参数，包括设备地址、登录用户、密码等
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    struLoginInfo.bUseAsyncLogin = 0; //同步登录方式
    strcpy(struLoginInfo.sDeviceAddress, "10.16.4.55"); //设备 IP 地址
    struLoginInfo.wPort = 8000; //设备服务端口
    strcpy(struLoginInfo.sUserName, "admin"); //设备登录用户名
    strcpy(struLoginInfo.sPassword, "hik123456"); //设备登录密码

    //设备信息，输出参数
    NET_DVR_DEVICEINFO_V40 struDeviceInfoV40 = {0};

    IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfoV40);
    if (IUserID < 0)
    {
        printf("Login failed, error code: %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }
}
```

```
//设置报警回调函数
NET_DVR_SetDVRMessageCallBack_V31(MessageCallback, NULL);

//启用布防
LONG IHandle;
NET_DVR_SETUPALARM_PARAM struAlarmParam={0};
struAlarmParam.dwSize=sizeof(struAlarmParam);
//火点检测不需要设置其他报警布防参数，不支持

IHandle = NET_DVR_SetupAlarmChan_V41(IUserID, & struAlarmParam);
if (IHandle < 0)
{
    printf("NET_DVR_SetupAlarmChan_V41 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

Sleep(50000); //等待过程中，如果设备上传报警信息，在报警回调函数里面接收和处理报警信息

//撤销布防上传通道
if (!NET_DVR_CloseAlarmChan_V30(IHandle))
{
    printf("NET_DVR_CloseAlarmChan_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//注销用户
NET_DVR_Logout(IUserID);
//释放 SDK 资源
NET_DVR_Cleanup();
return;
}
```

4.5 报警监听模块的示例代码

[相关模块流程图](#)

```
#include <stdio.h>
#include <iostream>
#include "Windows.h"
#include "HCNetSDK.h"
```

```
using namespace std;

void CALLBACK MessageCallback(LONG ICommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD dwBufLen, void*
pUser)
{
    int i;
    switch(ICommand)
    {
        case COMM_ALARM_V30:
        {
            NET_DVR_ALARMINFO_V30 struAlarmInfo = {0};
            memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO_V30));
            switch (struAlarmInfo.dwAlarmType)
            {
                case 3: //移动侦测报警
                {
                    for (i=0; i<MAX_CHANNUM_V30; i++)
                    {
                        if (struAlarmInfo.byChannel[i] == 1)
                        {
                            printf("发移动侦测报警, 报警通道号:%d\n", i+1);
                        }
                    }
                    break;
                }
                default:
                {
                    break;
                }
            }
            break;
        }
        case COMM_ALARM_AUDIOEXCEPTION: //音频异常侦测报警
        {
            NET_DVR_AUDIOEXCEPTION_ALARM struAudioException = {0};
            memcpy(&struAudioException, pAlarmInfo, sizeof(NET_DVR_ALARMINFO_V30));
            switch (struAudioException.byAlarmType)
            {
                case 1:
                {
                    printf("音频输入异常报警!");
                    break;
                }
                case 2:
                {
                    printf("音频输入突变报警!");
                    break;
                }
                case 3:
                {
                    printf("声强陡降报警!");
                    break;
                }
                case 4:
```

```
                printf("音频丢失报警!");
                break;

            default:
                printf("未知报警!");
                break;
        }
        break;
    }
}

default:
    printf("其他报警, 报警信息类型: %d\n", ICommand);
    break;
}
}

void main() {
    //-----
    // 初始化
    NET_DVR_Init();
    //设置连接时间与重连时间
    NET_DVR_SetConnectTime(2000, 1);
    NET_DVR_SetReconnect(10000, true);

    //启用监听
    LONG IHandle;
    IHandle = NET_DVR_StartListen_V30(NULL, 7200, MessageCallback, NULL);
    if (IHandle < 0)
    {
        printf("NET_DVR_StartListen_V30 error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Logout(IUserID);
        NET_DVR_Cleanup();
        return;
    }

    Sleep(5000);
    //停止监听
    if (!NET_DVR_StopListen_V30(IHandle))
    {
        printf("NET_DVR_StopListen_V30 error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Logout(IUserID);
        NET_DVR_Cleanup();
        return;
    }

    //释放 SDK 资源
```

```
NET_DVR_Cleanup();  
return;  
}
```

4.6 透明通道模块的示例代码

[相关模块流程图](#)

```
#include <stdio.h>  
#include <iostream>  
#include "Windows.h"  
#include "HCNetSDK.h"  
using namespace std;  
  
//回调透传数据函数的外部实现  
void CALLBACK g_fSerialDataCallBack(LONG lSerialHandle, char *pRecvDataBuffer, DWORD dwBufSize, DWORD dwUser)  
{  
    //..... 处理接收到的透传数据，pRecvDataBuffer 中存放接收到的数据  
}  
  
void main() {  
    //-----  
    // 初始化  
    NET_DVR_Init();  
    //设置连接时间与重连时间  
    NET_DVR_SetConnectTime(2000, 1);  
    NET_DVR_SetReconnect(10000, true);  
  
    //-----  
    // 注册设备  
    LONG lUserID;  
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;  
    lUserID = NET_DVR_Login_V30("192.0.0.64", 8000, "admin", "12345", &struDeviceInfo);  
    if (lUserID < 0)  
    {  
        printf("Login error, %d\n", NET_DVR_GetLastError());  
        NET_DVR_Cleanup();  
        return;  
    }  
  
    //设置 232 为透明通道模式（使用 232 透明通道时调用，485 不需要）  
    DWORD dwReturned = 0;  
    NET_DVR_RS232CFG_V30 struRS232Cfg;  
    memset(&struRS232Cfg, 0, sizeof(NET_DVR_RS232CFG_V30));  
    if (!NET_DVR_GetDVRConfig(lUserID, NET_DVR_GET_RS232CFG_V30, 0, &struRS232Cfg, sizeof(NET_DVR_RS232CFG_V30),
```

```

&dwReturned))
{
    printf("NET_DVR_GET_RS232CFG_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

struRS232Cfg.struRs232.dwWorkMode = 2;
//设置 232 为透明通道模式; 0: 窄带传输, 1: 控制台, 2: 透明通道
if (!NET_DVR_SetDVRConfig(IUserID, NET_DVR_SET_RS232CFG_V30, 0, &(struRS232Cfg), sizeof(NET_DVR_RS232CFG)))
{
    printf("NET_DVR_SET_RS232CFG_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//建立透明通道
LONG ITranHandle;
int iSelSerialIndex = 1; //1:232 串口; 2:485 串口
ITranHandle = NET_DVR_SerialStart(IUserID, iSelSerialIndex, g_fSerialDataCallBack, IUserID);
//设置回调函数获取透传数据
if (ITranHandle < 0)
{
    printf("NET_DVR_SerialStart error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//通过透明通道发送数据
LONG ISerialChan = 0; //使用 485 时该值有效, 从 1 开始; 232 时设置为 0
char szSendBuf[1016] = {0};
if (!NET_DVR_SerialSend(ITranHandle, ISerialChan, szSendBuf, sizeof(szSendBuf)))
//szSendBuf 为发送数据的缓冲区
{
    printf("NET_DVR_SerialSend error, %d\n", NET_DVR_GetLastError());
    NET_DVR_SerialStop(ITranHandle);
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

```

```
//断开透明通道
NET_DVR_SerialStop(ITranHandle);
//注销用户
NET_DVR_Logout(IUserID);
//释放 SDK 资源
NET_DVR_Cleanup();
return;
}
```


5 函数说明

5.1 SDK 初始化

5.1.1 初始化 SDK **NET_DVR_Init**

函 数: BOOL NET_DVR_Init()

参 数: 无

返回值: TRUE 表示成功, FALSE 表示失败。

说 明: 调用设备网络 SDK 其他函数的前提。

[返回目录](#)

5.1.2 释放 SDK 资源 **NET_DVR_Cleanup**

函 数: BOOL NET_DVR_Cleanup()

参 数: 无

返回值: TRUE 表示成功, FALSE 表示失败。

说 明: 在结束之前最后调用。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

[返回目录](#)

5.2 SDK 本地功能

SDK 本地参数配置

5.2.1 获取 SDK 本地参数 **NET_DVR_GetSDKLocalCfg**

函 数: BOOL NET_DVR_GetSDKLocalCfg(NET_SDK_LOCAL_CFG_TYPE enumType, void *lpOutBuff)

参 数: [in] enumType 配置类型, 不同的取值对应不同的 SDK 参数, 详见表 5.1

[out] lpOutBuff 输出参数, 不同的配置类型, 输出参数对应不同的结构, 详见表 5.1

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

说 明:

表 5.1 本地参数类型

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

[返回目录](#)

5.2.2 设置 SDK 本地参数 **NET_DVR_SetSDKLocalCfg**

函 数: BOOL NET_DVR_SetSDKLocalCfg(NET_SDK_LOCAL_CFG_TYPE enumType, void* const lpInBuff)
参 数: [in] enumType 配置类型, 不同的取值对应不同的 SDK 参数, 详见表 5.2
[in] lpInBuff 输入参数, 不同的配置类型, 输出参数对应不同的结构, 详见表 5.2

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

说 明:

表 5.2 本地参数类型

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

[返回目录](#)

连接和接收超时时间及重连设置

5.2.3 设置网络连接超时时间和连接尝试次数 **NET_DVR_SetConnectTime**

函 数: BOOL NET_DVR_SetConnectTime(DWORD dwWaitTime,DWORD dwTryTime)
参 数: [in]dwWaitTime 超时时间, 单位毫秒, 取值范围[300,75000], 实际最大超时时间因系统的 connect 超时时间而不同。
[in]dwTryTimes 连接尝试次数 (保留)
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
说 明: **SDK 默认建立连接的超时时间为 3 秒。** SDK4.0 及以后版本中当设置的超时时间超过或低于限制的值时接口不返回失败, 将取最接近的上下限限制值作为实际的超时时间。

[返回目录](#)

5.2.4 设置重连功能 **NET_DVR_SetReconnect**

函 数: BOOL NET_DVR_SetReconnect (DWORD dwInterval,BOOL bEnableRecon)
参 数: [in]dwInterval 重连间隔, 单位:毫秒
[in]bEnableRecon 是否重连, 0-不重连, 1-重连, 参数默认为 1
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
说 明: 该接口可以同时控制预览、透明通道和布防的重连功能。不调用该接口时, SDK 默认启动预览、透明通道和布防的重连功能, 重连时间间隔为 5 秒。

[返回目录](#)

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

函 数: BOOL NET_DVR_SetRecvTimeOut(DWORD nRecvTimeOut)
参 数: [in] nRecvTimeOut 接收超时时间, 单位毫秒, 默认为 5000, 最小为 3000 毫秒
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
说 明: 该接口用于设置接收超时时间, 例如预览接收实时流数据、回放下载接收录像数据、报警接收报警信息等接收超时时间。

[返回目录](#)

多网卡绑定

5.2.6 获取所有 IP, 用于支持多网卡接口 **NET_DVR_GetLocalIP**

函 数: BOOL NET_DVR_GetLocalIP(char strIP[16][16], DWORD *pValidNum, BOOL *pEnableBind)
参 数: [out] strIP 存放 IP 的缓冲区, 不能为空
[out] pValidNum 所有有效 IP 的数量

[out] pEnableBind 是否绑定

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

说 明: 该接口获取客户端本地多网卡的所有 IP 地址, 可以通过接口 NET_DVR_SetValidIP 选择要使用的 IP 地址。

[返回目录](#)

5.2.7 设置 IP 绑定 **NET_DVR_SetValidIP**

函 数: BOOL NET_DVR_SetValidIP(DWORD dwIPIndex, BOOL bEnableBind)

参 数: [in] dwIPIndex 选择使用的 IP 下标, 由 NET_DVR_GetLocalIP 获取
[in] bEnableBind 是否绑定

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

说 明:

[返回目录](#)

SDK 版本、状态和能力

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

函 数: DWORD NET_DVR_GetSDKBuildVersion()

参 数:

返回值: 获取 SDK 的版本号和 build 信息。

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

[返回目录](#)

5.2.9 获取当前 SDK 的状态信息 **NET_DVR_GetSDKState**

函 数: BOOL NET_DVR_GetSDKState(LPNET_DVR_SDKSTATE pSDKState);

参 数: [out]pSDKState SDK 状态信息, 详见结构体: NET_DVR_SDKSTATE

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

说 明:

[返回目录](#)

5.2.10 获取当前 SDK 的功能信息 **NET_DVR_GetSDKAbility**

函 数: BOOL NET_DVR_GetSDKAbility(LPNET_DVR_SDKABL pSDKAbI)

参 数: [out] pSDKAbI SDK 功能信息, 详见结构体: NET_DVR_SDKABL

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

说 明:

[返回目录](#)

SDK 启用写日志

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

函 数: BOOL NET_DVR_SetLogToFile(DWORD bLogEnable, char* strLogDir, BOOL bAutoDel)

参 数: [in]bLogEnable 日志的等级 (默认为 0):
 0-表示关闭日志
 1-表示只输出 ERROR 错误日志
 2-输出 ERROR 错误信息和 DEBUG 调试信息
 3-输出 ERROR 错误信息、DEBUG 调试信息和 INFO 普通信息等所有信息
 [in]strLogDir 日志文件的路径, windows 默认值为"C:\\SdkLog\\"; linux 默认值
 "/home/sdklog/"
 [in]bAutoDel 是否删除超出的文件数, 默认值为 TRUE

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

说 明:

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

[返回目录](#)

异常消息回调

5.2.12 注册异常消息回调函数 **NET_DVR_SetExceptionCallBack_V30**

函 数: [Windows 系统下:](#)

BOOL NET_DVR_SetExceptionCallBack_V30 (UINT nMessage, HWND hWnd, fExceptionCallBack
 cbExceptionCallBack, void* pUser)

[Linux 系统下:](#)

BOOL NET_DVR_SetExceptionCallBack_V30(UINT nMessage, void* hWnd, fExceptionCallBack
 cbExceptionCallBack, void* pUser)

参 数: [in]nMessage 消息, Linux 下该参数保留
 [in]hWnd 接收异常消息的窗口句柄, Linux 下该参数保留

[in]cbExceptionCallBack	接收异常消息的回调函数，回调当前异常的相关信息
[in]pUser	用户数据
typedef void(CALLBACK* fExceptionCallBack)(DWORD dwType, LONG IUserID, LONG IHandle, void *pUser)	
[out]dwType	异常或重连等消息的类型，详见表 5.3
[out]IUserID	登录 ID
[out]IHandle	出现异常的相应类型的句柄
[out]pUser	用户数据

表 5.3 异常消息类型

dwType 宏定义	宏定义值	含义
EXCEPTION_EXCHANGE	0x8000	用户交互时异常（注册心跳超时，心跳间隔为 2 分钟）
EXCEPTION_AUDIOEXCHANGE	0x8001	语音对讲异常
EXCEPTION_ALARM	0x8002	报警异常
EXCEPTION_PREVIEW	0x8003	网络预览异常
EXCEPTION_SERIAL	0x8004	透明通道异常
EXCEPTION_RECONNECT	0x8005	预览时重连
EXCEPTION_ALARMRECONNECT	0x8006	报警时重连
EXCEPTION_SERIALRECONNECT	0x8007	透明通道重连
SERIAL_RECONNECTSUCCESS	0x8008	透明通道重连成功
EXCEPTION_PLAYBACK	0x8010	回放异常
EXCEPTION_DISKFMT	0x8011	硬盘格式化
EXCEPTION_EMAILTEST	0x8013	邮件测试异常
EXCEPTION_BACKUP	0x8014	备份异常
PREVIEW_RECONNECTSUCCESS	0x8015	预览时重连成功
ALARM_RECONNECTSUCCESS	0x8016	报警时重连成功
RESUME_EXCHANGE	0x8017	用户交互恢复
NETWORK_FLOWTEST_EXCEPTION	0x8018	网络流量检测异常

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

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

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

示例代码：

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

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

[返回目录](#)

获取错误信息

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

函 数: DWORD NET_DVR_GetLastError()

参 数:

返回值: 返回最后操作的错误码。详见[错误码宏定义](#)

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

[返回目录](#)

5.2.14 返回最后操作的错误码信息 **NET_DVR_GetErrorMsg**

函 数: char* NET_DVR_GetErrorMsg(LONG *pErrorNo)

参 数: [out]pErrorNo 错误码数值的指针

返回值: 返回值为错误码信息的指针。错误码主要分为网络通讯库错误码、RTSP 通讯库错误码和软硬解库错误码。详见[错误码宏定义](#)

说 明:

[返回目录](#)

5.3 用户注册

5.3.1 激活设备 **NET_DVR_ActivateDevice**

函 数: BOOL NET_DVR_ActivateDevice(char* sDVRIP, WORD wDVRPort, LPNET_DVR_ACTIVATECFG lpActivateCfg)

参 数: [in]sDVRIP 设备 IP 地址

[in]wDVRPort 设备端口

[in]lpActivateCfg 激活参数，包括激活使用的初始密码

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

说 明: 出厂设备需要先激活，然后再使用激活使用的初始密码登录设备。

[返回目录](#)

5.3.2 IPServer 或者 DDNS 域名解析，获取动态 IP 地址和端口号

NET_DVR_GetDVRIPByResolveSvr_EX

函 数: BOOL NET_DVR_GetDVRIPByResolveSvr_EX (char* sServerIP, WORD wServerPort, BYTE* sDVRName, WORD wDVRNameLen, BYTE* sDVRSerialNumber, WORD wDVRSerialLen, char* sGetIP, DWORD* dwPort)

参 数: [in]sServerIP 解析服务器的 IP 地址

[in]wServerPort 解析服务器的端口号，IP Server 解析服务器端口号为 7071，HiDDNS 服务器的端口号为 80

[in]sDVRName 设备名称

[in]wDVRNameLen 设备名称的长度

[in]sDVRSerialNumber 设备的序列号

[in]wDVRSerialLen 设备序列号的长度

[out]sGetIP 获取到的设备 IP 地址指针

[out]dwPort 获取到的设备端口号指针

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

说 明: 该接口中的设备名称和设备序列号不能同时为空。通过设备域名或者序列号解析出设备当前 IP

地址和端口，然后调用 [NET_DVR_Login_V40](#) 登录设备。

支持的解析服务器有 IPServer 和 hiDDNS。

[返回目录](#)

5.3.3 用户注册设备 **NET_DVR_Login_V40**

- 函 数: LONG NET_DVR_Login_V40(LPNET_DVR_USER_LOGIN_INFO pLoginInfo, LPNET_DVR_DEVICEINFO_V40 lpDeviceInfo)
- 参 数: [in]pLoginInfo 登录参数, 包括设备地址、登录用户、密码等
[out]lpDeviceInfo 设备信息(同步登录即 pLoginInfo 中 bUseAsynLogin 为 0 时有效)
- 返回值: 异步登录的状态、用户 ID 和设备信息通过 NET_DVR_USER_LOGIN_INFO 结构体中设置的回调函数(fLoginResultCallBack)返回。对于同步登录, 接口返回-1 表示登录失败, 其他值表示返回的用户 ID 值。用户 ID 具有唯一性, 后续对设备的操作都需要通过此 ID 实现。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
- 说 明:
- pLoginInfo 中 bUseAsynLogin 为 0 时登录为同步模式, 接口返回成功即表示登录成功; pLoginInfo 中 bUseAsynLogin 为 1 时登录为异步模式, 登录是否成功在输入参数设置的回调函数中返回。
 - 设备同时最多允许 128 个用户注册。
 - SDK 支持 2048 个注册, 返回 UserID 的取值范围为 0~2047。

[返回目录](#)

5.3.4 用户注销 **NET_DVR_Logout**

- 函 数: BOOL NET_DVR_Logout(LONG lUserID)
- 参 数: [in]lUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
- 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
- 说 明:

[返回目录](#)

5.4 获取设备能力集

5.4.1 获取设备能力集 **NET_DVR_GetDeviceAbility**

- 函 数: BOOL NET_DVR_GetDeviceAbility(LONG lUserID, DWORD dwAbilityType, char* pInBuf, DWORD dwInLength, char* pOutBuf, DWORD dwOutLength)
- 参 数: [in]lUserID NET_DVR_Login_V40 的返回值
[in]dwAbilityType 能力类型, 详见表 5.4
[in] pInBuf 输入缓冲区指针 (按照设备规定的能力参数的描述方式组合, XML 文本或结构体形式, 详见表 5.5)
[in]dwInLength 输入缓冲区的长度
[out]pOutBuf 输出缓冲区指针 (按照设备规定的能力集的描述方式, XML 文本或结构体形式, 详见表 5.5)
[in]dwOutLength 接收数据的缓冲区的长度

表 5.4 设备能力集类型

dwAbilityType 宏定义	宏定义值	含义
DEVICE_SOFTWARE_ABILITY	0x001	设备软硬件能力
DEVICE_ENCODE_ALL_ABILITY_V20	0x008	设备所有编码能力
IPC_FRONT_PARAMETER_V20	0x009	设备前端参数
DEVICE_USER_ABILITY	0x00c	设备用户管理参数能力
DEVICE_NETAPP_ABILITY	0x00d	设备网络应用参数能力
DEVICE_VIDEOPIC_ABILITY	0x00e	设备图像参数能力
DEVICE_JPEG_CAP_ABILITY	0x00f	设备 JPEG 抓图能力
DEVICE_SERIAL_ABILITY	0x010	设备 RS232 和 RS485 串口能力
DEVICE_ABILITY_INFO	0x011	设备通用能力类型，具体能力根据发送的能力节点来区分

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

说明： 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 5.5 所示。

表 5.5 设备能力集描述

能力类型宏定义	能力类型说明	pinBuf	pOutBuf
DEVICE_SOFTWARE_ABILITY	获取设备软硬件能力	无	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	编码能力获取输入描述	设备所有编码能力 XML 描述
IPC_FRONT_PARAMETER_V20	获取设备前端参数	前端参数能力获取输入描述	设备前端参数 XML 描述
DEVICE_USER_ABILITY	获取设备用户管理参数能力	用户管理参数能力获取输入描述	设备用户管理参数能力 XML 描述
DEVICE_NETAPP_ABILITY	获取设备网络应用参数能力	网络应用参数能力获取输入描述	设备网络应用参数能力 XML 描述
DEVICE_VIDEOPIC_ABILITY	获取设备图像参数能力	图像参数能力获取输入描述	设备图像参数能力 XML 描述
DEVICE_JPEG_CAP_ABILITY	获取设备 JPEG 抓图能力	JPEG 抓图能力获取输入描述	设备 JPEG 抓图能力 XML 描述
DEVICE_SERIAL_ABILITY	获取设备 RS232 和 RS485 串口能力	串口能力获取输入描述	设备串口能力 XML 描述
DEVICE_ABILITY_INFO	设备通用能力类型，具体能力根据发送的能力节点来区分	获取 PTZ 能力集	PTZ 能力 XML 描述(PTZAbility)
		获取报警事件处理能力集	报警事件处理能力 XML 描述(EventAbility)
		获取录像相关能力集	录像相关能力 XML 描述(RecordAbility)
		获取协议接入能力集	设备协议接入能力 XML 描述(AccessProtocolAbility)
		获取安全认证配置能力集	安全认证配置能力 XML 描述(SecurityAbility)
		获取智能通道控制能力集	智能通道控制能力 XML 描述(VcaCtrlAbility)
		获取智能通道分析能力集	智能通道分析能力 XML 描述

			(VcaChanAbility)
--	--	--	------------------

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

[返回目录](#)

5.4.2 获取设备能力集 **NET_DVR_GetSTDAbility**

函 数： BOOL NET_DVR_GetSTDAbility(LONG IUserID, DWORD dwAbilityType, LPNET_DVR_STD_ABILITY lpAbilityParam)

参 数： [in]IUserID 用户 ID 号，NET_DVR_Login_V40 的返回值
[in]dwAbilityType 能力类型，具体定义见表 5.6
[in&out]lpAbilityParam 设备能力集参数（包括输入和输出参数），不同的能力集对应不同的输入输出参数

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

说 明：

表 5.6 标准协议能力集描述

宏定义	宏定义值	含义	lpCondBuffer	lpOutBuffer
NET_DVR_GET_THERMAL_CAPABILITIES	3634	获取热成像能力	NULL	ThermalCap
NET_DVR_GET_FIREDETECTION_CAPABILITIES	3635	获取火点检测配置能力集	4 个字节（DWORD）通道号	FireDetection
NET_DVR_GET_LOWPOWER_CAPABILITIES	3731	获取低功耗配置能力集	4 个字节（DWORD）通道号	LowPower
NET_DVR_GET_ZOOMLINKAGE_CAPABILITIES	3731	获取变倍联动配置能力集	4 个字节（DWORD）通道号	ZoomLinkage
NET_DVR_GET_EVENT_TRIGGERS_CAPABILITIES	3501	获取事件触发能力集	NULL	EventTriggerCap
NET_DVR_GET_MANUALRANGING_CAPABILITIES	6675	获取手动测距配置能力集	NULL	ManualRanging

[返回目录](#)

5.5 实时预览

5.5.1 实时预览 **NET_DVR_RealPlay_V40**

函 数： LONG NET_DVR_RealPlay_V40(LONG IUserID, LPNET_DVR_PREVIEWINFO lpPreviewInfo, REALDATACALLBACK fRealDataCallBack_V30, void *pUser)

参 数： [in] IUserID NET_DVR_Login_V40 的返回值
[in] lpPreviewInfo 预览参数，包括码流类型、取流协议、通道号、预览窗口句柄等，详见结构体：NET_DVR_PREVIEWINFO
[in] fRealDataCallBack_V30 码流数据回调函数
[in] pUser 用户数据

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

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

表 5.7 码流数据类型

dwDataType 宏定义	宏定义值	含义
NET_DVR_SYSHEAD	1	系统头数据
NET_DVR_STREAMDATA	2	流数据（包括复合流或音视频分开的视频流数据）
NET_DVR_AUDIOSTREAMDATA	3	音频数据

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

- 说明：**
- 该接口预览参数结构中可以设置当前预览操作是否阻塞（通过 bBlocked 参数设置），若设为不阻塞，表示发起与设备的连接就认为连接成功，如果发生码流接收失败、播放失败等情况以预览异常的方式通知上层。在循环播放的时候可以减短停顿的时间，与 NET_DVR_RealPlay 处理一致。若设为阻塞，表示直到播放操作完成才返回成功与否。
 - 该接口中的回调函数可以置为空，这样该函数将不回调码流数据给用户，不过用户仍可以通过接口 [NET_DVR_SetRealDataCallBack](#) 或 [NET_DVR_SetStandardDataCallBack](#) 注册捕获码流数据的回调函数以捕获码流数据。
 - fRealDataCallBack_V30 回调函数中不能执行可能会占用时间较长的接口或操作，不建议调用该 SDK（HCNetSDK.dll）本身的接口。
 - Linux 64 位系统不支持软解码功能，因此需要将窗口句柄传 NULL，设置回调函数，只取流不解码显示。
 - 客户端异常离线时，设备端对取流连接的保持时间为 10 秒。

[返回目录](#)

5.5.2 停止预览 NET_DVR_StopRealPlay

函数：LONG NET_DVR_StopRealPlay (LONG IRealHandle)

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

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

说明：

[返回目录](#)

5.5.3 获取预览时用来解码和显示的播放库句柄 NET_DVR_GetRealPlayerIndex

函数：int NET_DVR_GetRealPlayerIndex(LONG IRealHandle)

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

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

说明：用户可以通过返回的句柄自行实现播放库 SDK 提供的其他功能，详见本公司提供的软解码库函数说明《播放器 SDK 编程指南》。例如使用 PlayM4_GetBMP(LONG nPort,.....)、

PlayM4_GetJPEG(LONG nPort,.....)这两个接口时，即可实现将当前预览图像以 BMP 或 JPEG 格式抓图保存到内存中： PlayM4_GetBMP(NET_DVR_GetRealPlayerIndex(),.....)
PlayM4_GetJPEG(NET_DVR_GetRealPlayerIndex(),.....)

[返回目录](#)

5.6 强制 I 帧和刷新帧

5.6.1 强制 I 帧 **NET_DVR_RemoteControl**

函 数： BOOL NET_DVR_RemoteControl(LONG lUserID, DWORD dwCommand, LPVOID lpInBuffer, DWORD dwInBufferSize)

参 数： [in]lUserID 用户 ID 号，NET_DVR_Login_V40 的返回值
[in]dwCommand 控制命令，详见表 5.8
[in]lpInBuffer 输入参数，具体内容跟控制命令相关，详见表 5.8
[in]dwInBufferSize 输入参数长度

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

说 明： 不同的控制功能对应不同的命令号，同时 lpInBuffer 对应不同的结构体，如表 5.8 所示。
NET_DVR_I_FRAME 结构体里面指定强制的通道号、码流类型（主子码流或者其他）。

表 5.8 远程控制命令

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer 对应结构体
NET_DVR_MAKE_I_FRAME	3402	强制 I 帧	NET_DVR_I_FRAME

[返回目录](#)

5.7 预览显示视频参数配置

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

函 数： BOOL NET_DVR_ClientGetVideoEffect(LONG lRealHandle,DWORD *pBrightValue, DWORD *pContrastValue,DWORD *pSaturationValue,DWORD *pHueValue)

参 数： [in]lRealHandle NET_DVR_RealPlay_V40 的返回值
[out]pBrightValue 亮度指针，取值范围[1,10]
[out]pContrastValue 对比度指针，取值范围[1,10]
[out]pSaturationValue 饱和度指针，取值范围[1,10]
[out]pHueValue 色度指针，取值范围[1,10]

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： 需要预览才能获取视频参数。

[返回目录](#)

5.7.2 获取预览视频显示参数 **NET_DVR_GetVideoEffect**

函 数: BOOL NET_DVR_GetVideoEffect(LONG IUserID, LONG IChannel,DWORD *pBrightValue, DWORD *pContrastValue,DWORD *pSaturationValue,DWORD *pHueValue)

参 数: [in]IRealHandle NET_DVR_Login_V40 的返回值
[in]IChannel 通道号
[out] pBrightValue 亮度指针, 取值范围[1,10]
[out] pContrastValue 对比度指针, 取值范围[1,10]
[out] pSaturationValue 饱和度指针, 取值范围[1,10]
[out] pHueValue 色度指针, 取值范围[1,10]

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: 登录设备获取通道的视频参数。

[返回目录](#)

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

函 数: BOOL NET_DVR_ClientSetVideoEffect(LONG IRealHandle,DWORD pBrightValue, DWORD pContrastValue,DWORD pSaturationValue,DWORD pHueValue)

参 数: [in]IRealHandle NET_DVR_RealPlay_V40 的返回值
[in]dwBrightValue 亮度, 取值范围[1,10]
[in]dwContrastValue 对比度, 取值范围[1,10]
[in]dwSaturationValue 饱和度, 取值范围[1,10]
[in]dwHueValue 色度, 取值范围[1,10]

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: 需要预览才能设置视频参数。

[返回目录](#)

5.7.4 设置预览视频显示参数 **NET_DVR_SetVideoEffect**

函 数: BOOL NET_DVR_SetVideoEffect(LONG IUserID, LONG IChannel,DWORD *pBrightValue, DWORD *pContrastValue,DWORD *pSaturationValue,DWORD *pHueValue)

参 数: [in]IRealHandle NET_DVR_Login_V40 的返回值
[in]IChannel 通道号
[in]dwBrightValue 亮度, 取值范围[1,10]
[in]dwContrastValue 对比度, 取值范围[1,10]
[in]dwSaturationValue 饱和度, 取值范围[1,10]
[in]dwHueValue 色度, 取值范围[1,10]

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: 登录设备设置通道的视频参数。

[返回目录](#)

5.8 预览画面叠加字符和图像

5.8.1 预览画面叠加字符和图像，Linux 下无此接口 **NET_DVR_RegisterDrawFun**

函 数: BOOL NET_DVR_RegisterDrawFun(LONG lRealHandle, fDrawFun cbDrawFun, DWORD dwUser)

参 数: [in]lRealHandle NET_DVR_RealPlay_V40 的返回值

[in]cbDrawFun 画图回调函数

[in]dwUser 用户数据

typedef void(CALLBACK *fDrawFun)(LONG lRealHandle, HDC hDc, DWORD dwUser)

[out]lRealHandle 当前的预览句柄

[out]hDc 画图 DC

[out]dwUser 用户数据

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: 该接口主要完成注册回调函数，获得当前表面的 device context。用户可以在这个 DC 上画图或写字，就好像在窗口的客户区 DC 上绘图，但这个 DC 不是窗口客户区的 DC，而是播放器 DirectDraw 里的 Off-Screen 表面的 DC。

[返回目录](#)

5.9 预览时播放声音控制

5.9.1 设置声音播放模式 **NET_DVR_SetAudioMode**

函 数: BOOL NET_DVR_SetAudioMode(DWORD dwMode)

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

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

说 明: 不调用该接口设置声音播放模式，默认为独占播放。

[返回目录](#)

5.9.2 独占声卡模式下开启声音 **NET_DVR_OpenSound**

函 数: BOOL NET_DVR_OpenSound(LONG lRealHandle)

参 数: [in]lRealHandle NET_DVR_RealPlay_V40 的返回值

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

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

[返回目录](#)

5.9.3 独占声卡模式下开启声音 **NET_DVR_CloseSound**

函 数: BOOL NET_DVR_CloseSound()

参 数: 无

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

说 明:

[返回目录](#)

5.9.4 共享声卡模式下开启声音 **NET_DVR_OpenSoundShare**

函 数: BOOL NET_DVR_OpenSoundShare(LONG lRealHandle)

参 数: [in]lRealHandle NET_DVR_RealPlay_V40 的返回值

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

说 明:

[返回目录](#)

5.9.5 共享声卡模式下关闭声音 **NET_DVR_CloseSoundShare**

函 数: BOOL NET_DVR_CloseSoundShare (LONG lRealHandle)

参 数: [in]lRealHandle NET_DVR_RealPlay_V40 的返回值

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

说 明:

[返回目录](#)

5.9.6 调节播放音量 **NET_DVR_Volume**

函 数: BOOL NET_DVR_Volume(LONG lRealHandle,WORD wVolume)

参 数: [in]lRealHandle NET_DVR_RealPlay_V40 的返回值

[in]wVolume 音量, 取值范围[0,0xffff]

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

说 明:

[返回目录](#)

5.10 实时数据回调和录像

5.10.1 注册回调函数，捕获实时码流数据 **NET_DVR_SetRealDataCallBack**

函 数： `BOOL NET_DVR_SetRealDataCallBack(LONG lRealHandle, fRealDataCallBack cbRealDataCallBack, DWORD dwUser)`

参 数： `[in]lRealHandle` 预览句柄，NET_DVR_RealPlay_V40 的返回值
`[in]cbRealDataCallBack` 码流数据回调函数
`[in]dwUser` 用户数据

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

`[out]lRealHandle` 当前的预览句柄
`[out]dwDataType` 数据类型，详见表 5.9
`[out]pBuffer` 存放数据的缓冲区指针
`[out]dwBufSize` 缓冲区大小
`[out]dwUser` 用户数据

表 5.9 码流数据类型

dwDataType 宏定义	宏定义值	含义
NET_DVR_SYSHEAD	1	系统头数据
NET_DVR_STREAMDATA	2	流数据（包括复合流或音视频分开的视频流数据）
NET_DVR_AUDIOSTREAMDATA	3	音频数据

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

说 明：

- 此函数包括开始和停止用户处理 SDK 捕获的数据，当回调函数 `cbRealDataCallBack` 设为非 NULL 值时，表示回调和处理数据；当设为 NULL 时表示停止回调和处理数据。回调的第一个包是 40 个字节的文件头，供后续解码使用，之后回调的是压缩的码流。回调数据最大为 256K 字节。
- `cbRealDataCallBack` 回调函数中不能执行可能会占用时间较长的接口或操作，不建议调用该 SDK（HCNetSDK.dll）本身的接口。

[返回目录](#)

5.10.2 注册回调函数，捕获实时码流数据（标准码流）

NET_DVR_SetStandardDataCallBack

函 数： `BOOL NET_DVR_SetStandardDataCallBack(LONG lRealHandle, fStdDataCallBack cbStdDataCallBack, DWORD dwUser)`

参 数： `[in]lRealHandle` 预览句柄，NET_DVR_RealPlay_V40 的返回值
`[in]cbStdDataCallBack` 标准码流回调函数
`[in]dwUser` 用户数据

```
typedef void(CALLBACK *fStdDataCallBack)(LONG lRealHandle, DWORD dwDataType, BYTE *pBuffer,
DWORD dwBufSize, DWORD dwUser)
```

[out]IRealHandle	当前的预览句柄
[out]dwDataType	数据类型，详见表 5.10
[out]pBuffer	存放数据的缓冲区指针
[out]dwBufSize	缓冲区大小
[out]dwUser	用户数据

表 5.10 标准码流数据类型

dwDataType 宏定义	宏定义值	含义
NET_DVR_SYSHEAD	1	系统头数据
NET_DVR_STD_VIDEODATA	4	标准视频流数据
NET_DVR_STD_AUDIODATA	5	标准音频流数据
NET_DVR_PRIVATE_DATA	2 或者 112	私有数据，包括智能信息叠加等

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

- 说明：**
- 此函数包括开始和停止用户处理 SDK 捕获的数据，当回调函数 cbStdDataCallBack 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，之后回调的是标准码流（含 12 字节的 RTP 头）。
 - cbStdDataCallBack 回调函数中不能执行可能会占用时间较长的接口或操作，不建议调用该 SDK（HCNetSDK.dll）本身的接口。
 - 此函数仅支持对于支持 RTSP 协议取流的设备标准码流回调。**

[返回目录](#)

5.10.3 捕获数据并保存到指定的文件中 **NET_DVR_SaveRealData**

函数：BOOL NET_DVR_SaveRealData(LONG IRealHandle, char *sFileName)

参数：[in]IRealHandle NET_DVR_RealPlay_V40 的返回值
[in]sFileName 文件路径指针，绝对路径，包括文件名

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

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

[返回目录](#)

5.10.4 停止数据捕获 **NET_DVR_StopSaveRealData**

函数：BOOL NET_DVR_StopSaveRealData(LONG IRealHandle)

参数：[in]IRealHandle NET_DVR_RealPlay_V40 的返回值

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

说明：

[返回目录](#)

5.11 预览抓图

5.11.1 设置抓图模式 **NET_DVR_SetCapturePictureMode**

函 数: BOOL NET_DVR_SetCapturePictureMode(DWORD dwCaptureMode)

参 数: [in]dwCaptureMode 抓图模式

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

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

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

[返回目录](#)

5.11.2 预览时, 单帧数据捕获并保存成图片 **NET_DVR_CapturePicture**

函 数: BOOL NET_DVR_CapturePicture(LONG lRealHandle, char *sPicFileName)

参 数: [in]lRealHandle NET_DVR_RealPlay_V40 的返回值
[in]sPicFileName 保存图象的文件路径。路径长度和操作系统有关, sdk 不做限制, windows 默认路径长度小于等于 256 字节 (包括文件名在内)。

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

说 明:

- 在调用该接口之前可以调用 NET_DVR_SetCapturePictureMode 设置抓图模式, 默认为 BMP 模式。如果抓图模式为 BMP 模式, 抓取的是 BMP 图片, 保存路径后缀应为.bmp; 如果抓图模式为 JPEG 模式, 抓取的是 JPEG 图片, 保存路径后缀应为.jpg。
- 若设备的当前分辨率为 2CIF, 播放库做了相关处理, 抓取的图像为 4CIF。
- 调用 NET_DVR_CapturePicture 进行抓图, 要求在调用 NET_DVR_RealPlay_V40 等接口时传入非空的播放句柄 (播放库解码显示), 否则时接口会返回失败, 调用次序错误。

[返回目录](#)

5.12 设备抓图

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

函 数: BOOL NET_DVR_CaptureJPEGPicture(LONG lUserID, LONG lChannel, LPNET_DVR_JPEGPARA lpJpegPara, char *sPicFileName)

参 数: [in]lUserID NET_DVR_Login_V40 的返回值
[in]lChannel 通道号
[in]lpJpegPara JPEG 图像参数, 包括抓图分辨率、抓图质量
[in]sPicFileName 保存 JPEG 图的文件路径, 路径长度和操作系统有关, sdk 不做限制, windows 默认路径长度小于等于 256 字节 (包括文件名在内)。

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

过错误码判断出错原因。

说明： 该接口用于设备的单帧数据捕获，并保存成 JPEG 图片。抓图分辨率需要设备支持，IPC 设备支持当前视频分辨率的抓取。

[返回目录](#)

5.12.2 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： BOOL NET_DVR_CaptureJPEGPicture_NEW(LONG lUserID, LONG lChannel, LPNET_DVR_JPEGPARA lpJpegPara, char *sJpegPicBuffer, DWORD dwPicSize, LPDWORD lpSizeReturned)

参数：

[in]lUserID	NET_DVR_Login_V40 的返回值
[in]lChannel	通道号
[in]lpJpegPara	JPEG 图像参数，包括抓图分辨率、抓图质量
[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： 该接口用于设备的单帧数据捕获，并保存成 JPEG 图片。抓图分辨率需要设备支持，IPC 设备支持当前视频分辨率的抓取。

[返回目录](#)

5.13 云台控制

云台控制操作

5.13.1 云台控制操作（需先启动图像预览）NET_DVR_PTZControl

函数： BOOL NET_DVR_PTZControl(LONG lRealHandle, DWORD dwPTZCommand, DWORD dwStop)

参数：

[in]lRealHandle	NET_DVR_RealPlay_V40 的返回值
[in]dwPTZCommand	云台控制命令，详见表 5.11
[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： 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 5.11 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
TILT_DOWN_ZOOM_IN	58	云台下俯和焦距变大(倍率变大)
TILT_DOWN_ZOOM_OUT	59	云台下俯和焦距变小(倍率变小)
PAN_LEFT_ZOOM_IN	60	云台左转和焦距变大(倍率变大)
PAN_LEFT_ZOOM_OUT	61	云台左转和焦距变小(倍率变小)
PAN_RIGHT_ZOOM_IN	62	云台右转和焦距变大(倍率变大)
PAN_RIGHT_ZOOM_OUT	63	云台右转和焦距变小(倍率变小)
UP_LEFT_ZOOM_IN	64	云台上仰和左转和焦距变大(倍率变大)
UP_LEFT_ZOOM_OUT	65	云台上仰和左转和焦距变小(倍率变小)
UP_RIGHT_ZOOM_IN	66	云台上仰和右转和焦距变大(倍率变大)
UP_RIGHT_ZOOM_OUT	67	云台上仰和右转和焦距变小(倍率变小)
DOWN_LEFT_ZOOM_IN	68	云台下俯和左转和焦距变大(倍率变大)
DOWN_LEFT_ZOOM_OUT	69	云台下俯和左转和焦距变小(倍率变小)

DOWN_RIGHT_ZOOM_IN	70	云台下俯和右转和焦距变大(倍率变大)
DOWN_RIGHT_ZOOM_OUT	71	云台下俯和右转和焦距变小(倍率变小)
TILT_UP_ZOOM_IN	72	云台上仰和焦距变大(倍率变大)
TILT_UP_ZOOM_OUT	73	云台上仰和焦距变小(倍率变小)

[返回目录](#)

5.13.2 云台控制操作（不用启动图像预览） **NET_DVR_PTZControl_Other**

函 数： BOOL NET_DVR_PTZControl_Other(LONG IUserID, LONG IChannel, DWORD dwPTZCommand, DWORD dwStop)

参 数： [in]IUserID NET_DVR_Login_V40 的返回值
[in]IChannel 通道号
[in]dwPTZCommand 云台控制命令，详见表 5.11
[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： 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](#)

5.13.3 带速度的云台控制操作（需先启动图像预览）

NET_DVR_PTZControlWithSpeed

函 数： BOOL NET_DVR_PTZControlWithSpeed(LONG IRealHandle, DWORD dwPTZCommand, DWORD dwStop, DWORD dwSpeed)

参 数： [in]IRealHandle NET_DVR_RealPlay_V40 的返回值
[in] dwPTZCommand 云台控制命令，详见表 5.11
[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。

[返回目录](#)

5.13.4 带速度的云台控制操作（不用启动图像预览）

NET_DVR_PTZControlWithSpeed_Other

函 数： BOOL NET_DVR_PTZControlWithSpeed(LONG IUserID, LONG IChannel, DWORD dwPTZCommand,

DWORD dwStop, DWORD dwSpeed)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IChannel 通道号
[in]dwPTZCommand 云台控制命令, 详见表 5.11
[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: 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

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

函 数: BOOL NET_DVR_PTZPreset(LONG IRealHandle,DWORD dwPTZPresetCmd,DWORD dwPresetIndex)

参 数: [in]IRealHandle NET_DVR_RealPlay_V40 的返回值
[in]dwPTZPresetCmd 操作云台预置点命令, 详见表 5.12
[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: 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 5.12 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](#)

5.13.6 云台预置点操作 **NET_DVR_PTZPreset_Other**

函 数: BOOL NET_DVR_PTZPreset_Other(LONG IUserID,LONG IChannel,DWORD dwPTZPresetCmd,DWORD dwPresetIndex))

参 数: [in]IUserID NET_DVR_Login_V40 的返回值

[in]lChannel 通道号

[in]dwPTZPresetCmd 操作云台预置点命令，详见表 5.12

[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： 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

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

函 数： BOOL NET_DVR_PTZCruise(LONG lRealHandle,DWORD dwPTZCruiseCmd,BYTE byCruiseRoute, BYTE byCruisePoint, WORD wInput)

参 数： [in]lRealHandle NET_DVR_RealPlay_V40 的返回值

[in]dwPTZCruiseCmd 操作云台巡航命令，详见表 5.13

[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： 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 5.13 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](#)

5.13.8 云台巡航操作 **NET_DVR_PTZCruise_Other**

- 函 数:** BOOL NET_DVR_PTZCruise_Other(LONG IUserID, LONG IChannel, DWORD dwPTZCruiseCmd, BYTE byCruiseRoute, BYTE byCruisePoint, WORD wInput)
- 参 数:**
- | | |
|---------------------|---|
| [in] IUserID | NET_DVR_Login_V40 的返回值 |
| [in] IChannel | 通道号 |
| [in] dwPTZCruiseCmd | 操作云台巡航命令, 详见表 5.13 |
| [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:** 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](#)

5.13.9 批量巡航路径 **NET_DVR_GetDeviceConfig**

- 函 数:** BOOL NET_DVR_GetDeviceConfig(LONG IUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpOutBuffer, DWORD dwOutBufferSize)
- 参 数:**
- | | |
|----------------------|--|
| [in] IUserID | 用户 ID 号, NET_DVR_Login_V40 的返回值 |
| [in] dwCommand | 设备配置命令, 详见表 5.14 |
| [in] dwCount | 一次要获取配置参数的个数, 0 和 1 都表示 1 个信息, 2 表示 2 个信息, 最大 64 个 |
| [in] lpInBuffer | 配置条件缓冲区, 详见表 5.14 |
| [in] dwInBufferSize | 缓冲区长度 |
| [out] lpStatusList | 错误信息列表, 和要查询的配置一一对应, 例如 lpStatusList[2]就对应 lpInBuffer[2], 由用户分配内存, 每个错误信息为 4 个字节, 参数值: 0 或者 1 表示成功, 其他值为失败对应的错误号 |
| [out] lpOutBuffer | 设备返回的参数内容 (详见表 5.14), 和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 0 值, 对应 lpOutBuffer 的内容就是无效的 |
| [in] dwOutBufferSize | 输出缓冲区大小 |
- 返回值:** TRUE 表示成功, 但不代表每一个配置都成功, 哪一个成功, 对应查看 lpStatusList[n]值; FALSE 表示全部失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
- 说 明:** 该接口是带有发送数据的批量获取配置信息的通用接口。lpInBuffer 指定需要获取的信息, lpOutBuffer 保存获取到的 dwCount 个配置信息。不同的 dwCommand 对应不同的结构体和命令号, 如表 5.14 所示。

表 5.14 参数批量获取命令

dwCommand 宏定义	含义	lpInBuffer 对应结构体	lpOutBuffer 对应结构体	宏定义值
NET_GET_CRUISEPOINT_V40	获取巡航路径配置	dwCount 个 NET_DVR_CRUISEPOINT_COND	dwCount 个 NET_DVR_CRUISEPOINT_V40	1018

[返回目录](#)

云台轨迹操作

5.13.10 云台轨迹操作，需先启动预览 **NET_DVR_PTZTrack**

- 函 数: BOOL NET_DVR_PTZTrack(LONG lRealHandle, DWORD dwPTZTrackCmd)
- 参 数: [in]lRealHandle NET_DVR_RealPlay_V40 的返回值
[in]dwPTZTrackCmd 操作云台巡航命令，详见表 5.15
- 返回值: TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。
- 说 明: 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 5.15 轨迹操作命令

dwPTZTrackCmd 宏定义	宏定义值	含义
STA_MEM_CRUISE	34	开始记录轨迹
STO_MEM_CRUISE	35	停止记录轨迹
RUN_CRUISE	36	开始轨迹

[返回目录](#)

5.13.11 云台轨迹操作 **NET_DVR_PTZTrack_Other**

- 函 数: BOOL NET_DVR_PTZTrack_Other(LONG lUserID, LONG lChannel, DWORD dwPTZTrackCmd)
- 参 数: [in]lUserID NET_DVR_Login_V40 的返回值
[in]lChannel 通道号
[in]dwPTZTrackCmd 操作云台巡航命令，详见表 5.15
- 返回值: TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。
- 说 明: 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](#)

透明云台控制

5.13.12 透明云台操作，需先启动预览 **NET_DVR_TransPTZ**

函 数： BOOL NET_DVR_TransPTZ(LONG lRealHandle, char *pPTZCodeBuf, DWORD dwBufSize)
参 数： [in]lRealHandle NET_DVR_RealPlay_V40 的返回值
 [in]pPTZCodeBuf 存放云台控制码缓冲区的指针
 [in]dwBufSize 云台控制码的长度
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： 使用该接口能直接通过设备将云台控制码信息直接传输给云台设备，而无需配置解码器。

[返回目录](#)

5.13.13 透明云台操作 **NET_DVR_TransPTZ_Other**

函 数： BOOL NET_DVR_TransPTZ(LONG lUserID, LONG lChannel, char *pPTZCodeBuf, DWORD dwBufSize)
参 数： [in]lUserID NET_DVR_Login_V40 的返回值
 [in]lChannel 通道号
 [in]pPTZCodeBuf 存放云台控制码缓冲区的指针
 [in]dwBufSize 云台控制码的长度
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： 使用该接口能直接通过设备将云台控制码信息直接传输给云台设备，而无需配置解码器。

[返回目录](#)

云台区域缩放控制

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

函 数： BOOL NET_DVR_PTZSelZoomIn(LONG lRealHandle, LPNET_DVR_POINT_FRAME pStruPointFrame);
参 数： [in]lRealHandle NET_DVR_RealPlay_V40 的返回值
 [in]pStruPointFrame 云台图像区域位置信息，详见结构体：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： 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）： $xTop = \text{鼠标当前所选区域的左上点的值} * 255 / 352$ 。
缩小条件： $xBottom$ 减去 $xTop$ 的值大于 2。放大条件： $xBottom$ 减去 $xTop$ 的值大于 0，且 $yBottom$ 减去 $yTop$ 的值大于 0。

[返回目录](#)

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

函 数： BOOL NET_DVR_PTZSelZoomIn_EX(LONG lUserID, LONG lChannel, LPNET_DVR_POINT_FRAME pStruPointFrame)

参 数： [in]lUserID NET_DVR_Login_V40 的返回值
[in]lChannel 通道号
[in]pStruPointFrame 云台图像区域位置信息，详见结构体：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： 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：xBottom 减去 xTop 的值大于 2。放大条件：xBottom 减去 xTop 的值大于 0，且 yBottom 减去 yTop 的值大于 0。

[返回目录](#)

5.14 报警布撤防

设置报警等信息上传的回调函数

5.14.1 注册报警消息回调函数 **NET_DVR_SetDVRMessageCallBack_V31**

函 数： BOOL NET_DVR_SetDVRMessageCallBack_V31(MSGCallBack fMessageCallBack, void* pUser)

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

typedef BOOL (CALLBACK* MSGCallBack_V31)(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD dwBufLen, void *pUser)

[out]lCommand 上传的消息类型，详见表 5.16

[out]pAlarmer 报警设备信息，详见结构体：NET_DVR_ALARMER

[out]pAlarmInfo 报警信息，详见表 5.17

[out]dwBufLen 报警信息缓存大小

[out]pUser 用户数据

表 5.16 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 以下版本协议)
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 以上版本协议)
COMM_ALARM_RULE	0x1102	行为分析信息上传
COMM_ALARM_AUDIOEXCEPTION	0x1150	声音报警信息上传
COMM_FIREDETECTION_ALARM	0x4991	火点检测报警上传

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

说明: V31 接口回调函数有返回值, 需要返回 TRUE。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, 其关系见表 5.17。

表 5.17 报警信息结构

消息类型 (ICommand)	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM	设备报警信息(V3.0 以下版本协议)	NET_DVR_ALARMINFO
COMM_ALARM_V30	设备报警信息(V3.0 以上版本协议)	NET_DVR_ALARMINFO_V30
COMM_ALARM_RULE	行为分析信息	NET_VCA_RULE_ALARM
COMM_ALARM_AUDIOEXCEPTION	声音报警信息	NET_DVR_AUDIOEXCEPTION_ALARM
COMM_FIREDETECTION_ALARM	火点检测报警信息	NET_DVR_FIREDETECTION_ALARM

[返回目录](#)

布防撤防

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

函数: LONG NET_DVR_SetupAlarmChan_V41(LONG IUserID, LPNET_DVR_SETUPALARM_PARAM lpSetupParam)

参数: [in] IUserID NET_DVR_Login_V40 的返回值
[in] lpSetupParam 报警布防参数, 详见结构体: NET_DVR_SETUPALARM_PARAM

返回值: -1 表示失败, 其他值作为 NET_DVR_CloseAlarmChan_V30 函数的句柄参数。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 启动布防前, 需要调用注册回调函数的接口 (如 [NET_DVR_SetDVRMessageCallBack_V31](#)) 才能获取到上传的报警等信息。

[返回目录](#)

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

函数: BOOL NET_DVR_CloseAlarmChan_V30(LONG IAlarmHandle)

参数: [in] IAlarmHandle NET_DVR_SetupAlarmChan_V41 的返回值

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

说明:

[返回目录](#)

5.15 监听报警

5.15.1 启动监听, 接收设备主动上传的报警等信息 **NET_DVR_StartListen_V30**

函数: LONG NET_DVR_StartListen_V30(char *sLocalIP, WORD wLocalPort, MSGCallBack DataCallback, void*

pUserData = NULL)

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

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

[out]ICommand 上传的消息类型, 详见表 5.18
 [out]pAlarmer 报警设备信息, 详见结构体: NET_DVR_ALARMER
 [out]pAlarmInfo 报警信息, 详见表 5.19
 [out]dwBufLen 报警信息缓存大小
 [out]pUser 用户数据

表 5.18 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 以上版本协议)
COMM_ALARM_RULE	0x1102	行为分析信息上传
COMM_ALARM_AUDIOEXCEPTION	0x1150	声音报警信息上传

返回值: -1 表示失败, 其他值作为 NET_DVR_CloseAlarmChan_V30 函数的句柄参数。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明: 该接口中回调函数的第一个参数 (ICommand) 和第三个参数 (pAlarmInfo) 是密切关联的, 其关系见表 5.19。

- SDK 最大能支持 512 路监听。
- 要使 PC 能够收到设备主动发过来的报警等信息, 必须将设备的网络配置中的“远程管理主机地址”或者“远程报警主机地址”设置成 PC 机的 IP 地址 (与接口中的 sLocalIP 参数一致), “远程管理主机端口号”或者“远程报警主机端口号”设置成 PC 机的监听端口号 (与接口中的 wLocalPort 参数一致)。
- 该接口中的回调函数优先级高于其他回调函数, 即设置了该接口中的回调函数, 其他回调函数将接收不到报警信息。

表 5.19 报警信息结构

消息类型 (ICommand)	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM	设备报警信息(V3.0 以下版本协议)	NET_DVR_ALARMINFO
COMM_ALARM_V30	设备报警信息(V3.0 以上版本协议)	NET_DVR_ALARMINFO_V30
COMM_ALARM_RULE	行为分析信息	NET_VCA_RULE_ALARM
COMM_ALARM_AUDIOEXCEPTION	声音报警信息	NET_DVR_AUDIOEXCEPTION_ALARM

[返回目录](#)

5.15.2 停止监听 (支持多线程) **NET_DVR_StopListen_V30**

函 数: BOOL NET_DVR_StopListen_V30(LONG IListenHandle)

参 数: [in]IListenHandle 监听句柄, NET_DVR_StartListen_V30 的返回值
 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

5.16 远程参数配置

系统参数配置

5.16.1 获取设备参数 NET_DVR_GetDVRConfig

函 数: BOOL NET_DVR_GetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpOutBuffer, DWORD dwOutBufferSize, LPDWORD lpBytesReturned)
 参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
 [in]dwCommand 设备配置命令, 不同的命令对应不同的配置功能, 详见表 5.20
 [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 5.20
 [out]lpOutBuffer 接收数据的缓冲指针
 [in]dwOutBufferSize 接收数据的缓冲长度(以字节为单位), 不能为 0
 [out]lpBytesReturned 实际收到的数据长度指针, 不能为 NULL
 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
 说 明: 不同的获取功能对应不同的结构体和命令号, 如表 5.20 所示。

表 5.20 获取设备参数

dwCommand 宏定义	dwCommand 含义	通道号	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_DEVICECFG_V40	获取设备参数(扩展)	无效	NET_DVR_DEVICECFG_V40	1100
NET_DVR_GET_TIMECFG	获取时间参数	无效	NET_DVR_TIME	118
NET_DVR_GET_DECODERCFG_V30	获取 RS485 云台解码器参数	通道号	NET_DVR_DECODERCFG_V30	1042
NET_DVR_GET_RS232CFG_V30	获取 RS232 串口参数	无效	NET_DVR_RS232CFG_V30	1036
NET_DVR_GET_EXCEPTIONCFG_V40	获取异常参数	组号, 从 0 开始, 每组 32 种异常	NET_DVR_EXCEPTION_V40	6177
NET_DVR_GET_ZONEANDDST	获取时区和夏时制参数	无效	NET_DVR_ZONEANDDST	128
NET_DVR_GET_DEVSERVER_CFG	获取模块服务配置	无效	NET_DVR_DEVSERVER_CFG	3257
NET_DVR_GET_USERCFG_V40	获取用户参数	组号, 从 0 开始, 每组 32 个用户	NET_DVR_USER_V40	6187
NET_DVR_GET_SECURITY_CFG	获取安全认证配置	无效	NET_DVR_SECURITY_CFG	147

[返回目录](#)

5.16.2 设置设备参数 **NET_DVR_SetDVRConfig**

函 数: BOOL NET_DVR_SetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpInBuffer, DWORD dwInBufferSize)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 设备配置命令, 不同的命令对应不同的配置功能, 详见表 5.21
[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 5.21
[in]lpInBuffer 输入数据的缓冲指针
[in]dwInBufferSize 输入数据的缓冲长度(以字节为单位)

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

说 明: 不同的获取功能对应不同的结构体和命令号, 如表 5.21 所示。

表 5.21 设置设备参数

dwCommand 宏定义	dwCommand 含义	通道号	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_DEVICECFG_V40	设置设备参数(扩展)	无效	NET_DVR_DEVICECFG_V40	1101
NET_DVR_SET_TIMECFG	设置时间参数	无效	NET_DVR_TIME	119
NET_DVR_SET_DECODERCFG_V30	设置 RS485 云台解码器参数	通道号	NET_DVR_DECODERCFG_V30	1043
NET_DVR_SET_RS232CFG_V30	设置 RS232 串口参数	无效	NET_DVR_RS232CFG_V30	1037
NET_DVR_SET_EXCEPTIONCFG_V40	设置异常参数	组号, 从 0 开始, 每组 32 种异常	NET_DVR_EXCEPTION_V40	6178
NET_DVR_SET_ZONEANDDST	设置时区和夏时制参数	无效	NET_DVR_ZONEANDDST	129
NET_DVR_SET_DEVSERVER_CFG	设置模块服务配置	无效	NET_DVR_DEVSERVER_CFG	3258
NET_DVR_SET_USERCFG_V40	设置用户参数	组号, 从 0 开始, 每组 32 个用户	NET_DVR_USER_V40	6188
NET_DVR_SET_SECURITY_CFG	设置安全认证配置	无效	NET_DVR_SECURITY_CFG	148

[返回目录](#)

通道参数配置

5.16.3 获取通道参数 **NET_DVR_GetDVRConfig**

函 数: BOOL NET_DVR_GetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpOutBuffer, DWORD dwOutBufferSize, LPDWORD lpBytesReturned)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 设备配置命令, 不同的命令对应不同的配置功能, 详见表 5.22
[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 5.22
[out]lpOutBuffer 接收数据的缓冲指针
[in]dwOutBufferSize 接收数据的缓冲长度(以字节为单位), 不能为 0
[out]lpBytesReturned 实际收到的数据长度指针, 不能为 NULL

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

说明: 不同的获取功能对应不同的结构体和命令号, 如表 5.22 所示。

- 通道号是指设备视频通道号, 通过注册设备返回的设备信息 (NET_DVR_DEVICEINFO_V30) 获取模拟通道个数 (byChanNum)、模拟通道起始通道号 (byStartChan) 和设备支持的最大 IP 通道数 (byIPChanNum+ byHighDChanNum*256)、数字通道起始通道号 (byStartDChan)。
- 对于热成像设备, 可能有两个通道, 第一个通道为可见光通道, 第二个通道为热成像通道。

表 5.22 获取设备参数

dwCommand 宏定义	dwCommand 含义	通道号	IpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_PICCFG_V40	获取图像参数	通道号	NET_DVR_PICCFG_V40	6179
NET_DVR_GET_COMPRESSCFG_V30	获取压缩参数	通道号	NET_DVR_COMPRESSIONCFG_V30	1040
NET_DVR_GET_RECORDCFG_V40	获取录像参数	通道号	NET_DVR_RECORD_V40	1008
NET_DVR_GET_SHOWSTRING_V30	获取叠加字符参数	通道号	NET_DVR_SHOWSTRING_V30	1030
NET_DVR_GET_CCDPARAMCFG_EX	获取前端参数(扩展)	通道号	NET_DVR_CAMERAPARAMCFG_EX	3368
NET_DVR_GET_ISP_CAMERAPARAMCFG	获取 ISP 前端参数配置	通道号	NET_DVR_ISP_CAMERAPARAMCFG	3255
NET_DVR_GET_AUDIO_INPUT	获取音频输入参数	通道号	NET_DVR_AUDIO_INPUT_PARAM	3201
NET_DVR_GET_AUDIOOUT_VOLUME	获取输出音频大小	通道号	NET_DVR_AUDIOOUT_VOLUME	3237
NET_DVR_GET_LOW_LIGHTCFG	获取快球低照度信息	通道号	NET_DVR_LOW_LIGHT_CFG	3303
NET_DVR_GET_FOCUSMODECFG	获取快球聚焦模式信息	通道号	NET_DVR_FOCUSMODE_CFG	3305
NET_DVR_GET_INFRARECFG	获取快球红外信息	通道号	NET_DVR_INFRARE_CFG	3307
NET_DVR_GET_AEMODECFG	获取快球其他参数信息	通道号	NET_DVR_AEMODECFG	3309
NET_DVR_GET_CORRIDOR_MODE	获取旋转功能配置	通道号	NET_DVR_CORRIDOR_MODE	3354

[返回目录](#)

5.16.4 设置通道参数 **NET_DVR_SetDVRConfig**

函 数: BOOL NET_DVR_SetDVRConfig(LONG UserID, DWORD dwCommand, LONG IChannel, LPVOID lpInBuffer, DWORD dwInBufferSize)

参 数: [in]UserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 设备配置命令, 不同的命令对应不同的配置功能, 详见表 5.23
[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 5.23
[in]lpInBuffer 输入数据的缓冲指针
[in]dwInBufferSize 输入数据的缓冲长度(以字节为单位)

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

说明: 不同的获取功能对应不同的结构体和命令号, 如表 5.23 所示。

- 通道号是指设备视频通道号, 通过注册设备返回的设备信息 (NET_DVR_DEVICEINFO_V30) 获取模拟通道个数 (byChanNum)、模拟通道起始通道号 (byStartChan) 和设备支持的最大 IP 通道数 (byIPChanNum+ byHighDChanNum*256)、数字通道起始通道号 (byStartDChan)。
- 对于热成像设备, 可能有两个通道, 第一个通道为可见光通道, 第二个通道为热成像通道。

表 5.23 设置设备参数

dwCommand 宏定义	dwCommand 含义	通道号	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_PICCFG_V40	设置图像参数	通道号	NET_DVR_PICCFG_V40	6180
NET_DVR_SET_COMPRESSCFG_V30	设置压缩参数	通道号	NET_DVR_COMPRESSIONCFG_V30	1041
NET_DVR_SET_RECORDCFG_V40	设置录像参数	通道号	NET_DVR_RECORD_V40	1009
NET_DVR_SET_SHOWSTRING_V30	设置叠加字符参数	通道号	NET_DVR_SHOWSTRING_V30	1031
NET_DVR_SET_CCDPARAMCFG_EX	设置前端参数(扩展)	通道号	NET_DVR_CAMERAPARAMCFG_EX	3369
NET_DVR_SET_ISP_CAMERAPARAMCFG	设置 ISP 前端参数配置	通道号	NET_DVR_ISP_CAMERAPARAMCFG	3256
NET_DVR_SET_AUDIO_INPUT	设置音频输入参数	通道号	NET_DVR_AUDIO_INPUT_PARAM	3202
NET_DVR_SET_AUDIOOUT_VOLUME	设置输出音频大小	通道号	NET_DVR_AUDIOOUT_VOLUME	3238
NET_DVR_SET_LOW_LIGHTCFG	设置快球低照度参数	通道号	NET_DVR_LOW_LIGHT_CFG	3304
NET_DVR_SET_FOCUSMODECFG	设置快球聚焦模式参数	通道号	NET_DVR_FOCUSMODE_CFG	3306
NET_DVR_SET_INFRARECFG	设置快球红外参数	通道号	NET_DVR_INFRARE_CFG	3308
NET_DVR_SET_AEMODECFG	设置快球其他参数	通道号	NET_DVR_AEMODECFG	3310
NET_DVR_SET_CORRIDOR_MODE	设置旋转功能配置	通道号	NET_DVR_CORRIDOR_MODE	3355

[返回目录](#)

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

- 函 数：** BOOL NET_DVR_GetDeviceConfig(LONG lUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpOutBuffer, DWORD dwOutBufferSize)
- 参 数：**
- [in] lUserID 用户 ID 号，NET_DVR_Login_V40 的返回值
 - [in] dwCommand 设备配置命令，详见表 5.24
 - [in] dwCount 一次要获取配置参数的个数，0 和 1 都表示 1 个信息，2 表示 2 个信息，最大 64 个
 - [in] lpInBuffer 配置条件缓冲区，详见表 5.25
 - [in] dwInBufferSize 缓冲区长度
 - [out] lpStatusList 错误信息列表，和要查询的配置一一对应，例如 lpStatusList[2]就对应 lpInBuffer[2]，由用户分配内存，每个错误信息为 4 个字节，参数值：0 或者 1 表示成功，其他值为失败对应的错误号
 - [out] lpOutBuffer 设备返回的参数内容（详见表 5.25），和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 0 值，对应 lpOutBuffer 的内容就是无效的
 - [in] dwOutBufferSize 输出缓冲区大小
- 返回值：** TRUE 表示成功，但不代表每一个配置都成功，哪一个成功，对应查看 lpStatusList[n]值；FALSE 表示全部失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。
- 说 明：** 该接口是带有发送数据的批量获取配置信息的通用接口。lpInBuffer 指定需要获取的信息，lpOutBuffer 保存获取到的 dwCount 个配置信息。不同的 dwCommand 对应不同的结构体和命令号，如表 5.25 所示。

表 5.24 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 5.25 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](#)

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

- 函 数: BOOL NET_DVR_SetDeviceConfig(LONG lUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpInParamBuffer, DWORD dwInParamBufferSize)
- 参 数:
- [in] lUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
 - [in] dwCommand 设备配置命令, 详见表 5.26
 - [in] dwCount 一次要设置的配置参数个数, 0 和 1 都表示 1 个, 2 表示 2 个, 最大 64 个
 - [in] lpInBuffer 配置条件缓冲区, 详见表 5.27
 - [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 5.27), 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: 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 dwCount 个, lpInParamBuffer 是设置 dwCount 个配置的参数信息。不同的获取功能对应不同的结构体和命令号, 如表 5.27 所示。

表 5.26 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 5.27 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](#)

网络参数配置

5.16.7 获取网络参数 **NET_DVR_GetDVRConfig**

函 数: BOOL NET_DVR_GetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpOutBuffer, DWORD dwOutBufferSize, LPDWORD lpBytesReturned)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
 [in]dwCommand 设备配置命令, 不同的命令对应不同的配置功能, 详见表 5.28
 [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 5.28
 [out]lpOutBuffer 接收数据的缓冲指针
 [in]dwOutBufferSize 接收数据的缓冲长度(以字节为单位), 不能为 0
 [out]lpBytesReturned 实际收到的数据长度指针, 不能为 NULL

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

说 明: 不同的获取功能对应不同的结构体和命令号, 如表 5.28 所示。

表 5.28 获取设备参数

dwCommand 宏定义	dwCommand 含义	通道号	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_NETCFG_V30	获取网络参数	无效	NET_DVR_NETCFG_V30	1000
NET_DVR_GET_NETAPPCFG	获取网络应用参数(NTP/DDNS)	无效	NET_DVR_NETAPPCFG	222
NET_DVR_GET_NTPCFG	获取网络应用参数(NTP)	无效	NET_DVR_NTPPARA	224
NET_DVR_GET_DDNSCFG_V30	获取网络应用参数(DDNS)	无效	NET_DVR_DDNSPARA_V30	1010
NET_DVR_GET_EMAILCFG_V30	获取网络应用参数(EMAIL)	无效	NET_DVR_EMAILCFG_V30	1012
NET_DVR_GET_SNMPCFG	获取 SNMP 参数	无效	NET_DVR_SNMPCFG	1112
NET_DVR_GET_NAT_CFG	获取 NAT 映射参数	无效	NET_DVR_NAT_CFG	6111
NET_DVR_GET_GBT28181_ACCESS_CFG	获取 GBT28181 协议接入配置	无效	NET_DVR_GBT28181_ACCESS_CFG	3249
NET_DVR_GET_EZVIZ_ACCESS_CFG	获取 EZVIZ 接入参数	无效	NET_DVR_EZVIZ_ACCESS_CFG	3398
NET_DVR_GET_CMS_CFG	获取平台参数	无效	NET_DVR_CMS_PARAM	2070

[返回目录](#)

5.16.8 设置网络参数 **NET_DVR_SetDVRConfig**

函 数: BOOL NET_DVR_SetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpInBuffer, DWORD dwInBufferSize)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
 [in]dwCommand 设备配置命令, 不同的命令对应不同的配置功能, 详见表 5.29
 [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 5.29
 [in]lpInBuffer 输入数据的缓冲指针
 [in]dwInBufferSize 输入数据的缓冲长度(以字节为单位)

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

过错误码判断出错原因。

说明： 不同的获取功能对应不同的结构体和命令号，如表 5.29 所示。

表 5.29 设置设备参数

dwCommand 宏定义	dwCommand 含义	通道号	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_NETCFG_V30	设置网络参数	无效	NET_DVR_NETCFG_V30	1001
NET_DVR_SET_NETAPPCFG	设置网络应用参数(NTP/DDNS)	无效	NET_DVR_NETAPPCFG	223
NET_DVR_SET_NTPCFG	设置网络应用参数(NTP)	无效	NET_DVR_NTPPARA	225
NET_DVR_SET_DDNSCFG_V30	设置网络应用参数(DDNS)	无效	NET_DVR_DDNSPARA_V30	1011
NET_DVR_SET_EMAILCFG_V30	设置网络应用参数(EMAIL)	无效	NET_DVR_EMAILCFG_V30	1013
NET_DVR_SET_SNMPCFG	设置 SNMP 参数	无效	NET_DVR_SNMPCFG	1113
NET_DVR_SET_NAT_CFG	设置 NAT 映射参数	无效	NET_DVR_NAT_CFG	6112
NET_DVR_SET_GBT28181_ACCESS_CFG	设置 GBT28181 协议接入配置	无效	NET_DVR_GBT28181_ACCESS_CFG	3250
NET_DVR_SET_EZVIZ_ACCESS_CFG	设置 EZVIZ 接入参数	无效	NET_DVR_EZVIZ_ACCESS_CFG	3399
NET_DVR_SET_CMS_CFG	设置平台参数	无效	NET_DVR_CMS_PARAM	2071

[返回目录](#)

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

函 数： BOOL NET_DVR_GetDeviceConfig(LONG lUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpOutBuffer, DWORD dwOutBufferSize)

参 数：

[in] lUserID	用户 ID 号，NET_DVR_Login_V40 的返回值
[in] dwCommand	设备配置命令，详见表 5.30
[in] dwCount	一次要获取配置参数的个数，0 和 1 都表示 1 个信息，2 表示 2 个信息，最大 64 个
[in] lpInBuffer	配置条件缓冲区，详见表 5.31
[in] dwInBufferSize	缓冲区长度
[out] lpStatusList	错误信息列表，和要查询的配置一一对应，例如 lpStatusList[2]就对应 lpInBuffer[2]，由用户分配内存，每个错误信息为 4 个字节，参数值：0 或者 1 表示成功，其他值为失败对应的错误号
[out] lpOutBuffer	设备返回的参数内容（详见表 5.31），和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 0 值，对应 lpOutBuffer 的内容就是无效的
[in] dwOutBufferSize	输出缓冲区大小

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

说 明： 该接口是带有发送数据的批量获取配置信息的通用接口。lpInBuffer 指定需要获取的信息，lpOutBuffer 保存获取得到的 dwCount 个配置信息。不同的 dwCommand 对应不同的结构体和命令号，如表 5.31 所示。

表 5.30 参数批量获取命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_GBT28181_CHANINFO_CFG	获取 GBT28181 协议接入设备编码通道信息	3251
NET_DVR_GET_GBT28181_ALARMINFO_CFG	获取 GBT28181 协议接入设备报警输入通道信息	3253
NET_DVR_GET_FTPCFG_V40	获取 FTP 信息	6162

表 5.31 批量获取设备参数

NET_DVR_GET_GBT28181_CHANINFO_CFG	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_GET_GBT28181_CHANINFO_CFG
NET_DVR_GET_GBT28181_ALARMINFO_CFG	dwCount 个 NET_DVR_ALARMINFO	dwCount 个 NET_DVR_GET_GBT28181_ALARMINFO_CFG
NET_DVR_GET_FTPCFG_V40	dwCount 个 NET_DVR_FTP_TYPE	dwCount 个 NET_DVR_FTPCFG_V40

[返回目录](#)

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

- 函 数：** BOOL NET_DVR_SetDeviceConfig(LONG lUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpInParamBuffer, DWORD dwInParamBufferSize)
- 参 数：**
- [in] lUserID 用户 ID 号，NET_DVR_Login_V40 的返回值
 - [in] dwCommand 设备配置命令，详见表 5.32
 - [in] dwCount 一次要设置的配置参数个数，0 和 1 都表示 1 个，2 表示 2 个，最大 64 个
 - [in] lpInBuffer 配置条件缓冲区，详见表 5.33
 - [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 5.33），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：** 该接口是带有发送数据的批量设置子设备配置信息的通用接口，lpInBuffer 指定需要设置哪 dwCount 个，lpInParamBuffer 是设置 dwCount 个配置的参数信息。不同的获取功能对应不同的结构体和命令号，如表 5.33 所示。

表 5.32 参数批量设置命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_SET_GBT28181_CHANINFO_CFG	设置 GBT28181 协议接入设备编码通道	3252
NET_DVR_SET_GBT28181_ALARMINFO_CFG	设置 GBT28181 协议接入设备报警输入通道	3254

NET_DVR_SET_FTPCFG_V40	设置 FTP 信息	6163
------------------------	-----------	------

表 5.33 批量设置设备参数

dwCommand	lpInBuffer 对应结构体	lpInParamBuffer 对应结构体
NET_DVR_SET_GBT28181_CHANINFO_CFG	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_GBT28181_CHANINFO_CFG
NET_DVR_SET_GBT28181_ALARMINGCFG	dwCount 个 NET_DVR_ALARMING_INFO	dwCount 个 NET_DVR_GBT28181_ALARMINGCFG
NET_DVR_SET_FTPCFG_V40	dwCount 个 NET_DVR_FTP_TYPE	dwCount 个 NET_DVR_FTPCFG_V40

[返回目录](#)

Smart 参数配置

5.16.11 批量获取配置信息 NET_DVR_GetDeviceConfig

- 函 数：** BOOL NET_DVR_GetDeviceConfig(LONG IUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpOutBuffer, DWORD dwOutBufferSize)
- 参 数：**
- [in] IUserID 用户 ID 号，NET_DVR_Login_V40 的返回值
 - [in] dwCommand 设备配置命令，详见表 5.34
 - [in] dwCount 一次要获取配置参数的个数，0 和 1 都表示 1 个信息，2 表示 2 个信息，最大 64 个
 - [in] lpInBuffer 配置条件缓冲区，详见表 5.35
 - [in] dwInBufferSize 缓冲区长度
 - [out] lpStatusList 错误信息列表，和要查询的配置一一对应，例如 lpStatusList[2]就对应 lpInBuffer[2]，由用户分配内存，每个错误信息为 4 个字节，参数值：0 或者 1 表示成功，其他值为失败对应的错误号
 - [out] lpOutBuffer 设备返回的参数内容（详见表 5.35），和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 0 值，对应 lpOutBuffer 的内容就是无效的
 - [in] dwOutBufferSize 输出缓冲区大小
- 返回值：** TRUE 表示成功，但不代表每一个配置都成功，哪一个成功，对应查看 lpStatusList[n]值；FALSE 表示全部失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。
- 说 明：** 该接口是带有发送数据的批量获取配置信息的通用接口。lpInBuffer 指定需要获取的信息，lpOutBuffer 保存获取到的 dwCount 个配置信息。不同的 dwCommand 对应不同的结构体和命令号，如表 5.35 所示。

表 5.34 参数批量获取命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_AUDIOEXCEPTIONPARAM	获取音频异常配置	3366

表 5.35 批量获取设备参数

dwCommand	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_GET_AUDIOEXCEPTIONPARAM	dwCount 个 NET_DVR_CHANNEL_GROUP	dwCount 个 NET_DVR_AUDIO_EXCEPTION

[返回目录](#)

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

- 函 数：** BOOL NET_DVR_SetDeviceConfig(LONG lUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpInParamBuffer, DWORD dwInParamBufferSize)
- 参 数：**
- [in] lUserID 用户 ID 号，NET_DVR_Login_V40 的返回值
 - [in] dwCommand 设备配置命令，详见表 5.36
 - [in] dwCount 一次要设置的配置参数个数，0 和 1 都表示 1 个，2 表示 2 个，最大 64 个
 - [in] lpInBuffer 配置条件缓冲区，详见表 5.37
 - [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 5.37），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：** 该接口是带有发送数据的批量设置子设备配置信息的通用接口，lpInBuffer 指定需要设置哪 dwCount 个，lpInParamBuffer 是设置 dwCount 个配置的参数信息。不同的获取功能对应不同的结构体和命令号，如表 5.37 所示。

表 5.36 参数批量设置命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_SET_AUDIOEXCEPTIONPARAM	设置音频异常配置	3367

表 5.37 批量设置设备参数

dwCommand	lpInBuffer 对应结构体	lpInParamBuffer 对应结构体
NET_DVR_SET_AUDIOEXCEPTIONPARAM	dwCount 个 NET_DVR_CHANNEL_GROUP	dwCount 个 NET_DVR_AUDIO_EXCEPTION

[返回目录](#)

报警输入输出配置

5.16.13 获取设备参数 **NET_DVR_GetDVRConfig**

函 数: BOOL NET_DVR_GetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpOutBuffer, DWORD dwOutBufferSize, LPDWORD lpBytesReturned)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 设备配置命令, 不同的命令对应不同的配置功能, 详见表 5.38
[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 5.38
[out]lpOutBuffer 接收数据的缓冲指针
[in]dwOutBufferSize 接收数据的缓冲长度(以字节为单位), 不能为 0
[out]lpBytesReturned 实际收到的数据长度指针, 不能为 NULL

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

说 明: 不同的获取功能对应不同的结构体和命令号, 如表 5.38 所示。

表 5.38 获取设备参数

dwCommand 宏定义	dwCommand 含义	通道号	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_DEVICECFG_V40	获取设备参数(扩展)	无效	NET_DVR_DEVICECFG_V40	1100
NET_DVR_GET_TIMECFG	获取时间参数	无效	NET_DVR_TIME	118
NET_DVR_GET_DECODERCFG_V30	获取 RS485 云台解码器参数	通道号	NET_DVR_DECODERCFG_V30	1042
NET_DVR_GET_RS232CFG_V30	获取 RS232 串口参数	无效	NET_DVR_RS232CFG_V30	1036
NET_DVR_GET_ALARMINCFG_V40	获取报警输入参数	报警输入号, 从 0 开始	NET_DVR_ALARMINCFG_V40	6181
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_EXCEPTIONCFG_V40	获取异常参数	组号, 从 0 开始, 每组 32 个用户	NET_DVR_EXCEPTION_V40	6177
NET_DVR_GET_ZONEANDDST	获取时区和夏时制参数	无效	NET_DVR_ZONEANDDST	128
NET_DVR_GET_DEVSERVER_CFG	获取模块服务配置	无效	NET_DVR_DEVSERVER_CFG	3257
NET_DVR_GET_USERCFG_V40	获取用户参数	组号, 从 0 开始, 每组 32 个用户	NET_DVR_USER_V40	6187
NET_DVR_GET_SECURITY_CFG	获取安全认证配置	无效	NET_DVR_SECURITY_CFG	147

[返回目录](#)

5.16.14 设置设备参数 **NET_DVR_SetDVRConfig**

函 数: BOOL NET_DVR_SetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpInBuffer, DWORD dwInBufferSize)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 设备配置命令, 不同的命令对应不同的配置功能, 详见表 5.39
[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 5.39

[in]lpInBuffer 输入数据的缓冲指针

[in]dwInBufferSize 输入数据的缓冲长度(以字节为单位)

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

说明: 不同的获取功能对应不同的结构体和命令号, 如表 5.39 所示。

表 5.39 设置设备参数

dwCommand 宏定义	dwCommand 含义	通道号	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_DEVICECFG_V40	设置设备参数(扩展)	无效	NET_DVR_DEVICECFG_V40	1101
NET_DVR_SET_TIMECFG	设置时间参数	无效	NET_DVR_TIME	119
NET_DVR_SET_DECODERCFG_V30	设置 RS485 云台解码器参数	通道号	NET_DVR_DECODERCFG_V30	1043
NET_DVR_SET_RS232CFG_V30	设置 RS232 串口参数	无效	NET_DVR_RS232CFG_V30	1037
NET_DVR_SET_ALARMINCFG_V40	设置报警输入参数	报警输入号, 从 0 开始	NET_DVR_ALARMINCFG_V40	6182
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_EXCEPTIONCFG_V40	设置异常参数	组号, 从 0 开始, 每组 32 个用户	NET_DVR_EXCEPTION_V40	6178
NET_DVR_SET_ZONEANDDST	设置时区和夏时制参数	无效	NET_DVR_ZONEANDDST	129
NET_DVR_SET_DEVSERVER_CFG	设置模块服务配置	无效	NET_DVR_DEVSERVER_CFG	3258
NET_DVR_SET_USERCFG_V40	设置用户参数	组号, 从 0 开始, 每组 32 个用户	NET_DVR_USER_V40	6188
NET_DVR_SET_SECURITY_CFG	设置安全认证配置	无效	NET_DVR_SECURITY_CFG	148

[返回目录](#)

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

函 数: BOOL NET_DVR_GetAlarmOut_V30(LONG lUserID, LPNET_DVR_ALARMOUTSTATUS_V30 lpAlarmOutState)

参 数: [in]lUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[out]lpAlarmOutState 报警输出状态

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

说明:

[返回目录](#)

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

函 数: BOOL NET_DVR_SetAlarmOut(LONG lUserID, LONG lAlarmOutPort, LONG lAlarmOutStatic)

参 数: [in]lUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]lAlarmOutPort 报警输出号。初始输出号从 0 开始, 0x00ff 表示全部模拟输出, 0xff00 表示全部数字输出。
[in]lAlarmOutStatic 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:

[返回目录](#)

RTSP 协议参数配置

5.16.17 获取 RTSP 协议参数 **NET_DVR_GetRtspConfig**

函 数: BOOL NET_DVR_GetRtspConfig(LONG IUserID, DWORD dwCommand, LPNET_DVR_RTSPCFG lpOutBuffer, DWORD dwOutBufferSize)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 保留, 置为 0
[out]lpOutBuffer 输出缓存
[in]dwOutBufferSize 存放输出数据的缓冲区大小

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

说 明:

[返回目录](#)

5.16.18 设置 RTSP 协议参数 **NET_DVR_SetRtspConfig**

函 数: BOOL NET_DVR_SetRtspConfig(LONG IUserID, DWORD dwCommand, LPNET_DVR_RTSPCFG lpInBuffer, DWORD dwInBufferSize)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 保留, 置为 0
[in]lpInBuffer 输入缓存
[in]dwOutBufferSize 输入缓存的大小, 大小为结构体 NET_DVR_RTSPCFG 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:

[返回目录](#)

5.17 热成像相关功能

热成像参数配置

5.17.1 获取设备的配置信息 **NET_DVR_GetSTDConfig**

函 数: BOOL NET_DVR_GetSTDConfig(LONG IUserID, DWORD dwCommand, LPNET_DVR_STD_CONFIG lpConfigParam)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值

[in]dwCommand 设备配置命令，详见表 5.40

[in&out]IpConfigParam 配置输入输出参数，不同的配置功能对应不同的输入输出参数，详见表 5.40

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: 获取配置参数时，IpConfigParam 结构体中的 IpInBuffer 无效，设为 NULL。对于不同的配置功能（dwCommand），IpConfigParam 中 IpCondBuffer、IpOutBuffer 分别对应不同的内容，具体如表 5.40 所示。

- 热成像摄像机有两个视频通道，第一个通道为可见光通道，第二个通道为热成像通道，热成像相关功能配置只支持在热成像通道上配置。

表 5.40 参数获取配置命令

dwCommand 宏定义	含义	IpCondBuffer	IpOutBuffer	宏定义值
NET_DVR_GET_FIREDETECTION	获取火点检测参数	4 字节(DWORD)通道号	NET_DVR_FIREDETECTION_CFG	3636
NET_DVR_GET_FIREDETECTION_TRIGGER	获取火点检测联动配置	4 字节(DWORD)通道号	NET_DVR_EVENT_TRIGGER	3638
NET_DVR_GET_LOWPOWER	获取低功耗配置	4 字节(DWORD)通道号	NET_DVR_LOWPOWER	3732
NET_DVR_GET_ZOOMLINKAGE	获取变倍联动配置	4 字节(DWORD)通道号	NET_DVR_ZOOMLINKAGE	3735

[返回目录](#)

5.17.2 获取设备的配置信息 **NET_DVR_SetSTDConfig**

函 数: BOOL NET_DVR_SetSTDConfig(LONG lUserID, DWORD dwCommand, LPNET_DVR_STD_CONFIG IpConfigParam)

参 数: [in]lUserID 用户 ID 号，NET_DVR_Login_V40 的返回值

[in]dwCommand 设备配置命令，详见表 5.41

[in&out]IpConfigParam 配置输入输出参数，不同的配置功能对应不同的输入输出参数，详见表 5.41

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

说明: 设置配置参数时，IpConfigParam 结构体里面的 IpOutBuffer 无效，设为 NULL。对于不同的配置功能（dwCommand），IpConfigParam 中的 IpCondBuffer、IpInBuffer 分别对应不同的内容，具体如表 5.41 所示。

- 热成像摄像机有两个视频通道，第一个通道为可见光通道，第二个通道为热成像通道，热成像相关功能配置只支持在热成像通道上配置。

表 5.41 设置参数配置命令

dwCommand 宏定义	含义	IpCondBuffer	IpInBuffer	宏定义值
NET_DVR_SET_FIREDETECTION	设置火点检测参数	4 字节(DWORD)通道号	NET_DVR_FIREDETECTION_CFG	3637
NET_DVR_SET_FIREDETECTION_TRIGGER	设置火点检测联动配置	4 字节(DWORD)通道号	NET_DVR_EVENT_TRIGGER	3639
NET_DVR_SET_LOWPOWER	设置低功耗配置	4 字节(DWORD)通道号	NET_DVR_LOWPOWER	3733
NET_DVR_SET_ZOOMLINKAGE	设置变倍联动配置	4 字节(DWORD)通道号	NET_DVR_ZOOMLINKAGE	3736
NET_DVR_SET_MANUALRANGING	设置手动测距参数	4 字节(DWORD)通道号	NET_DVR_MANUALRANGING_CFG	6677

[返回目录](#)

远程控制

5.17.3 远程控制 **NET_DVR_STDControl**

函 数: BOOL NET_DVR_STDControl(LONG IUserID, DWORD dwCommand, LPNET_DVR_STD_CONTROL lpControlParam)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 控制命令, 详见表 5.42
[in&out]lpControlParam 远程控制输入输出参数, 不同的控制功能对应不同的输入输出参数, 详见表 5.42

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

说 明: 对于不同的配置功能 (dwCommand), lpControlParam 中的 lpCondBuffer 对应不同的内容, 如表 5.42 所示。

- 热成像摄像机有两个视频通道, 第一个通道为可见光通道, 第二个通道为热成像通道, 该功能只支持在热成像通道上设置。
- 设备是否支持火点可见光镜头聚焦变倍能力, 可以通过设备能力集进行判断, 对应热成像能力集(ThermalCap), 相关接口: [NET_DVR_GetSTDAbility](#), 能力集类型: NET_DVR_GET_THERMAL_CAPABILITIES, 节点: <isFireFocusZoomSupport>。

表 5.42 远程控制命令

dwCommand 宏定义	宏定义值	含义	lpCondBuffer 对应结构体
NET_DVR_FIRE_FOCUSZOOM_CTRL	6670	根据火点对可见光镜头进行聚焦变倍	4 字节(DWORD)通道号

[返回目录](#)

手动快门补偿

5.17.4 热成像仪手动快门补偿 **NET_DVR_ShutterCompensation**

函 数: BOOL NET_DVR_ShutterCompensation(LONG IUserID)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值

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

说 明:

[返回目录](#)

坏点校正

5.17.5 热成像仪坏点校正 **NET_DVR_CorrectDeadPixel**

函 数: BOOL NET_DVR_CorrectDeadPixel(LONG IUserID, LONG IChannel,

LPNET_DVR_CORRECT_DEADPIXEL_PARAM lpInParam)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IChannel 通道号
[in]lpInParam 坏点校正参数

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

[返回目录](#)

辅助聚焦控制

5.17.6 控制一键聚焦 **NET_DVR_FocusOnePush**

函 数: BOOL NET_DVR_FocusOnePush(LONG IUserID, LONG IChannel)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]IChannel 通道号

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

说 明:

[返回目录](#)

恢复镜头电机默认位置

5.17.7 恢复镜头电机默认位置 **NET_DVR_ResetLens**

函 数: BOOL NET_DVR_ResetLens(LONG IUserID, LONG IChannel)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]IChannel 通道号

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

说 明:

[返回目录](#)

5.18 录像文件回放、下载、锁定及备份

录像文件的查找

5.18.1 根据文件类型、时间查找设备录像文件 **NET_DVR_FindFile_V40**

函 数: LONG NET_DVR_FindFile_V40(LONG IUserID, LPNET_DVR_FILECOND_V40 pFindCond)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]pFindCond 待查找的文件信息结构, 包括通道号、文件类型、时间等信息, 详见结构体: NET_DVR_FILECOND_V40

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: 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_V40](#) 接口来获取文件信息。

[返回目录](#)

5.18.2 逐个获取查找到的文件信息 NET_DVR_FindNextFile_V40

函 数: LONG NET_DVR_FindNextFile_V40(LONG IFindHandle, LPNET_DVR_FINDDATA_V40 lpFindData)

参 数: [in]IFindHandle 文件查找句柄, NET_DVR_FindFile_V40 的返回值
[in]lpFindData 保存查找到的文件信息的指针, 包括文件名、文件时间、文件类型等, 详见结构体: NET_DVR_FILECOND_V40

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 5.43。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

表 5.43 查找状态信息

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: 在调用该接口获取查找文件之前, 必须先调用 NET_DVR_FindFile_V40 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](#)

5.18.3 关闭文件查找, 释放资源 NET_DVR_FindClose_V30

函 数: BOOL NET_DVR_FindClose_V30(LONG IFindHandle)

参 数: [in]IFindHandle 文件查找句柄, NET_DVR_FindFile_V40 的返回值

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

说 明:

[返回目录](#)

回放录像文件

5.18.4 按文件名回放录像文件 NET_DVR_PlayBackByName

函 数: LONG NET_DVR_PlayBackByName(LONG IUserID, char *sPlayBackFileName, HWND hWnd)

参 数: [in]IUserID NET_DVR_Login_V40 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:

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

Linux 下:

- Linux 64 位系统不支持软解码功能，因此需要将窗口句柄传空，设置回调函数，只取流不解码显示。
- 对于 4.1 或者以上的版本的 SDK，HWND 表示播放窗口的句柄，定义为：

```
typedef unsigned int HWND;
```

如果使用 Qt 进行界面开发，示例如下：

```
NET_DVR_CLIENTINFO tmpclientinfo;
```

```
QWidget m_framePlayWnd;
```

```
tmpclientinfo.hPlayWnd = (HWND)m_framePlayWnd.GetPlayWndId();
```

- 对于 4.1 以前的版本的 SDK，HWND 定义如下：

```
typedef struct __PLAYRECT
```

```
{
```

```
    int x;                      //显示框左上角横坐标
```

```
    int y;                      //显示框左上角纵坐标
```

```
    int uWidth;                //显示框宽度
```

```
    int uHeight;               //显示框高度
```

```
}PLAYRECT;
```

```
typedef PLAYRECT HWND;
```

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

[返回目录](#)

5.18.5 按时间回放录像文件 NET_DVR_PlayBackByTime_V40

函 数: LONG NET_DVR_PlayBackByTime_V40(LONG lUserID, LPNET_DVR_VOD_PARA pVodPara)

参 数: [in]lUserID NET_DVR_Login_V40 的返回值

[in]pVodPara 回放参数，包括回放的通道、起止时间、窗口句柄等，详见结构体：NET_DVR_VOD_PARA

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

说 明:

- 该接口指定了当前要播放的录像文件，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。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。

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

[返回目录](#)

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

函 数: `BOOL NET_DVR_PlayBackControl_V40(LONG lPlayHandle, DWORD dwControlCode, LPVOID lpInBuffer, DWORD dwInLen, LPVOID lpOutBuffer, DWORD *lpOutLen)`

参 数: `[in]lPlayHandle` 播放句柄，`NET_DVR_PlayBackByName` 或 `NET_DVR_PlayBackByTime_V40` 的返回值
`[in]dwControlCode` 控制录像回放状态命令，详见表 5.44
`[in]lpInBuffer` 指向输入参数的指针，详见表 5.45
`[in]dwInLen` 输入参数的长度。未使用，保留。
`[out]lpOutBuffer` 指向输出参数的指针，详见表 5.45
`[out]lpOutLen` 输出参数的长度

表 5.44 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
<code>NET_DVR_PLAYFAST</code>	5	快放
<code>NET_DVR_PLAYSLOW</code>	6	慢放
<code>NET_DVR_PLAYNORMAL</code>	7	正常速度播放（在暂停后调用将恢复暂停前的速度播放）
<code>NET_DVR_PLAYFRAME</code>	8	单帧放（恢复正常回放使用 <code>NET_DVR_PLAYNORMAL</code> 命令）
<code>NET_DVR_PLAYSTARTAUDIO</code>	9	打开声音
<code>NET_DVR_PLAYSTOPAUDIO</code>	10	关闭声音
<code>NET_DVR_PLAYAUDIOVOLUME</code>	11	调节音量，取值范围[0,0xffff]
<code>NET_DVR_PLAYSETPOS</code>	12	改变文件回放的进度
<code>NET_DVR_PLAYGETPOS</code>	13	获取按文件或者按时间回放的进度
<code>NET_DVR_PLAYGETTIME</code>	14	获取当前已经播放的时间(按文件回放的时候有效)
<code>NET_DVR_PLAYGETFRAME</code>	15	获取当前已经播放的帧数(按文件回放的时候有效)
<code>NET_DVR_GETTOTALFRAMES</code>	16	获取当前播放文件总的帧数(按文件回放的时候有效)
<code>NET_DVR_GETTOTALTIME</code>	17	获取当前播放文件总的时间(按文件回放的时候有效)
<code>NET_DVR_SETSPEED</code>	24	设置码流速度
<code>NET_DVR_KEEPAIVE</code>	25	保持与设备的心跳(如果回调阻塞，建议 2 秒发送一次)
<code>NET_DVR_SET_TRANS_TYPE</code>	32	设置转封装类型

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

说 明: 该接口中的第三个参数是否需要输入数值与控制命令有关，详见表 5.45。

- 特别指出，当控制命令是开始播放（即 NET_DVR_PLAYSTART）时，第三个参数的值表示播放当前文件的偏移量，若该值为 0 表示从文件的起始位置播放，若该值不为 0 则表示断点续传的文件位置（Byte）。
- 该接口中的第五个参数表示当前控制命令操作所获取到的相应的参数，控制命令中的 NET_DVR_PLAYGETPOS、NET_DVR_PLAYGETTIME、NET_DVR_PLAYGETFRAME、NET_DVR_GETTOTALFRAMES 和 NET_DVR_GETTOTALTIME 都能通过该参数得到对应的值，详见上表。
- 当命令值为 NET_DVR_PLAYGETPOS 时，获取文件回放或者下载进度时，0-100 表示正常的进度值，大于 100 的值表示回放或者下载异常；**获取按时间回放或下载进度时，只能获取的进度值是 0、100（结束）、200（异常）。**
- NET_DVR_SET_TRANS_TYPE 设置转封装类型需要在开始播放（即 NET_DVR_PLAYSTART）之前调用。

表 5.45 回放控制参数

状态命令宏定义	状态命令说明	IpInBuf	IpOutBuf
NET_DVR_PLAYSTART	开始播放	一个 4 字节整型的偏移量	无
NET_DVR_PLAYSETPOS	改变回放的进度	一个 4 字节整型的进度值（0-100）	无
NET_DVR_PLAYGETPOS	获取回放的进度	无	一个 4 字节整型的进度值（0-100）
NET_DVR_PLAYGETTIME	获取当前已播放的时间（按文件回放有效）	无	一个 4 字节整型的时间值
NET_DVR_PLAYGETFRAME	获取当前已播放的帧数（按文件回放有效）	无	一个 4 字节整型的帧数值
NET_DVR_GETTOTALFRAMES	获取当前播放文件总的帧数（按文件回放有效）	无	一个 4 字节整型的帧数值
NET_DVR_GETTOTALTIME	获取当前播放文件总的时间（按文件回放有效）	无	一个 4 字节整型的时间值
NET_DVR_SETSPEED	设置码流速度	一个 4 字节整型的速度值	无
NET_DVR_SET_TRANS_TYPE	设置转封装类型	一个 4 字节整型的转码类型：1-PS，2-TS，3-RTP	无

[返回目录](#)

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

函 数： BOOL NET_DVR_StopPlayBack(LONG IPlayHandle)

参 数： [in]IPlayHandle 回放句柄，NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime_V40 的返回值

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

说 明：

[返回目录](#)

回放录像文件时的数据捕获

5.18.8 捕获回放的录像数据，并保存成文件 **NET_DVR_PlayBackSaveData**

函 数： BOOL NET_DVR_PlayBackSaveData(LONG IPlayHandle, char *sFileName)

参 数： [in] IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime_V40 的返回值
[in] sFileName 保存数据的文件路径

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

[返回目录](#)

5.18.9 停止保存录像数据 **NET_DVR_StopPlayBackSave**

函 数： BOOL NET_DVR_StopPlayBackSave(LONG IPlayHandle)

参 数： [in] IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime_V40 的返回值

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

说 明：

[返回目录](#)

5.18.10 注册回调函数，捕获录像数据 **NET_DVR_SetPlayDataCallBack_V40**

函 数： BOOL NET_DVR_SetPlayDataCallBack_V40(LONG IPlayHandle, fPlayDataCallBack cbPlayDataCallBack, void *pUser)

参 数： [in] IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime_V40 的返回值
[in] fPlayDataCallBack 录像数据回调函数
[in] pUser 用户数据

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

[out] IPlayHandle 当前的录像播放句柄
[out] dwDataType 数据类型，详见表 5.46
[out] pBuffer 存放数据的缓冲区指针
[out] dwBufSize 缓冲区大小
[out] pUser 用户数据

表 5.46 回放回调数据类型

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

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

说 明：

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

[返回目录](#)

回放的其他操作

5.18.11 获取录像回放时显示的 OSD 时间 **NET_DVR_GetPlayBackOsdTime**

函 数： BOOL NET_DVR_GetPlayBackOsdTime(LONG IPlayHandle, LPNET_DVR_TIME lpOsdTime)

参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或
NET_DVR_PlayBackByTime_V40 的返回值
[out]lpOsdTime 获取的 OSD 时间的指针，详见结构体：NET_DVR_TIME

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

说 明：

[返回目录](#)

5.18.12 录像回放时抓图，并保存在文件中 **NET_DVR_PlayBackCaptureFile**

函 数： BOOL NET_DVR_PlayBackCaptureFile(LONG IPlayHandle, char *sFileName)

参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或
NET_DVR_PlayBackByTime_V40 的返回值
[in]sFileName 保存图片数据的文件路径

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： 回放时抓下来的图片时间要比抓图时间延后，这是因为预览画面上的 OSD 时间是解码完成的显示时间，而解码缓冲区会有将近 1M 左右的数据还没有解出来，要抓取的图片数据是网络缓冲里面的。目前解码库没有直接从解码缓冲区中取出数据的接口。

[返回目录](#)

5.18.13 刷新显示回放窗口 **NET_DVR_RefreshPlay**

函 数： BOOL NET_DVR_RefreshPlay(LONG IPlayHandle)

参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或

NET_DVR_PlayBackByTime_V40 的返回值

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

说明: 当用户暂停或者单帧回放时, 如果刷新了窗口, 则窗口中的图像因为刷新而消失, 此时调用这个接口可以重新显示最后一帧画面。此接口只在暂停和单帧播放时有效。

[返回目录](#)

5.18.14 获取回放时用来解码显示的播放库句柄

NET_DVR_GetPlayBackPlayerIndex

函数: int NET_DVR_GetPlayBackPlayerIndex(LONG IPlayHandle)

参数: [in]IPlayHandle 播放句柄, 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: 用户可以通过返回的句柄自行实现播放库 SDK 提供的其他功能, 详见本公司提供的软解码库函数说明《播放器 SDK 编程指南》。

例如使用 PlayM4_GetBMP(LONG nPort,.....)、PlayM4_GetJPEG(LONG nPort,.....)这两个接口时, 即可实现将当前预览图像以 BMP 或 JPEG 格式抓图保存到内存中:

PlayM4_GetBMP(NET_DVR_GetRealPlayerIndex(),.....)

PlayM4_GetJPEG(NET_DVR_GetRealPlayerIndex(),.....)。

[返回目录](#)

下载录像文件

5.18.15 按文件名下载录像文件 NET_DVR_GetFileByName

函数: LONG NET_DVR_GetFileByName(LONG IUserID,char *sDVRFileName,char *sSavedFileName)

参数: [in]IUserID NET_DVR_Login_V40 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: 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_V40 接口的开始播放控制命令 (NET_DVR_PLAYSTART) 才能实现下载。

[返回目录](#)

5.18.16 按时间下载录像文件 NET_DVR_GetFileByTime_V40

函数: LONG NET_DVR_GetFileByTime_V40(LONG IUserID, char *sSavedFileName, LPNET_DVR_PLAYCOND pDownloadCond)

参数: [in] IUserID 用户 ID, NET_DVR_Login_V40 的返回值

[in] sSavedFileName 下载后保存到 PC 机的文件路径，需为绝对路径，包括文件名
 [in] pDownloadCond 下载条件，包括通道号、起止时间等，详见结构体：
 NET_DVR_PLAYCOND

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

说 明：

- 该接口指定了当前要下载的录像文件，调用成功后，还需要调用 NET_DVR_PlayBackControl_V40 接口的开始播放控制命令（NET_DVR_PLAYSTART）才能实现下载。
- V5.0.3.2 或以后版本，通过该接口保存录像，文件最大限制为 1024MB，大于 1024M 时，SDK 自动新建文件进行保存，文件开始将 40 字节头自动写入，文件名命名规则为“在接口传入的文件名基础上增加数字标识(例如：*_1.mp4、*_2.mp4)”。

[返回目录](#)

5.18.17 控制录像下载的状态 NET_DVR_PlayBackControl_V40

函 数： BOOL NET_DVR_PlayBackControl_V40(LONG lPlayHandle, DWORD dwControlCode, LPVOID
 lpInBuffer, DWORD dwInLen, LPVOID lpOutBuffer, DWORD *lpOutLen)

参 数： [in]lPlayHandle 下载句柄，NET_DVR_GetFileByName 或
 NET_DVR_GetFileByTime_V40 的返回值
 [in]dwControlCode 控制录像下载状态命令，详见表 5.47
 [in] lpInBuffer 指向输入参数的指针，详见表 5.48
 [in]dwInLen 输入参数的长度
 [out]lpOutBuffer 指向输出参数的指针，详见表 5.48
 [out]lpOutLen 输出参数的长度

表 5.47 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_PLAYSETPOS	12	改变文件下载的进度（按文件下载时有效）
NET_DVR_PLAYGETPOS	13	获取文件下载的进度（按文件下载时有效）
NET_DVR_GETTOTALFRAMES	16	获取当前下载文件总的帧数(按文件下载时有效)
NET_DVR_GETTOTALTIME	17	获取当前下载文件总的时间(按文件下载时有效)
NET_DVR_SETSPEED	24	设置下载速度，速度单位：kbps，最小为 256kbps，最大为设备带宽
NET_DVR_SET_TRANS_TYPE	32	设置转封装类型

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

说 明： 该接口中的第三个参数是否需要输入数值与控制命令有关，详见表 5.48。

- 当控制命令是开始下载（即 NET_DVR_PLAYSTART）时，第三个参数的值表示下载当前文件的偏移量，若该值为 0 表示从文件的起始位置下载，若该值不为 0 则表示断点续传的文件位置(Byte)。断点续传需要设备支持。
- 当命令值为 NET_DVR_PLAYGETPOS 时，获取文件回放或者下载进度时，0-100 表示正常的进度

值，大于 100 的值表示回放或者下载异常。

- 按时间下载时，只能获取进度值：0、100（结束）、200（异常）。
- NET_DVR_SET_TRANS_TYPE 设置转封装类型需要在开始播放（即 NET_DVR_PLAYSTART）之前调用。

表 5.48 下载控制参数

状态命令宏定义	状态命令说明	IpInBuf	IpOutBuf
NET_DVR_PLAYSTART	开始下载	一个 4 字节整型的偏移量	无
NET_DVR_PLAYSETPOS	改变下载的进度	一个 4 字节整型的进度值 (0-100)	无
NET_DVR_PLAYGETPOS	获取下载的进度	无	一个 4 字节整型的进度值 (0-100)
NET_DVR_GETTOTALFRAMES	获取当前下载文件总的帧数（按文件回放有效）	无	一个 4 字节整型的帧数值
NET_DVR_GETTOTALTIME	获取当前下载文件总的时间（按文件回放有效）	无	一个 4 字节整型的时间值
NET_DVR_SETSPEED	设置下载速度	一个 4 字节整型的速度值	无
NET_DVR_SET_TRANS_TYPE	设置转封装类型	一个 4 字节整型的转码类型： 1-PS, 2-TS, 3-RTP	无

[返回目录](#)

5.18.18 停止下载录像文件 NET_DVR_StopGetFile

函 数： BOOL NET_DVR_StopGetFile(LONG lFileHandle)

参 数： [in]lFileHandle 下载句柄，NET_DVR_GetFileByName 或
NET_DVR_GetFileByTime_V40 的返回值

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

说 明：

[返回目录](#)

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

函 数： int NET_DVR_GetDownloadPos(LONG lFileHandle)

参 数： [in]lFileHandle 下载句柄，NET_DVR_GetFileByName 或
NET_DVR_GetFileByTime_V40 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；正常范围 0-100，如返回 200 表明出现网络异常。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明： 该接口用于获取按文件名下载录像文件时的下载进度。

[返回目录](#)

录像文件锁定和解锁

5.18.20 按文件名锁定录像文件 **NET_DVR_LockFileByName**

函 数: `BOOL NET_DVR_LockFileByName(LONG IUserID, char *sLockFileName)`

参 数: `[in]IUserID` `NET_DVR_Login_V40` 的返回值
`[in]sLockFileName` 要锁定的录像文件名, 文件名长度需小于 100 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: 在使用该接口锁定录像文件前, 可以先调用录像文件查找的接口获取文件名。当文件被锁定后, 将不会被覆盖。

[返回目录](#)

5.18.21 按文件名解锁录像文件 **NET_DVR_UnlockFileByName**

函 数: `BOOL NET_DVR_UnlockFileByName(LONG IUserID, char *sUnlockFileName)`

参 数: `[in]IUserID` `NET_DVR_Login_V40` 的返回值
`[in]sUnlockFileName` 要解锁的录像文件名

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: 在使用该接口锁定录像文件前, 可以先调用录像文件查找的接口获取文件名。

[返回目录](#)

5.19 图片的查找、回放下载

查找图片

5.19.1 根据类型和时间查找图片 **NET_DVR_FindPicture**

函 数: `LONG NET_DVR_FindPicture(LONG IUserID, NET_DVR_FIND_PICTURE_PARAM* pFindParam)`

参 数: `[in]IUserID` `NET_DVR_Login_V40` 的返回值
`[in]pFindParam` 图片查找条件信息

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

说 明: 该接口指定了要查找的图片的类型和查找时间范围, 调用成功后, 就可以调用 `NET_DVR_FindNextPicture` 接口来获取图片信息。注: 该接口查找的是设备本地的图片, 可通过 `NET_DVR_SetDVRConfig` 配置设备的抓图计划 (`NET_DVR_SCHED_CAPTURECFG`)。

[返回目录](#)

5.19.2 逐个获取查找到的图片 **NET_DVR_FindNextPicture_V40**

函 数: `LONG NET_DVR_FindNextPicture_V40(LONG IFindHandle, LPNET_DVR_FIND_PICTURE_V40`

lpFindData)

- 参 数: [in]IFindHandle 图片查找句柄, NET_DVR_FindPicture 的返回值
[out]lpFindData 保存图片信息的指针
- 返回值: -1 表示失败, 其他值表示当前的获取状态等信息, 如表 5.49 所示。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

表 5.49 查找状态信息

宏定义	宏定义值	含义
NET_DVR_FILE_SUCCESS	1000	获取图片信息成功
NET_DVR_FILE_NOFIND	1001	未查找到图片
NET_DVR_ISFINDING	1002	正在查找请等待
NET_DVR_NOMOREFILE	1003	没有更多的图片, 查找结束
NET_DVR_FILE_EXCEPTION	1004	查找图片时异常

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

[返回目录](#)

5.19.3 关闭图片查找, 释放资源 NET_DVR_CloseFindPicture

- 函 数: BOOL NET_DVR_CloseFindPicture(LONG IFindHandle)
- 参 数: [in]IFindHandle 图片查找句柄, NET_DVR_FindPicture 的返回值
- 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

回放（下载）图片

5.19.4 图片回放 NET_DVR_GetPicture_V30

- 函 数: BOOL NET_DVR_GetPicture_V30(LONG IUserID, char *sDVRFileName, char *sSavedFileBuf, DWORD dwBufLen, DWORD *lpdwRetLen)
- 参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]sDVRFileName 图片名称
[in]sSavedFileName 保存图片的缓冲区
[in]dwBufLen 缓冲区大小
[out]lpdwRetLen 实际收到的数据长度指针, 不能为 NULL
- 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明: 图片为 JPEG 格式, 数据保存在缓冲区 sSavedFileName 中, 回放显示需要应用层实现。

[返回目录](#)

5.20 语音对讲、转发及广播

语音对讲(Windows 32 位系统支持)

5.20.1 启动语音对讲 **NET_DVR_StartVoiceCom_V30**

函 数: LONG NET_DVR_StartVoiceCom_V30(LONG IUserID, DWORD dwVoiceChan, BOOL bNeedCBNoEncData, fVoiceDataCallBack cbVoiceDataCallBack, void* pUser)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
 [in]dwVoiceChan 语音通道号, 从 1 开始
 [in]bNeedCBNoEncData 需要回调的语音数据类型: 0- 编码后的语音数据, 1- 编码前的 PCM 原始数据
 [in]cbVoiceDataCallBack 音频数据回调函数
 [in]pUser 用户数据指针

```
typedef void(CALLBACK *fVoiceDataCallBack)(LONG lVoiceComHandle, char *pRecvDataBuffer,
DWORD dwBufSize, BYTE byAudioFlag, void *pUser)
```

[out]lVoiceComHandle NET_DVR_StartVoiceCom_V30 的返回值
 [out]pRecvDataBuffer 存放音频数据的缓冲区指针
 [out]dwBufSize 音频数据大小
 [out]byAudioFlag 音频数据类型: 0- 本地采集的数据, 1- 设备发送过来的语音数据
 [out]pUser 用户数据指针

返回值: -1 表示失败, 其他值作为 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: Windows 7 操作系统下, 如果不外接音频设备, 该接口将返回失败。在调用开始语音对讲之前可先配置设备的语音对讲音频编码类型, 即可先调用参数配置中的 NET_DVR_COMPRESSION_AUDIO 结构配置。

当前音频为 G722 编码时, 音频数据的采样频率为 16000, 16 位采样且是单通道的。因此, 音频播放格式应如下定义:

```
const int SAMPLES_PER_SECOND = 16000;
const int CHANNEL = 1;
const int BITS_PER_SAMPLE = 16;
WAVEFORMATEX m_wavFormatEx;
m_wavFormatEx.cbSize = sizeof(m_wavFormatEx);
m_wavFormatEx.nBlockAlign = CHANNEL * BITS_PER_SAMPLE / 8;
m_wavFormatEx.nChannels = CHANNEL;
m_wavFormatEx.nSamplesPerSec = SAMPLES_PER_SECOND;
m_wavFormatEx.wBitsPerSample = BITS_PER_SAMPLE;
m_wavFormatEx.nAvgBytesPerSec = SAMPLES_PER_SECOND*m_wavFormatEx.nBlockAlign
```

当前音频为 G711 或者 G726 编码时, 音频数据的采样频率为 8000, 16 位采样且是单通道的。因此, 音频播放格式应如下定义:

```

const int SAMPLES_PER_SECOND_G711_MU = 8000;
const int CHANNEL = 1;
const int BITS_PER_SAMPLE = 16;
WAVEFORMATEX m_wavFormatEx;
m_wavFormatEx.cbSize = sizeof(m_wavFormatEx);
m_wavFormatEx.nBlockAlign = CHANNEL * BITS_PER_SAMPLE / 8;
m_wavFormatEx.nChannels = CHANNEL;
m_wavFormatEx.nSamplesPerSec = SAMPLES_PER_SECOND_G711_MU;
m_wavFormatEx.wBitsPerSample = BITS_PER_SAMPLE;
m_wavFormatEx.nAvgBytesPerSec = SAMPLES_PER_SECOND_G711_MU *
m_wavFormatEx.nBlockAlign;

```

[返回目录](#)

5.20.2 设置语音对讲客户端的音量 **NET_DVR_SetVoiceComClientVolume**

函 数： BOOL NET_DVR_SetVoiceComClientVolume(LONG lVoiceComHandle, WORD wVolume)

参 数： [in]lVoiceComHandle NET_DVR_StartVoiceCom_V30 的返回值
[in]wVolume 设置音量，取值范围[0,0xffff]

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

说 明：

[返回目录](#)

5.20.3 停止语音对讲或者语音转发 **NET_DVR_StopVoiceCom**

函 数： BOOL NET_DVR_StopVoiceCom(LONG lVoiceComHandle)

参 数： [in]lVoiceComHandle NET_DVR_StartVoiceCom_V30 或
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：

[返回目录](#)

语音转发

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

函 数： LONG NET_DVR_StartVoiceCom_MR_V30(LONG lUserID, DWORD dwVoiceChan, fVoiceDataCallBack
cbVoiceDataCallBack, void* pUser)

参 数： [in]lUserID NET_DVR_Login_V40 的返回值
[in]dwVoiceChan 语音通道号，从 1 开始
[in]cbVoiceDataCallBack 音频数据回调函数，得到的数据是编码以后的音频数据，需调用我们提供的音频解码函数（详见音频编解码章节的说明）后可得到 PCM 数据

[in]pUser 用户数据指针

```
typedef void(CALLBACK *fVoiceDataCallBack)(LONG IVoiceComHandle,char *pRecvDataBuffer,
DWORD dwBufSize, BYTE byAudioFlag,void*pUser)
```

[out]IVoiceComHandle NET_DVR_StartVoiceCom_MR_V30 的返回值

[out]pRecvDataBuffer 存放音频数据的缓冲区指针

[out]dwBufSize 音频数据大小

[out]byAudioFlag 音频数据类型：1-设备发送过来的音频数据

[out]pUser 用户数据指针

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：在调用开始语音转发之前可先配置设备的音频编码类型，即可先调用参数配置中的 NET_DVR_COMPRESSION_AUDIO 结构配置。

当前音频为 G722 编码时，音频数据的采样频率为 16000，16 位采样且是单通道的。因此，音频播放格式应如下定义：

```
const int SAMPLES_PER_SECOND = 16000;
const int CHANNEL = 1;
const int BITS_PER_SAMPLE = 16;
WAVEFORMATEX m_wavFormatEx;
m_wavFormatEx.cbSize = sizeof(m_wavFormatEx);
m_wavFormatEx.nBlockAlign = CHANNEL * BITS_PER_SAMPLE / 8;
m_wavFormatEx.nChannels = CHANNEL;
m_wavFormatEx.nSamplesPerSec = SAMPLES_PER_SECOND;
m_wavFormatEx.wBitsPerSample = BITS_PER_SAMPLE;
m_wavFormatEx.nAvgBytesPerSec = SAMPLES_PER_SECOND*m_wavFormatEx.nBlockAlign
```

当前音频为 G711 或者 G726 编码时，音频数据的采样频率为 8000，16 位采样且是单通道的。因此，音频播放格式应如下定义：

```
const int SAMPLES_PER_SECOND_G711_MU = 8000;
const int CHANNEL = 1;
const int BITS_PER_SAMPLE = 16;
WAVEFORMATEX m_wavFormatEx;
m_wavFormatEx.cbSize = sizeof(m_wavFormatEx);
m_wavFormatEx.nBlockAlign = CHANNEL * BITS_PER_SAMPLE / 8;
m_wavFormatEx.nChannels = CHANNEL;
m_wavFormatEx.nSamplesPerSec = SAMPLES_PER_SECOND_G711_MU;
m_wavFormatEx.wBitsPerSample = BITS_PER_SAMPLE;
m_wavFormatEx.nAvgBytesPerSec = SAMPLES_PER_SECOND_G711_MU*
m_wavFormatEx.nBlockAlign;
```

[返回目录](#)

5.20.5 转发语音数据 NET_DVR_VoiceComSendData

函 数：BOOL NET_DVR_VoiceComSendData(LONG IVoiceComHandle, char *pSendBuf, DWORD dwBufSize)

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。当前是 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。

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

说明: 该接口实现将获取到的经过编码后的音频数据转发给设备。

[返回目录](#)

5.20.6 停止语音对讲或语音转发 **NET_DVR_StopVoiceCom**

函数: BOOL NET_DVR_StopVoiceCom (LONG lVoiceComHandle)

参数: [in]lVoiceComHandle NET_DVR_StartVoiceCom_V30 或
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:

[返回目录](#)

语音广播(Windows 32 位系统支持)

5.20.7 启动语音广播的 PC 端声音捕获 **NET_DVR_ClientAudioStart_V30**

函数: BOOL NET_DVR_ClientAudioStart_V30(fVoiceDataCallBack cbVoiceDataCallBack, void *pUser)

参数: [in]cbVoiceDataCallBack 音频数据回调函数
[in]pUser 用户数据

typedef void(CALLBACK *fVoiceDataCallBack)(char *pRecvDataBuffer,DWORD dwBufSize, void *pUser)

[out]pRecvDataBuffer 存放 PC 本地采集的音频数据（PCM）的缓冲区指针

[out]dwBufSize 音频数据大小

[out]pUser 用户数据指针

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

说明: Windows 7 操作系统下，如果不外接音频设备，该接口将返回失败。实现语音广播功能需先调用 NET_DVR_ClientAudioStart_V30 接口采集本地 PC 的音频数据，再调用 NET_DVR_AddDVR 或者 NET_DVR_AddDVR_V30 逐个添加设备同时将采集到的数据发送给设备。

[返回目录](#)

5.20.8 添加设备的某个语音通道到可以接收 PC 端声音的广播组

NET_DVR_AddDVR_V30

函数: LONG NET_DVR_AddDVR_V30(LONG lUserID, DWORD dwVoiceChan)

参数: [in]lUserID NET_DVR_Login_V40 的返回值
[in]dwVoiceChan 语音通道号，从 1 开始

返回值: -1 表示失败, 其他值作为 NET_DVR_DeIDVR_V30 等函数的参数。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明: 实现语音广播功能需先调用 NET_DVR_ClientAudioStart_V30 接口采集本地 PC 的音频数据, 再调用 NET_DVR_AddDVR 或者 NET_DVR_AddDVR_V30 逐个添加设备同时将采集到的数据发送给设备。SDK 最大支持添加 512 个设备。

[返回目录](#)

5.20.9 从可接收 PC 机声音的广播组里删除该设备的语音通道

NET_DVR_DeIDVR_V30

函数: LONG NET_DVR_DeIDVR_V30(LONG IUserID)

参数: [in]IUserID NET_DVR_Login_V40 的返回值

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

说明:

[返回目录](#)

5.20.10 停止语音广播的 PC 端声音捕获 NET_DVR_ClientAudioStop

函数: BOOL NET_DVR_ClientAudioStop()

参数:

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

说明:

[返回目录](#)

音频压缩参数

5.20.11 获取当前生效的对讲音频压缩参数 NET_DVR_GetCurrentAudioCompress

函数: BOOL NET_DVR_GetCurrentAudioCompress(LONG IUserID, LPNET_DVR_COMPRESSION_AUDIO lpCompressAudio)

参数: [in] IUserID 用户 ID, NET_DVR_Login_V40 的返回值
[in] lpCompressAudio 音频压缩参数

返回值: -1 表示失败, 其他值为音频编码句柄。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:

[返回目录](#)

音频编解码(Windows 32 位系统支持)

G722 音频编解码

5.20.12 初始化音频编码 **NET_DVR_InitG722Encoder**

函 数: void* NET_DVR_InitG722Encoder()

参 数:

返回值: -1 表示失败, 其他值为音频编码句柄。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:

[返回目录](#)

5.20.13 G722 音频编码 **NET_DVR_EncodeG722Frame**

函 数: BOOL NET_DVR_EncodeG722Frame(void *pEncodeHandle, unsigned char* pInBuffer, unsigned char* pOutBuffer)

参 数: [in]pEncodeHandle 音频编码句柄, NET_DVR_InitG722Encoder 的返回值
[in]pInBuffer 输入缓冲区, 按采样标准 (采样频率为 16000, 16 位采样, 单通道) 获取的 PCM 音频数据, 规定输入数据的大小为 1280 字节

[out]pOutBuffer 输出缓冲区, 编码后的输出数据大小为 80 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: 主要为配合语音对讲、转发功能而设定, 当需将客户端的原始音频数据发送至设备端, 可采用音频编码函数将原始数据压缩编码后再发往设备端; 客户端获取设备端发送过来的压缩码流, 可调用音频解码函数 NET_DVR_DecodeG722Frame 进行数据解码。在调用编解码函数之前都需要做相应的初始化操作, 在结束调用后还需要做释放资源的操作。

[返回目录](#)

5.20.14 释放音频编码资源 **NET_DVR_ReleaseG722Encoder**

函 数: void NET_DVR_ReleaseG722Encoder(void *pEncodeHandle)

参 数: [in]pEncodeHandle 音频编码句柄, NET_DVR_InitG722Encoder 的返回值

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

说 明:

[返回目录](#)

5.20.15 初始化音频解码 **NET_DVR_InitG722Decoder**

函 数: void* NET_DVR_InitG722Decoder(int nBitrate = 16000)

参 数: [in]nBitrate 编码采样频率, 这里我们规定采样频率为 16000

返回值: -1 表示失败, 其他值为音频解码句柄。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:

[返回目录](#)

5.20.16 G722 音频解码 **NET_DVR_DecodeG722Frame**

函数: `BOOL NET_DVR_DecodeG722Frame(void *pDecHandle, unsigned char* pInBuffer, unsigned char* pOutBuffer)`

参数: [in]pDecHandle 音频解码句柄, NET_DVR_InitG722Decoder 的返回值
 [in]pInBuffer 输入缓冲区, 编码数据大小为 80 字节
 [out]pOutBuffer 输出缓冲区, 按采样标准 (采样频率为 16000, 16 位采样, 单通道) 获取的 PCM 音频数据, 规定输出数据的大小为 1280 字节

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

说明: 主要为配合语音对讲、转发功能而设定, 当需将客户端的原始音频数据发送至设备端, 可采用音频编码函数 `NET_DVR_EncodeG722Frame` 将原始数据压缩编码后再发往设备端; 客户端获取设备端发送过来的压缩码流, 可调用音频解码函数进行数据解码。在调用编解码函数之前都需要做相应的初始化操作, 在结束调用后还需要做释放资源的操作。

[返回目录](#)

5.20.17 释放音频解码资源 **NET_DVR_ReleaseG722Decoder**

函数: `void NET_DVR_ReleaseG722Decoder(void *pDecHandle)`

参数: [in]pDecHandle 音频解码句柄, NET_DVR_InitG722Decoder 的返回值

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

说明:

[返回目录](#)

G711 音频编解码

5.20.18 G711 音频编码 **NET_DVR_EncodeG711Frame**

函数: `BOOL NET_DVR_EncodeG711Frame(unsigned int iType, unsigned char *pInBuffer, unsigned char *pOutBuffer)`

参数: [in]iType 编码类型: 0-Mu law 编码, 非 0-A law 编码
 [in]pInBuffer 输入缓冲区, 按采样标准 (采样频率为 8000, 16 位采样, 单通道) 获取的 PCM 音频数据, 规定输入数据的大小为 320 字节
 [out]pOutBuffer 输出缓冲区, 编码后输出数据大小为 160 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: 主要为配合语音对讲而设定。当需将客户端的原始音频数据发送至设备端, 可采用音频编码函数将原始数据压缩编码后再发往设备端; 客户端获取设备端发送过来的压缩码流, 可调用音频解码函数 `NET_DVR_DecodeG711Frame` 进行解码。在调用编解码函数之前无需做初始化操作。

[返回目录](#)

5.20.19 G711 音频解码 **NET_DVR_DecodeG711Frame**

- 函 数: `BOOL NET_DVR_DecodeG711Frame(unsigned int iType, unsigned char *pInBuffer, unsigned char *pOutBuffer)`
- 参 数: `[in]iType` 编码类型: 0-Mu law 编码, 非 0-A law 编码
`[in]pInBuffer` 输入缓冲区, 编码数据大小为 160 字节
`[out]pOutBuffer` 输出缓冲区, 按采样标准 (采样频率为 8000, 16 位采样, 单通道) 获取的 PCM 音频数据, 规定输出数据的大小为 320 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: 主要为配合语音对讲、转发功能而设定, 当需将客户端的原始音频数据发送至设备端, 可采用音频编码函数 `NET_DVR_EncodeG711Frame` 将原始数据压缩编码后再发往设备端; 客户端获取设备端发送过来的压缩码流, 可调用音频解码函数进行数据解码。在调用编解码函数之前无需做初始化操作。

[返回目录](#)

G726 音频编解码

5.20.20 初始化音频编码 **NET_DVR_InitG726Encoder**

- 函 数: `void* NET_DVR_InitG726Encoder(void **pEncMoudle)`
- 参 数: `[out]pEncMoudle` 编码模块句柄, 编码时作为输入参数
- 返回值: -1 表示失败, 其他值为音频编码句柄。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:

[返回目录](#)

5.20.21 G726 音频编码 **NET_DVR_EncodeG726Frame**

- 函 数: `BOOL NET_DVR_EncodeG726Frame(void *pEncMoudle, unsigned char *pInBuffer, unsigned char *pOutBuffer, BYTE byReset)`
- 参 数: `[in]pEncMoudle` 音频编码句柄, `NET_DVR_InitG726Encoder` 的输出参数
`[in]pInBuffer` 输入缓冲区, 按采样标准 (采样频率为 8000, 16 位采样, 单通道) 获取的 PCM 音频数据, 规定输入数据的大小为 640 字节
`[out]pOutBuffer` 输出缓冲区, 编码后输出数据大小为 80 字节
`[in]byReset` 是否重置, 第一帧需要重置
- 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: 主要为配合语音对讲、转发功能而设定, 当需将客户端的原始音频数据发送至设备端, 可采用音频编码函数将原始数据压缩编码后再发往设备端; 客户端获取设备端发送过来的压缩码流, 可调用音频解码函数 `NET_DVR_DecodeG726Frame` 进行数据解码。在调用编解码函数之前都需要做相应的初始化操作, 在结束调用后还需要做释放资源的操作。

[返回目录](#)

5.20.22 释放音频编码资源 **NET_DVR_ReleaseG726Encoder**

函 数: void NET_DVR_ReleaseG726Encoder(void *pEncHandle)
参 数: [in]pEncHandle 音频编码句柄, NET_DVR_InitG726Encoder 的返回值
返回值: 无。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
说 明:

[返回目录](#)

5.20.23 初始化音频解码 **NET_DVR_InitG726Decoder**

函 数: void* NET_DVR_InitG726Decoder(void **pDecMoudle)
参 数: [out]pDecMoudle 解码模块句柄, 解码时作为输入参数
返回值: -1 表示失败, 其他值为音频解码句柄。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:

[返回目录](#)

5.20.24 G726 音频解码 **NET_DVR_DecomposeG726Frame**

函 数: BOOL NET_DVR_DecomposeG722Frame(void *pDecMoudle, unsigned char *pInBuffer, unsigned char *pOutBuffer, BYTE byReset)
参 数: [in]pDecMoudle 音频解码句柄, NET_DVR_InitG726Decoder 的输出参数
[in]pInBuffer 输入缓冲区, 编码数据大小为 80 字节
[out]pOutBuffer 输出缓冲区, 按采样标准 (采样频率为 8000, 16 位采样, 单通道) 获取的 PCM 音频数据, 规定输出数据的大小为 640 字节
[in]byReset 是否重置, 第一帧需要重置
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明: 主要为配合语音对讲、转发功能而设定, 当需将客户端的原始音频数据发送至设备端, 可采用音频编码函数 NET_DVR_EncodeG726Frame 将原始数据压缩编码后再发往设备端; 客户端获取设备端发送过来的压缩码流, 可调用音频解码函数进行数据解码。在调用编解码函数之前都需要做相应的初始化操作, 在结束调用后还需要做释放资源的操作。

[返回目录](#)

5.20.25 释放音频解码资源 **NET_DVR_ReleaseG726Decoder**

函 数: void NET_DVR_ReleaseG726Decoder(void *pDecHandle)
参 数: [in]pDecHandle 音频解码句柄, NET_DVR_InitG726Decoder 的返回值
返回值: 无返回值。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
说 明:

[返回目录](#)

5.21 数据透传

透明通道

5.21.1 建立透明通道 **NET_DVR_SerialStart**

函 数: LONG NET_DVR_SerialStart(LONG IUserID, LONG ISerialPort, fSerialDataCallBack
cbSerialDataCallBack, DWORD dwUser)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]ISerialPort 串口号: 1- 232 串口, 2- 485 串口
[in]cbSerialDataCallBack 透明通道数据回调函数
[in]dwUser 用户数据

```
typedef void(CALLBACK *fSerialDataCallBack)(LONG ISerialHandle, char *pRecvDataBuffer, DWORD dwBufSize, DWORD dwUser)
```

[out]ISerialHandle NET_DVR_SerialStart 的返回值
[out]pRecvDataBuffer 存放数据的缓冲区指针
[out]dwBufSize 数据大小
[out]dwUser 用户数据

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: 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](#)

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

函 数: BOOL NET_DVR_SerialSend(LONG ISerialHandle, LONG IChannel, char *pSendBuf, DWORD dwBufSize)

参 数: [in]ISerialHandle NET_DVR_SerialStart 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]pSendBuf 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:

[返回目录](#)

5.21.3 断开透明通道 **NET_DVR_SerialStop**

函 数: BOOL NET_DVR_SerialStop (LONG ISerialHandle)

参 数: [in]ISerialHandle NET_DVR_SerialStart 的返回值

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

说 明:

[返回目录](#)

向串口发送数据

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

函 数: `BOOL NET_DVR_SendToSerialPort(LONG lUserID, DWORD dwSerialPort, DWORD dwSerialIndex, char *pSendBuf, DWORD dwBufSize)`

参 数:

<code>[in]lUserID</code>	NET_DVR_Login_V40 的返回值
<code>[in]dwSerialPort</code>	串口类型: 1-232, 2-485
<code>[in]dwSerialIndex</code>	表示第几个 232 或者 485, 从 1 开始
<code>[in]pSendBuf</code>	发送数据的缓冲区指针
<code>[in]dwBufSize</code>	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:

[返回目录](#)

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

函 数: `BOOL NET_DVR_SendTo232Port(LONG lUserID, char *pSendBuf, DWORD dwBufSize)`

参 数:

<code>[in]lUserID</code>	NET_DVR_Login_V40 的返回值
<code>[in]pSendBuf</code>	发送数据的缓冲区指针
<code>[in]dwBufSize</code>	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:

[返回目录](#)

5.22 存储管理

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

函 数: `LONG NET_DVR_FormatDisk(LONG lUserID, LONG lDiskNumber)`

参 数:

<code>[in]lUserID</code>	NET_DVR_Login_V40 的返回值
<code>[in]lDiskNumber</code>	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: 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](#)

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

函 数: `BOOL NET_DVR_GetFormatProgress(LONG lFormatHandle, LONG *pCurrentFormatDisk, LONG *pCurrentDiskPos, LONG *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:

[返回目录](#)

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

函 数: BOOL NET_DVR_CloseFormatHandle(LONG 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:

[返回目录](#)

5.23 设备手动录像

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

函 数: BOOL NET_DVR_StartDVRRecord(LONG IUserID, LONG IChannel, LONG IRecordType)
 参 数: [in]IUserID NET_DVR_Login_V40 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:

- 录像类型设置需要设备支持, 网络摄像机和球机只支持手动录像一种类型。
- 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 和 NET_DVR_GetDVRWorkState; 结构体 NET_DVR_WORKSTATE_V30 和 NET_DVR_WORKSTATE) 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](#)

5.23.2 远程手动停止设备录像 **NET_DVR_StopDVRRecord**

函 数: BOOL NET_DVR_StopDVRRecord(LONG IUserID, LONG IChannel)

参 数: [in] IUserID NET_DVR_Login_V40 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:

[返回目录](#)

5.24 服务器测试

服务器测试

5.24.1 启动长连接远程配置 **NET_DVR_StartRemoteConfig**

函 数: LONG NET_DVR_StartRemoteConfig(LONG IUserID, DWORD dwCommand, LPVOID lpInBuffer, DWORD dwInBufferLen, fRemoteConfigCallback cbStateCallback, LPVOID pUserData)

参 数: [in] IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in] dwCommand 配置命令, 详见表 5.50
[in] lpInBuffer 输入参数, 具体内容跟配置命令相关详见表 5.50
[in] dwInBufferLen 输入缓冲的大小
[in] cbStateCallback 状态回调函数, 设为 NULL
[in] pUserData 用户数据

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

[out] dwType 配置状态, 无效
[out] lpBuffer 存放数据的缓冲区指针
[out] dwBufLen 缓冲区大小
[out] pUserData 用户数据

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

说 明: 长连接状态通过 NET_DVR_GetRemoteConfigState 获取。通过 [NET_DVR_GetDeviceAbility](#)(能力集类型: DEVICE_NETAPP_ABILITY) 可以获得 NTP、NAS、Email、FTP、IP 测试能力。

表 5.50 服务器测试命令

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer	cbStateCallback
NET_DVR_NTP_SERVER_TEST	3387	NTP 服务器测试	NET_DVR_SERVER_TEST_PARA	NULL
NET_DVR_NAS_SERVER_TEST	3388	NAS 服务器测试		
NET_DVR_EMAIL_SERVER_TEST	3389	Email 服务器测试		
NET_DVR_FTP_SERVER_TEST	3390	FTP 服务器测试		
NET_DVR_IP_TEST	3391	IP 测试		

[返回目录](#)

5.24.2 获取长连接配置的状态 **NET_DVR_GetRemoteConfigState**

函 数: BOOL NET_DVR_GetRemoteConfigState(LONG IHandle, void *pState)
参 数: [in] IHandle 句柄, NET_DVR_StartRemoteConfig 的返回值
[out] pState 返回的状态值, 详见表 5.51
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

表 5.51 长连接状态

pState 状态值	含义
29	设备操作失败
55	IP 地址不匹配
165	设备连接测试服务器失败
166	NAS 服务器挂载目录失败, 目录无效
167	NAS 服务器挂载目录失败, 没有权限或用户名密码错误
168	服务器使用域名, 但是没有配置 DNS, 可能造成域名无效
169	没有配置网关, 可能造成发送邮件失败
170	测试服务器的登陆用户名或密码错误
171	设备和 smtp 服务器交互异常
172	该用户在服务器上创建目录失败
173	该用户没有写入权限
174	IP 冲突

[返回目录](#)

5.24.3 关闭长连接配置接口所创建的句柄, 释放资源

NET_DVR_StopRemoteConfig

函 数: BOOL NET_DVR_StopRemoteConfig(LONG IHandle)
参 数: [in] IHandle 句柄, NET_DVR_StartRemoteConfig 的返回值
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

Email 测试

5.24.4 测试按已配置的 EMAIL 参数能否收发成功 **NET_DVR_StartEmailTest**

函 数: LONG NET_DVR_StartEmailTest(LONG IUserID)
参 数: [in] IUserID NET_DVR_Login_V40 的返回值

返回值: -1 表示失败, 其他值作为 NET_DVR_GetEmailTestProcess 、NET_DVR_StopEmailTest 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: 在调用此接口测试前, 需要配置相关的 EMAIL 参数, 详见 NET_DVR_GetDVRConfig、NET_DVR_SetDVRConfig 的网络应用参数 (EMAIL) 配置。

[返回目录](#)

5.24.5 获取邮件测试的进度 **NET_DVR_GetEmailTestProgress**

函数: BOOL NET_DVR_GetEmailTestProgress(LONG IEmailTestHandle, DWORD* pState)

参数: [in]IEmailTestHandle NET_DVR_StartEmailTest 的返回值
[out]pState 邮件测试的进度, 进度值的取值范围 (0,100), 其他值定义如表 5.52 所示。

表 5.52 邮件测试进度

pState 宏定义	宏定义值	含义
PROCESSING	0	正在处理
PROCESS_SUCCESS	100	过程完成
PROCESS_EXCEPTION	400	过程异常
PROCESS_FAILED	500	过程失败

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。

[返回目录](#)

5.24.6 停止邮件测试 **NET_DVR_StopEmailTest**

函数: BOOL NET_DVR_StopEmailTest(LONG IEmailTestHandle)

参数: [in]IEmailTestHandle NET_DVR_StartEmailTest 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。

[返回目录](#)

5.25 设备维护管理

获取设备工作状态

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

函数: BOOL NET_DVR_GetDVRWorkState_V30(LONG IUserID, LPNET_DVR_WORKSTATE_V30 lpWorkState)

参数: [in]IUserID NET_DVR_Login_V40 的返回值
[out]lpWorkState 获取的设备工作状态结构体参数

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

过错误码判断出错原因。

说 明：

[返回目录](#)

5.25.2 设备在线状态检测 **NET_DVR_RemoteControl**

函 数： BOOL NET_DVR_RemoteControl(LONG lUserID, DWORD dwCommand, LPVOID lpInBuffer, DWORD dwInBufferSize)

参 数： [in]lUserID 用户 ID 号，NET_DVR_Login_V40 的返回值
[in]dwCommand 控制命令，详见表 5.53
[in]lpInBuffer 输入参数，具体内容跟控制命令相关，详见表 5.53
[in]dwInBufferSize 输入参数长度

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

说 明： 该控制命令用于手动检测设备是否在线，接口返回 TRUE 表示在线，FALSE 表示与设备通信失败或者返回错误状态。设备在线状态自动巡检功能通过 [NET_DVR_SetSDKLocalCfg](#)（配置类型：NET_SDK_LOCAL_CFG_TYPE_CHECK_DEV）进行配置。

表 5.53 远程控制命令

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer 对应结构体
NET_DVR_CHECK_USER_STATUS	20005	检测设备是否在线	NULL

[返回目录](#)

5.25.3 启动设备状态巡检 **NET_DVR_StartGetDevState**

函 数： BOOL NET_DVR_StartGetDevState(LPNET_DVR_CHECK_DEV_STATE pParams)

参 数： [in] pParams 设备工作状态巡检参数，包括巡检时间、结果回调函数等，详见结构体：NET_DVR_CHECK_DEV_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： 启动后，SDK 定时巡检设备，获取到的设备状态信息在结构体的回调函数中返回。相当于实现定时调用 [NET_DVR_GetDeviceConfig](#)(命令：NET_DVR_GET_WORK_STATUS)。

[返回目录](#)

5.25.4 停止设备状态巡检 **NET_DVR_StopGetDevState**

函 数： BOOL NET_DVR_StopGetDevState()

参 数： 无

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

说 明： 停止巡检设备工作状态，释放资源。

[返回目录](#)

远程升级

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

函 数: BOOL NET_DVR_SetNetworkEnvironment(DWORD dwEnvironmentLevel)

参 数: [in]dwEnvironmentLevel 网络环境级别

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

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

说 明: 接口中的网络环境级别参数分为两类:

LOCAL_AREA_NETWORK 表示局域网环境(网络环境好, 通讯流畅);

WIDE_AREA_NETWORK 表示广域网环境(网络环境差, 易阻塞)。

在调用远程升级接口之前, 可以通过此接口适应不同的升级环境。

[返回目录](#)

5.25.6 远程升级 **NET_DVR_Upgrade**

函 数: LONG NET_DVR_Upgrade(LONG IUserID, char *sFileName)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值

[in]sFileName 升级的文件路径(包括文件名)。路径长度和操作系统有关, sdk 不做限制, windows 默认路径长度小于等于 256 字节(包括文件名在内)。

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

说 明:

[返回目录](#)

5.25.7 获取远程升级的进度 **NET_DVR_GetUpgradeProgress**

函 数: int NET_DVR_GetUpgradeProgress(LONG IUpgradeHandle)

参 数: [in]IUpgradeHandle NET_DVR_Upgrade 的返回值

返回值: -1 表示失败, 0~100 表示升级进度。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

5.25.8 获取远程升级的状态 **NET_DVR_GetUpgradeState**

函 数: int NET_DVR_GetUpgradeState(LONG IUpgradeHandle)

参 数: [in]IUpgradeHandle NET_DVR_Upgrade 的返回值

返回值： -1 表示失败，其他值定义如下：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；7- 升级包类型不匹配；8- 升级包版本不匹配。
接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说明：

[返回目录](#)

5.25.9 获取远程升级的阶段信息 **NET_DVR_GetUpgradeStep**

函数： LONG NET_DVR_GetUpgradeStep(LONG IUpgradeHandle, LONG *pSubProgress)

参数： [in]IUpgradeHandle NET_DVR_Upgrade 的返回值
 [in]pSubProgress 升级阶段子进度

返回值： -1 表示失败，其他值定义如表 5.54 所示。

表 5.54 升级阶段信息

宏定义	宏定义值	含义
STEP_RECV_DATA	1	接收升级包数据
STEP_UPGRADE	2	升级系统
STEP_BACKUP	3	备份系统
STEP_SEARCH	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：

[返回目录](#)

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

函数： BOOL NET_DVR_CloseUpgradeHandle(LONG IUpgradeHandle)

参数： [in]IUpgradeHandle NET_DVR_Upgrade 的返回值

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

说明：

[返回目录](#)

日志查找

5.25.11 查找设备的日志信息 **NET_DVR_FindDVRLog_V30**

函数： LONG NET_DVR_FindDVRLog_V30(LONG IUserID, LONG ISelectMode, DWORD dwMajorType, DWORD dwMinorType, LPNET_DVR_TIME lpStartTime, LPNET_DVR_TIME lpStopTime, BOOL bOnlySmart = FALSE)

参数： [in]IUserID NET_DVR_Login_V40 的返回值
 [in]ISelectMode 查询方式：0- 全部，1- 按类型，2- 按时间，3- 按时间和类型
 [in]dwMajorType 日志主类型（S.M.A.R.T 搜索时无效），0 表示全部类型，其他类型
 定义见表 5.55

[in]dwMinorType	日志次类型（S.M.A.R.T 搜索时无效），0 表示全部类型，根据不同的主类型的次类型定义如见表 5.56~表 5.59
[in]lpStartTime	文件的开始时间
[in]lpStopTime	文件结束时间
[in]bOnlySmart	是否只搜索带 S.M.A.R.T 信息的日志

返回值：-1 表示失败，其他值作为 NET_DVR_FindNextLog_V30 等函数的参数。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说 明：该接口如果用于搜索普通日志信息，一般日志设备支持 2000 条，而搜索带 S.M.A.R.T 信息的日志最大只支持 500 条。通常不需要搜索详细的 S.M.A.R.T 信息时，置 bOnlySmart 为 FALSE 即可完成所有日志信息的搜索。

S.M.A.R.T 信息：硬盘运行日志记录。

表 5.55 日志主类型

宏定义	宏定义值	含义
MAJOR_ALARM	0x1	报警
MAJOR_EXCEPTION	0x2	异常
MAJOR_OPERATION	0x3	操作
MAJOR_INFORMATION	0x4	日志附加信息

表 5.56 报警日志次类型

主类型的宏定义	宏定义值	含义
MAJOR_ALARM	0x1	报警
次类型的宏定义	宏定义值	含义
MINOR_ALARM_IN	0x1	报警输入
MINOR_ALARM_OUT	0x2	报警输出
MINOR_MOTDET_START	0x3	移动侦测报警开始
MINOR_MOTDET_STOP	0x4	移动侦测报警结束
MINOR_HIDE_ALARM_START	0x5	遮挡报警开始
MINOR_HIDE_ALARM_STOP	0x6	遮挡报警结束
MINOR_VCA_ALARM_START	0x7	智能报警开始
MINOR_VCA_ALARM_STOP	0x8	智能报警结束
MINOR_ITS_ALARM_START	0x09	交通事件报警开始
MINOR_ITS_ALARM_STOP	0x0a	交通事件报警结束
MINOR_NETALARM_START	0x0b	网络报警开始
MINOR_NETALARM_STOP	0x0c	网络报警结束
MINOR_NETALARM_RESUME	0x0d	网络报警恢复
MINOR_WIRELESS_ALARM_START	0x0e	无线报警开始
MINOR_WIRELESS_ALARM_STOP	0x0f	无线报警结束
MINOR_PIR_ALARM_START	0x10	人体感应报警开始

MINOR_PIR_ALARM_STOP	0x11	人体感应报警结束
MINOR_CALLHELP_ALARM_START	0x12	呼救报警开始
MINOR_CALLHELP_ALARM_STOP	0x13	呼救报警结束
MINOR_DETECTFACE_ALARM_START	0x16	人脸侦测报警开始
MINOR_DETECTFACE_ALARM_STOP	0x17	人脸侦测报警结束
MINOR_VCA_SECNECHANGE_DETECTION	0x1a	场景侦测报警
MINOR_SMART_REGION_EXITING_BEGIN	0x1b	离开区域侦测开始
MINOR_SMART_REGION_EXITING_END	0x1c	离开区域侦测结束
MINOR_SMART_LOITERING_BEGIN	0x1d	徘徊侦测开始
MINOR_SMART_LOITERING_END	0x1e	徘徊侦测结束
MINOR_VCA_ALARM_LINE_DETECTION_BEGIN	0x20	越界侦测开始
MINOR_VCA_ALARM_LINE_DETECTION_END	0x21	越界侦测结束
MINOR_VCA_ALARM_INTRUDE_BEGIN	0x22	区域侦测开始
MINOR_VCA_ALARM_INTRUDE_END	0x23	区域侦测结束
MINOR_VCA_ALARM_AUDIOINPUT	0x24	音频异常输入
MINOR_VCA_ALARM_AUDIOABNORMAL	0x25	声强突变
MINOR_VCA_DEFOCUS_DETECTION_BEGIN	0x26	虚焦侦测开始
MINOR_VCA_DEFOCUS_DETECTION_END	0x27	虚焦侦测结束
MINOR_VCA_FACE_ALARM_BEGIN	0x29	人脸侦测开始
MINOR_SMART_REGION_ENTRANCE_BEGIN	0x2a	进入区域侦测开始
MINOR_SMART_REGION_ENTRANCE_END	0x2b	进入区域侦测结束
MINOR_SMART_PEOPLE_GATHERING_BEGIN	0x2c	人员聚集侦测开始
MINOR_SMART_PEOPLE_GATHERING_END	0x2d	人员聚集侦测结束
MINOR_SMART_FAST_MOVING_BEGIN	0x2e	快速运动侦测开始
MINOR_SMART_FAST_MOVING_END	0x2f	快速运动侦测结束
MINOR_VCA_FACE_ALARM_END	0x30	人脸侦测结束
MINOR_VCA_SCENE_CHANGE_ALARM_BEGIN	0x31	场景变更侦测开始
MINOR_VCA_SCENE_CHANGE_ALARM_END	0x32	场景变更侦测结束
MINOR_VCA_ALARM_AUDIOINPUT_BEGIN	0x33	音频异常输入开始
MINOR_VCA_ALARM_AUDIOINPUT_END	0x34	音频异常输入结束
MINOR_VCA_ALARM_AUDIOABNORMAL_BEGIN	0x35	声强突变侦测开始
MINOR_VCA_ALARM_AUDIOABNORMAL_END	0x36	声强突变侦测结束
MINOR_VCA_ALARM_AUDIOSTEEPDROP	0x39	声强陡降
MINOR_SMART_PARKING_BEGIN	0x3c	停车侦测开始
MINOR_SMART_PARKING_END	0x3d	停车侦测结束

MINOR_SMART_UNATTENDED_BAGGAGE_BEGIN	0x3e	物品遗留侦测开始
MINOR_SMART_UNATTENDED_BAGGAGE_END	0x3f	物品遗留侦测结束
MINOR_SMART_OBJECT_REMOVAL_BEGIN	0x40	物品拿取侦测开始
MINOR_SMART_OBJECT_REMOVAL_END	0x41	物品拿取侦测结束
MINOR_SMART_VEHICLE_ALARM_START	0x46	车牌检测开始
MINOR_SMART_VEHICLE_ALARM_STOP	0x47	车牌检测结束
MINOR_THERMAL_FIREDETECTION	0x48	热成像火点检测侦测开始
MINOR_THERMAL_FIREDETECTION_END	0x49	热成像火点检测侦测结束

表 5.57 异常日志次类型

主类型的宏定义	宏定义值	含义
MAJOR_EXCEPTION	0x2	异常
次类型的宏定义	宏定义值	含义
MINOR_RAID_ERROR	0x20	阵列异常
MINOR_VI_LOST	0x21	视频信号丢失
MINOR_ILLEGAL_ACCESS	0x22	非法访问
MINOR_HD_FULL	0x23	硬盘满
MINOR_HD_ERROR	0x24	硬盘错误
MINOR_DCD_LOST	0x25	MODEM 掉线(保留)
MINOR_IP_CONFLICT	0x26	IP 地址冲突
MINOR_NET_BROKEN	0x27	网络断开
MINOR_REC_ERROR	0x28	录像出错
MINOR_IPC_NO_LINK	0x29	IPC 连接异常
MINOR_VI_EXCEPTION	0x2a	视频输入异常(只针对模拟通道)
MINOR_IPC_IP_CONFLICT	0x2b	IPC 的 IP 地址冲突
MINOR_SENCE_EXCEPTION	0x2c	场景异常
MINOR_PIC_REC_ERROR	0x2d	抓图出错,获取图片文件失败
MINOR_VI_MISMATCH	0x2e	视频制式不匹配

表 5.58 操作日志次类型

主类型的宏定义	宏定义值	含义
MAJOR_OPERATION	0x3	操作
次类型的宏定义	宏定义值	含义
MINOR_START_DVR	0x41	开机
MINOR_STOP_DVR	0x42	关机
MINOR_STOP_ABNORMAL	0x43	异常关机

MINOR_REBOOT_DVR	0x44	本地重启设备
MINOR_LOCAL_LOGIN	0x50	本地登陆
MINOR_LOCAL_LOGOUT	0x51	本地注销登陆
MINOR_LOCAL_CFG_PARM	0x52	本地配置参数
MINOR_LOCAL_PLAYBYFILE	0x53	本地按文件回放或下载
MINOR_LOCAL_PLAYBYTIME	0x54	本地按时间回放或下载
MINOR_LOCAL_START_REC	0x55	本地开始录像
MINOR_LOCAL_STOP_REC	0x56	本地停止录像
MINOR_LOCAL_PTZCTRL	0x57	本地云台控制
MINOR_LOCAL_PREVIEW	0x58	本地预览(保留不使用)
MINOR_LOCAL_MODIFY_TIME	0x59	本地修改时间(保留不使用)
MINOR_LOCAL_UPGRADE	0x5a	本地升级
MINOR_LOCAL_RECFILE_OUTPUT	0x5b	本地备份录像文件
MINOR_LOCAL_FORMAT_HDD	0x5c	本地初始化硬盘
MINOR_LOCAL_CFGFILE_OUTPUT	0x5d	导出本地配置文件
MINOR_LOCAL_CFGFILE_INPUT	0x5e	导入本地配置文件
MINOR_LOCAL_COPYFILE	0x5f	本地备份文件
MINOR_LOCAL_LOCKFILE	0x60	本地锁定录像文件
MINOR_LOCAL_UNLOCKFILE	0x61	本地解锁录像文件
MINOR_LOCAL_DVR_ALARM	0x62	本地手动清除和触发报警
MINOR_IPC_ADD	0x63	本地添加 IPC
MINOR_IPC_DEL	0x64	本地删除 IPC
MINOR_IPC_SET	0x65	本地设置 IPC
MINOR_LOCAL_START_BACKUP	0x66	本地开始备份
MINOR_LOCAL_STOP_BACKUP	0x67	本地停止备份
MINOR_LOCAL_COPYFILE_START_TIME	0x68	本地备份开始时间
MINOR_LOCAL_COPYFILE_END_TIME	0x69	本地备份结束时间
MINOR_LOCAL_ADD_NAS	0x6a	本地添加网络硬盘
MINOR_LOCAL_DEL_NAS	0x6b	本地删除 NAS 盘
MINOR_LOCAL_SET_NAS	0x6c	本地设置 NAS 盘
MINOR_REMOTE_LOGIN	0x70	远程登录
MINOR_REMOTE_LOGOUT	0x71	远程注销登陆
MINOR_REMOTE_START_REC	0x72	远程开始录像
MINOR_REMOTE_STOP_REC	0x73	远程停止录像
MINOR_START_TRANS_CHAN	0x74	开始透明传输

MINOR_STOP_TRANS_CHAN	0x75	停止透明传输
MINOR_REMOTE_GET_PARM	0x76	远程获取参数
MINOR_REMOTE_CFG_PARM	0x77	远程配置参数
MINOR_REMOTE_GET_STATUS	0x78	远程获取状态
MINOR_REMOTE_ARM	0x79	远程布防
MINOR_REMOTE_DISARM	0x7a	远程撤防
MINOR_REMOTE_REBOOT	0x7b	远程重启
MINOR_START_VT	0x7c	开始语音对讲
MINOR_STOP_VT	0x7d	停止语音对讲
MINOR_REMOTE_UPGRADE	0x7e	远程升级
MINOR_REMOTE_PLAYBYFILE	0x7f	远程按文件回放
MINOR_REMOTE_PLAYBYTIME	0x80	远程按时间回放
MINOR_REMOTE_PTZCTRL	0x81	远程云台控制
MINOR_REMOTE_FORMAT_HDD	0x82	远程格式化硬盘
MINOR_REMOTE_STOP	0x83	远程关机
MINOR_REMOTE_LOCKFILE	0x84	远程锁定文件
MINOR_REMOTE_UNLOCKFILE	0x85	远程解锁文件
MINOR_REMOTE_CFGFILE_OUTPUT	0x86	远程导出配置文件
MINOR_REMOTE_CFGFILE_INPUT	0x87	远程导入配置文件
MINOR_REMOTE_RECFILE_OUTPUT	0x88	远程导出录像文件
MINOR_REMOTE_DVR_ALARM	0x89	远程手动清除和触发报警
MINOR_REMOTE_IPC_ADD	0x8a	远程添加 IPC
MINOR_REMOTE_IPC_DEL	0x8b	远程删除 IPC
MINOR_REMOTE_IPC_SET	0x8c	远程设置 IPC
MINOR_REBOOT_VCA_LIB	0x8d	重启智能库
MINOR_REMOTE_ADD_NAS	0x8e	远程添加 NAS 盘
MINOR_REMOTE_DEL_NAS	0x8f	远程删除 NAS 盘
MINOR_REMOTE_SET_NAS	0x90	远程设置 NAS 盘
MINOR_LOCAL_START_REC_CDRW	0x91	本地开始刻录
MINOR_LOCAL_STOP_REC_CDRW	0x92	本地停止刻录
MINOR_REMOTE_START_REC_CDRW	0x93	远程开始刻录
MINOR_REMOTE_STOP_REC_CDRW	0x94	远程停止刻录
MINOR_LOCAL_PIC_OUTPUT	0x95	本地备份图片文件
MINOR_REMOTE_SET_NAS	0x96	远程备份图片文件
MINOR_LOCAL_INQUEST_RESUME	0x97	本地恢复审讯事件

MINOR_REMOTE_INQUEST_RESUME	0x98	远程恢复审讯事件
MINOR_REMOTE_BYPASS	0xd0	远程旁路
MINOR_REMOTE_UNBYPASS	0xd1	远程旁路恢复
MINOR_REMOTE_SET_ALARMIN_CFG	0xd2	远程设置报警输入参数
MINOR_REMOTE_GET_ALARMIN_CFG	0xd3	远程获取报警输入参数
MINOR_REMOTE_SET_ALARMOUT_CFG	0xd4	远程设置报警输出参数
MINOR_REMOTE_GET_ALARMOUT_CFG	0xd5	远程获取报警输出参数
MINOR_REMOTE_ALARMOUT_OPEN_MAN	0xd6	远程手动开启报警输出
MINOR_REMOTE_ALARMOUT_CLOSE_MAN	0xd7	远程手动关闭报警输出
MINOR_REMOTE_ALARM_ENABLE_CFG	0xd8	远程设置报警主机的 RS485 串口使能状态
MINOR_DBDATA_OUTPUT	0xd9	导出数据库记录
MINOR_DBDATA_INPUT	0xda	导入数据库记录
MINOR_MU_SWITCH	0xdb	级联切换
MINOR_MU_PTZ	0xdc	级联 PTZ 控制
MINOR_LOCAL_CONF_REB_RAID	0x101	本地配置自动重建
MINOR_LOCAL_CONF_SPARE	0x102	本地配置热备
MINOR_LOCAL_ADD_RAID	0x103	本地创建阵列
MINOR_LOCAL_DEL_RAID	0x104	本地删除阵列
MINOR_LOCAL_MIG_RAID	0x105	本地迁移阵列
MINOR_LOCAL_REB_RAID	0x106	本地手动重建阵列
MINOR_LOCAL_QUICK_CONF_RAID	0x107	本地一键配置
MINOR_LOCAL_ADD_VD	0x108	本地创建虚拟磁盘
MINOR_LOCAL_DEL_VD	0x109	本地删除虚拟磁盘
MINOR_LOCAL_RP_VD	0x10a	本地修复虚拟磁盘
MINOR_LOCAL_FORMAT_EXPANDVD	0x10b	本地扩展虚拟磁盘扩容
MINOR_LOCAL_RAID_UPGRADE	0x10c	本地 raid 卡升级
MINOR_LOCAL_STOP_RAID	0x10d	本地暂停 RAID 操作(即安全拔盘)
MINOR_REMOTE_CONF_REB_RAID	0x111	远程配置自动重建
MINOR_REMOTE_CONF_SPARE	0x112	远程配置热备
MINOR_REMOTE_ADD_RAID	0x113	远程创建阵列
MINOR_REMOTE_DEL_RAID	0x114	远程删除阵列
MINOR_REMOTE_MIG_RAID	0x115	远程迁移阵列
MINOR_REMOTE_REB_RAID	0x116	远程手动重建阵列
MINOR_REMOTE_QUICK_CONF_RAID	0x117	远程一键配置
MINOR_REMOTE_ADD_VD	0x118	远程创建虚拟磁盘

MINOR_REMOTE_DEL_VD	0x119	远程删除虚拟磁盘
MINOR_REMOTE_RP_VD	0x11a	远程修复虚拟磁盘
MINOR_REMOTE_FORMAT_EXPANDVD	0x11b	远程虚拟磁盘扩容
MINOR_REMOTE_RAID_UPGRADE	0x11c	远程 raid 卡升级
MINOR_REMOTE_STOP_RAID	0x11d	远程暂停 RAID 操作(即安全拔盘)
MINOR_LOCAL_START_PIC_REC	0x121	本地开始抓图
MINOR_LOCAL_STOP_PIC_REC	0x122	本地停止抓图
MINOR_LOCAL_SET_SNMP	0x125	本地配置 SNMP
MINOR_LOCAL_TAG_OPT	0x126	本地标签操作
MINOR_REMOTE_START_PIC_REC	0x131	远程开始抓图
MINOR_REMOTE_STOP_PIC_REC	0x132	远程停止抓图
MINOR_REMOTE_SET_SNMP	0x135	远程配置 SNMP
MINOR_REMOTE_TAG_OPT	0x136	远程标签操作
MINOR_REMOTE_TAG_OPT	0x136	远程标签操作
MINOR_LOCAL_VOUT_SWITCH	0x140	本地输出口切换操作

表 5.59 附加日志次类型

主类型的宏定义	宏定义值	含义
MAJOR_INFORMATION	0x4	附加信息
次类型的宏定义	宏定义值	含义
MINOR_HDD_INFO	0xa1	硬盘信息
MINOR_SMART_INFO	0xa2	S.M.A.R.T 信息
MINOR_REC_START	0xa3	开始录像
MINOR_REC_STOP	0xa4	停止录像
MINOR_REC_OVERDUE	0xa5	过期录像删除
MINOR_LINK_START	0xa6	连接前端设备
MINOR_LINK_STOP	0xa7	断开前端设备
MINOR_NET_DISK_INFO	0xa8	网络硬盘信息
MINOR_RAID_INFO	0xa9	raid 相关信息
MINOR_PIC_REC_START	0xb3	开始抓图
MINOR_PIC_REC_STOP	0xb4	停止抓图
MINOR_PIC_REC_OVERDUE	0xb5	过期图片文件删除

[返回目录](#)

5.25.12 逐条获取查找到的日志信息 NET_DVR_FindNextLog_V30

函 数： LONG NET_DVR_FindNextLog_V30(LONG lLogHandle, LPNET_DVR_LOG_V30 lpLogData)

参 数: [in]ILogHandle 日志查找句柄, NET_DVR_FindDVRLog_V30()的返回值
 [out]IpLogData 保存日志信息的指针

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

说 明: 在调用该接口获取查找日志之前, 必须先调用 NET_DVR_FindDVRLog_V30 得到当前的查找句柄。
[返回目录](#)

5.25.13 释放查找日志的资源 NET_DVR_FindLogClose_V30

函 数: BOOL NET_DVR_FindLogClose_V30(LONG ILogHandle)

参 数: [in]ILogHandle 日志查找句柄, NET_DVR_FindDVRLog_V30 的返回值

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

说 明:
[返回目录](#)

恢复设备默认参数

5.25.14 恢复设备默认参数 NET_DVR_RestoreConfig

函 数: BOOL NET_DVR_RestoreConfig(LONG IUserID)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值

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

说 明:
[返回目录](#)

5.25.15 完全恢复出厂默认参数 NET_DVR_RemoteControl

函 数: BOOL NET_DVR_RemoteControl(LONG IUserID, DWORD dwCommand, LPVOID lpInBuffer, DWORD dwInBufferSize)

参 数: [in] IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
 [in] dwCommand 控制命令, 详见表 5.60
 [in] lpInBuffer 输入参数, 跟控制命令相关, 详见列表
 [in] dwInBufferSize 输入参数长度

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

说 明: 不同的控制功能对应不同的命令号, 同时 lpInBuffer 对应不同的结构体, 如表 5.60 所示。

- 远程控制设备完全恢复出厂默认值, 包括设备 IP 地址会恢复成 192.0.0.64。只有 admin 管理员账号才支持该功能。
- 完全恢复之后设备会恢复为未激活状态, 调用 [NET_DVR_ActivateDevice](#) 可以激活设备。

表 5.60 远程控制命令

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer 对应结构体
NET_DVR_COMPLETE_RESTORE_CTRL	3420	设置完全恢复出厂值	NET_DVR_COMPLETE_RESTORE_INFO

[返回目录](#)

导入/导出配置文件

5.25.16 导出配置文件 **NET_DVR_GetConfigFile_V30**

函 数: BOOL NET_DVR_GetConfigFile_V30(LONG lUserID, char *sOutBuffer, DWORD dwOutSize, DWORD *pReturnSize)

参 数: [in]lUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[out] sOutBuffer 存放配置参数的缓冲区
[in]dwOutSize 缓冲区大小
[out]pReturnSize 实际获得的缓冲区大小

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

说 明: 当 sOutBuffer = NULL、dwOutSize = 0 且 pReturnSize != NULL 时用于获取参数配置文件的所需的缓冲区长度; 当 sOutBuffer != NULL 且 dwOutSize != 0 时用于获取参数配置文件的所需的缓冲区内容。

[返回目录](#)

5.25.17 导出配置文件 **NET_DVR_GetConfigFile**

函 数: BOOL NET_DVR_GetConfigFile(LONG lUserID, char *sFileName)

参 数: [in]lUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]sFileName 存放保存配置文件的文件路径 (二进制文件)

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

说 明:

[返回目录](#)

5.25.18 导入配置文件 **NET_DVR_SetConfigFile_EX**

函 数: BOOL NET_DVR_SetConfigFile_EX(LONG lUserID, char *sInBuffer, DWORD dwInSize)

参 数: [in]lUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]sInBuffer 存放配置参数的缓冲区
[in]dwInSize 缓冲区大小

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

说 明:

[返回目录](#)

5.25.19 导入配置文件 **NET_DVR_SetConfigFile**

函 数: BOOL NET_DVR_SetConfigFile(LONG IUserID, char *sFileName)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]sFileName 存放保存配置文件的文件路径 (二进制文件)

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

说 明:

[返回目录](#)

关机和重启

5.25.20 重启设备 **NET_DVR_RebootDVR**

函 数: BOOL NET_DVR_RebootDVR(LONG IUserID)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值

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

说 明:

[返回目录](#)

5.25.21 关闭设备 **NET_DVR_ShutDownDVR**

函 数: BOOL NET_DVR_ShutDownDVR(LONG IUserID)

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值

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

说 明:

[返回目录](#)

6 错误代码及说明

6.1 网络通讯库错误码

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

NET_DVR_DVRVOICEOPENED	31	设备语音对讲被占用。
NET_DVR_TIMEINPUTERROR	32	时间输入不正确。
NET_DVR_NOSPECFILE	33	回放时设备没有指定的文件。
NET_DVR_CREATEFILE_ERROR	34	创建文件出错。本地录像、保存图片、获取配置文件和远程下载录像时创建文件失败。
NET_DVR_FILEOPENFAIL	35	打开文件出错。设置配置文件、设备升级、上传审讯文件时打开文件失败。
NET_DVR_OPERNOTFINISH	36	上次的操作还没有完成
NET_DVR_GETPLAYTIMEFAIL	37	获取当前播放的时间出错。
NET_DVR_PLAYFAIL	38	播放出错。
NET_DVR_FILEFORMAT_ERROR	39	文件格式不正确。
NET_DVR_DIR_ERROR	40	路径错误
NET_DVR_ALLOC_RESOURCE_ERROR	41	SDK 资源分配错误。
NET_DVR_AUDIO_MODE_ERROR	42	声卡模式错误。当前打开声音播放模式与实际设置的模式不符出错。
NET_DVR_NOENOUGH_BUF	43	缓冲区太小。接收设备数据的缓冲区或存放图片缓冲区不足。
NET_DVR_CREATESOCKET_ERROR	44	创建 SOCKET 出错。
NET_DVR_SETSOCKET_ERROR	45	设置 SOCKET 出错。
NET_DVR_MAX_NUM	46	个数达到最大。分配的注册连接数、预览连接数超过 SDK 支持的最大数。
NET_DVR_USERNOTEXIST	47	用户不存在。注册的用户 ID 已注销或不可用。
NET_DVR_WRITEFLASHERROR	48	写 FLASH 出错。设备升级时写 FLASH 失败。
NET_DVR_UPGRADEFAIL	49	设备升级失败。网络或升级文件语言不匹配等原因升级失败。
NET_DVR_CARDHAVEINIT	50	解码卡已经初始化过。
NET_DVR_PLAYERFAILED	51	调用播放库中某个函数失败。
NET_DVR_MAX_USERNUM	52	登录设备的用户数达到最大。
NET_DVR_GETLOCALIPANDMACFAIL	53	获得本地 PC 的 IP 地址或物理地址失败。
NET_DVR_NOENCODEING	54	设备该通道没有启动编码。
NET_DVR_IPMISMATCH	55	IP 地址不匹配。
NET_DVR_MACMISMATCH	56	MAC 地址不匹配。
NET_DVR_UPGRADELANGMISMATCH	57	升级文件语言不匹配。
NET_DVR_MAX_PLAYERPORT	58	播放器路数达到最大。
NET_DVR_NOSPACEBACKUP	59	备份设备中没有足够空间进行备份。
NET_DVR_NODEVICEBACKUP	60	没有找到指定的备份设备。
NET_DVR_PICTURE_BITS_ERROR	61	图像像素位数不符，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_INTERCOM_SDK_ERROR	100	加载当前目录下的语音对讲库失败。
NET_DVR_NO_CURRENT_UPDATEFILE	101	没有正确的升级包。
NET_DVR_ALIAS_DUPLICATE	150	别名重复（EasyDDNS 的配置）
NET_ERR_USERNAME_LOCKED	153	用户名被锁定。
NET_DVR_TEST_SERVER_FAIL_CONNECT	165	连接测试服务器失败。
NET_DVR_NAS_SERVER_INVALID_DIR	166	NAS 服务器挂载目录失败，目录无效或者用户名密码错误。
NET_DVR_NAS_SERVER_NOENOUGH_PRI	167	NAS 服务器挂载目录失败，没有权限。
NET_DVR_EMAIL_SERVER_NOT_CONFIG_DNS	168	服务器使用域名，但是没有配置 DNS，可能造成域名无效。

NET_DVR_EMAIL_SERVER_NOT_CONFIG_GATEWAY	169	没有配置网关，可能造成发送邮件失败。
NET_DVR_TEST_SERVER_PASSWORD_ERROR	170	用户名密码不正确，测试服务器的用户名或密码错误。
NET_DVR_EMAIL_SERVER_CONNECT_EXCEPTION_WITH_SMTP	171	设备和 smtp 服务器交互异常。
NET_DVR_FTP_SERVER_FAIL_CREATE_DIR	172	FTP 服务器创建目录失败。
NET_DVR_FTP_SERVER_NO_WRITE_PIR	173	FTP 服务器没有写入权限。
NET_DVR_IP_CONFLICT	174	IP 冲突。
NET_ERR_ANR_ARMING_EXIST	178	断网续传布防连接已经存在（私有 SDK 协议布防连接已经建立的情况下，重复布防且选择断网续传功能时返回该错误）。
NET_ERR_UPLOADLINK_EXIST	179	断网续传上传连接已经存在（EHOME 协议和私有 SDK 协议不能同时支持断网续传，其中一种协议已经建议连接，另外一个连接建立时返回该错误）。
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
NET_ERR_TRANS_CHAN_START	1101	透明通道已打开，当前操作无法完成。
NET_ERR_DEV_UPGRADING	1102	设备正在升级
NET_ERR_MISMATCH_UPGRADE_PACK_TYPE	1103	升级包类型不匹配
NET_ERR_DEV_FORMATTING	1104	设备正在格式化
NET_ERR_MISMATCH_UPGRADE_PACK_VERSION	1105	升级包版本不匹配
能力集错误码		
XML_ABILITY_NOTSUPPORT	1000	不支持能力节点获取。
XML_ANALYZE_NOENOUGH_BUF	1001	输出内存不足。
XML_ANALYZE_FIND_LOCALXML_ERROR	1002	无法找到对应的本地 xml。
XML_ANALYZE_LOAD_LOCALXML_ERROR	1003	加载本地 xml 出错。
XML_NANLYZE_DVR_DATA_FORMAT_ERROR	1004	设备能力数据格式错误。
XML_ANALYZE_TYPE_ERROR	1005	能力集类型错误。
XML_ANALYZE_XML_NODE_ERROR	1006	XML 能力节点格式错误。
XML_INPUT_PARAM_ERROR	1007	输入的能力 XML 节点值错误。
XML_VERSION_MISMATCH	1008	XML 版本不匹配。

6.2 RTSP 通讯库错误码

错误名称	错误值	说明
NET_DVR_RTSP_ERROR_NOENOUGHPRI	401	无权限：服务器返回 401 时，转成这个错误码
NET_DVR_RTSP_ERROR_ALLOC_RESOURCE	402	分配资源失败
NET_DVR_RTSP_ERROR_PARAMETER	403	参数错误
NET_DVR_RTSP_ERROR_NO_URL	404	指定的 URL 地址不存在：服务器返回 404 时，转成这个错误码，例如请求不可用的通道号预览、请求不支持子码流的通道预览
NET_DVR_RTSP_ERROR_FORCE_STOP	406	用户中途强行退出
NET_DVR_RTSP_GETPORTFAILED	407	获取 RTSP 端口错误
NET_DVR_RTSP_DESCRIPTERROR	410	RTSP DESCRIBE 交互错误

NET_DVR_RTSP_DESCRIBESENDBTIMEOUT	411	RTSP DESCRIBE 发送超时
NET_DVR_RTSP_DESCRIBESENDERERROR	412	RTSP DESCRIBE 发送失败
NET_DVR_RTSP_DESCRIBERECEVTIMEOUT	413	RTSP DESCRIBE 接收超时
NET_DVR_RTSP_DESCRIBERECEVDATALOST	414	RTSP DESCRIBE 接收数据错误
NET_DVR_RTSP_DESCRIBERECEVERROR	415	RTSP DESCRIBE 接收失败
NET_DVR_RTSP_DESCRIBESERVERERR	416	RTSP DESCRIBE 服务器返回错误状态。例如服务器返回 400，可能是不支持子码流
NET_DVR_RTSP_SETUPERROR	420	RTSP SETUP 交互错误，一般是服务器返回的码流地址无法连接上，或者被服务器拒绝。（老版本的 SDK 可能返回错误号 419，为同样的错误原因）
NET_DVR_RTSP_SETUPSENDBTIMEOUT	421	RTSP SETUP 发送超时
NET_DVR_RTSP_SETUPSENDERERROR	422	RTSP SETUP 发送错误
NET_DVR_RTSP_SETUPRECEVTIMEOUT	423	RTSP SETUP 接收超时
NET_DVR_RTSP_SETUPRECEVDATALOST	424	RTSP SETUP 接收数据错误
NET_DVR_RTSP_SETUPRECEVERROR	425	RTSP SETUP 接收失败
NET_DVR_RTSP_OVER_MAX_CHAN	426	超过服务器最大连接数，或者服务器资源不足，服务器返回 453 时，转成这个错误码
NET_DVR_RTSP_SETUPSERVERERR	427	RTSP SETUP 服务器返回错误状态
NET_DVR_RTSP_PLAYERERROR	430	RTSP PLAY 交互错误
NET_DVR_RTSP_PLAYSENDBTIMEOUT	431	RTSP PLAY 发送超时
NET_DVR_RTSP_PLAYSENDERERROR	432	RTSP PLAY 发送错误
NET_DVR_RTSP_PLAYRECEVTIMEOUT	433	RTSP PLAT 接收超时
NET_DVR_RTSP_PLAYRECEVDATALOST	434	RTSP PLAY 接收数据错误
NET_DVR_RTSP_PLAYRECEVERROR	435	RTSP PLAY 接收失败
NET_DVR_RTSP_PLAYSERVERERR	436	RTSP PLAY 服务器返回错误状态
NET_DVR_RTSP_TEARDOWNERROR	440	RTSP TEARDOWN 交互错误
NET_DVR_RTSP_TEARDOWNSENDBTIMEOUT	441	RTSP TEARDOWN 发送超时
NET_DVR_RTSP_TEARDOWNSENDERERROR	442	RTSP TEARDOWN 发送错误
NET_DVR_RTSP_TEARDOWNRECEVTIMEOUT	443	RTSP TEARDOWN 接收超时
NET_DVR_RTSP_TEARDOWNRECEVDATALOST	444	RTSP TEARDOWN 接收数据错误
NET_DVR_RTSP_TEARDOWNRECEVERROR	445	RTSP TEARDOWN 接收失败
NET_DVR_RTSP_TEARDOWNSERVERERR	446	RTSP TEARDOWN 服务器返回错误状态

6.3 软解码库错误码

错误名称	错误值	说明
NET_PLAYM4_NOERROR	500	没有错误

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

6.4 语音对讲库错误码

错误名称	错误值	说明
NET_AUDIOINTERCOM_OK	600	没有错误
NET_AUDIOINTECOM_ERR_NOTSUPORT	601	不支持
NET_AUDIOINTECOM_ERR_ALLOC_MEMORY	602	内存申请错误
NET_AUDIOINTECOM_ERR_PARAMETER	603	参数错误
NET_AUDIOINTECOM_ERR_CALL_ORDER	604	调用次序错误

NET_AUDIOINTECOM_ERR_FIND_DEVICE	605	未发现设备
NET_AUDIOINTECOM_ERR_OPEN_DEVICE	606	不能打开设备
NET_AUDIOINTECOM_ERR_NO_CONTEXT	607	设备上下文出错
NET_AUDIOINTECOM_ERR_NO_WAVFILE	608	WAV 文件出错
NET_AUDIOINTECOM_ERR_INVALID_TYPE	609	无效的 WAV 参数类型
NET_AUDIOINTECOM_ERR_ENCODE_FAIL	610	编码失败
NET_AUDIOINTECOM_ERR_DECODE_FAIL	611	解码失败
NET_AUDIOINTECOM_ERR_NO_PLAYBACK	612	播放失败
NET_AUDIOINTECOM_ERR_DENOISE_FAIL	613	降噪失败
NET_AUDIOINTECOM_ERR_UNKOWN	619	未知错误