

设备(DVR&NVR)

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	14
3.1 SDK 基本调用的主要流程	14
3.2 IP 通道相关说明	16
3.3 实时预览模块流程	17
3.4 回放和下载模块流程	18
3.5 参数配置模块流程	19
3.6 远程设备维护模块流程	20
3.7 语音对讲转发模块流程	21
3.8 报警模块流程	22
3.8.1 报警（布防）流程	22
3.8.2 报警（监听）流程	23
3.9 透明通道模块流程	24
4 函数调用实例	25
4.1 IP 通道资源配置示例代码	25
4.2 预览模块的示例代码	29
4.3 回放和下载模块的示例代码	34
4.4 参数配置模块的示例代码	42
4.5 远程设备维护模块的示例代码	44
4.6 语音对讲转发模块的示例代码	46
4.7 报警模块的示例代码	47
4.8 透明通道模块的示例代码	52
5 函数说明	55
5.1 SDK 初始化	55
5.1.1 初始化 SDK NET_DVR_Init	55
5.1.2 释放 SDK 资源 NET_DVR_Cleanup	55
5.2 SDK 本地功能	55
SDK 本地参数配置	55
5.2.1 获取 SDK 本地参数 NET_DVR_GetSDKLocalCfg	55
5.2.2 设置 SDK 本地参数 NET_DVR_SetSDKLocalCfg	56
连接和接收超时时间及重连设置	57
5.2.3 设置网络连接超时时间和连接尝试次数 NET_DVR_SetConnectTime	57
5.2.4 设置重连功能 NET_DVR_SetReconnect	57
5.2.5 设置接收超时时间 NET_DVR_SetRecvTimeOut	57
多网卡绑定	57
5.2.6 获取所有 IP，用于支持多网卡接口 NET_DVR_GetLocalIP	57
5.2.7 设置 IP 绑定 NET_DVR_SetValidIP	58
SDK 版本、状态和能力	58

5.2.8	获取 SDK 的版本号和 build 信息 NET_DVR_GetSDKBuildVersion.....	58
5.2.9	获取当前 SDK 的状态信息 NET_DVR_GetSDKState.....	58
5.2.10	获取当前 SDK 的功能信息 NET_DVR_GetSDKAbility.....	58
	SDK 启用写日志.....	59
5.2.11	启用写日志文件 NET_DVR_SetLogToFile	59
	异常消息回调.....	59
5.2.12	注册异常消息回调函数 NET_DVR_SetExceptionCallBack_V30	59
	获取错误信息.....	62
5.2.13	返回最后操作的错误码 NET_DVR_GetLastError.....	62
5.2.14	返回最后操作的错误码信息 NET_DVR_GetErrorMsg.....	62
5.3	用户注册.....	62
5.3.1	激活设备 NET_DVR_ActivateDevice	62
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.....	62
5.3.3	用户注册设备 NET_DVR_Login_V40	63
5.3.4	用户注销 NET_DVR_Logout	63
5.4	获取设备能力集.....	64
5.4.1	获取设备能力集 NET_DVR_GetDeviceAbility	64
5.4.2	获取设备能力集(标准协议) NET_DVR_GetSTDAbility	65
5.5	实时预览.....	66
5.5.1	实时预览 NET_DVR_RealPlay_V40	66
5.5.2	停止预览 NET_DVR_StopRealPlay	67
5.5.3	获取预览时用来解码和显示的播放库句柄 NET_DVR_GetRealPlayerIndex.....	67
5.6	强制 I 帧.....	67
5.6.1	强制 I 帧 NET_DVR_RemoteControl.....	67
5.7	预览显示视频参数配置.....	68
5.7.1	获取预览视频显示参数 NET_DVR_ClientGetVideoEffect.....	68
5.7.2	获取预览视频显示参数 NET_DVR_GetVideoEffect.....	68
5.7.3	设置预览视频显示参数 NET_DVR_ClientSetVideoEffect	68
5.7.4	设置预览视频显示参数 NET_DVR_SetVideoEffect.....	69
5.8	预览画面叠加字符和图像.....	69
5.8.1	预览画面叠加字符和图像, Linux 下无此接口 NET_DVR_RigisterDrawFun	69
5.9	预览时播放声音控制.....	70
5.9.1	设置声音播放模式 NET_DVR_SetAudioMode	70
5.9.2	独占声卡模式下开启声音 NET_DVR_OpenSound.....	70
5.9.3	独占声卡模式下开启声音 NET_DVR_CloseSound.....	70
5.9.4	共享声卡模式下开启声音 NET_DVR_OpenSoundShare	70
5.9.5	共享声卡模式下关闭声音 NET_DVR_CloseSoundShare.....	71
5.9.6	调节播放音量 NET_DVR_Volume.....	71
5.10	实时预览数据捕获.....	71
5.10.1	注册回调函数, 捕获实时码流数据 NET_DVR_SetRealDataCallBack.....	71
5.10.2	注册回调函数, 捕获实时码流数据 (标准码流) NET_DVR_SetStandardDataCallBack.....	72
5.10.3	捕获数据并保存到指定的文件中 NET_DVR_SaveRealData	73
5.10.4	停止数据捕获 NET_DVR_StopSaveRealData	73
5.11	预览抓图.....	73

5.11.1	设置抓图模式 NET_DVR_SetCapturePictureMode	73
5.11.2	预览时, 单帧数据捕获并保存成图片 NET_DVR_CapturePicture	73
5.12	设备抓图	74
5.12.1	单帧数据捕获并保存成 JPEG 图片 NET_DVR_CaptureJPEGPicture	74
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	74
5.13	云台控制	75
	云台控制操作	75
5.13.1	云台控制操作 (需先启动图像预览) NET_DVR_PTZControl	75
5.13.2	云台控制操作 (不用启动图像预览) NET_DVR_PTZControl_Other	76
5.13.3	带速度的云台控制操作 (需先启动图像预览) NET_DVR_PTZControlWithSpeed	76
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	77
	云台预置点操作	77
5.13.5	云台预置点操作, 需先启动预览 NET_DVR_PTZPreset	77
5.13.6	云台预置点操作 NET_DVR_PTZPreset_Other	78
	云台巡航操作	78
5.13.7	云台巡航操作, 需先启动预览 NET_DVR_PTZPCruise	78
5.13.8	云台巡航操作 NET_DVR_PTZPCruise_Other	79
	云台轨迹操作	79
5.13.9	云台轨迹操作, 需先启动预览 NET_DVR_PTZTrack	79
5.13.10	云台轨迹操作 NET_DVR_PTZTrack_Other	80
	透明云台控制	80
5.13.11	透明云台操作, 需先启动预览 NET_DVR_TransPTZ	80
5.13.12	透明云台操作 NET_DVR_TransPTZ_Other	81
	云台区域缩放控制	81
5.13.13	云台图象区域选择放大或缩小 NET_DVR_PTZSelZoomIn	81
5.13.14	云台图象区域选择放大或缩小 NET_DVR_PTZSelZoomIn_Ex	81
5.14	录像文件回放、下载、锁定及备份	82
	获取通道录像起止时间	82
5.14.1	获取通道录像起止时间 NET_DVR_InquiryRecordTimeSpan	82
	即时刷新录像索引	82
5.14.2	即时刷新录像索引 NET_DVR_UpdateRecordIndex	82
	月历录像查询	82
5.14.3	获取月历录像分布 NET_DVR_GetDeviceConfig	82
	录像文件的查找	83
5.14.4	根据文件类型、时间查找设备录像文件 NET_DVR_FindFile_V40	83
5.14.5	逐个获取查找到的文件信息 NET_DVR_FindNextFile_V40	84
5.14.6	关闭文件查找, 释放资源 NET_DVR_FindClose_V30	84
	按事件查找录像文件	84
5.14.7	根据事件查找录像文件 NET_DVR_FindFileByEvent_V40	84
5.14.8	逐个获取查找到的事件录像信息 NET_DVR_FindNextEvent_V40	85
5.14.9	关闭文件查找, 释放资源 NET_DVR_FindClose_V30	85
	区域移动侦测智能搜索	85
5.14.10	开始智能搜索 NET_DVR_SmartSearch_V40	85
5.14.11	逐个获取查找到的智能录像信息 NET_DVR_SearchNextInfo	86

5.14.12	停止智能搜索 NET_DVR_StopSearch	86
	回放（正放或倒放）录像文件	87
5.14.13	按文件名回放录像文件 NET_DVR_PlayBackByName	87
5.14.14	按时间回放录像文件 NET_DVR_PlayBackByTime_V40	87
5.14.15	按文件名倒放录像文件 NET_DVR_PlayBackReverseByName	88
5.14.16	按时间倒放录像文件 NET_DVR_PlayBackReverseByTime_V40	89
5.14.17	控制录像回放的状态 NET_DVR_PlayBackControl_V40	89
5.14.18	停止回放录像文件 NET_DVR_StopPlayBack	92
	回放录像文件时的数据捕获	92
5.14.19	捕获回放的录像数据，并保存成文件 NET_DVR_PlayBackSaveData	92
5.14.20	停止保存录像数据 NET_DVR_StopPlayBackSave	92
5.14.21	注册回调函数，捕获录像数据 NET_DVR_SetPlayDataCallBack_V40	92
	录像标签的添加和删除	93
5.14.22	添加录像标签 NET_DVR_InsertRecordLabel	93
5.14.23	修改录像标签 NET_DVR_ModifyRecordLabel	94
5.14.24	删除录像标签 NET_DVR_DelRecordLabel	94
	录像标签的查找	94
5.14.25	搜索录像标签 NET_DVR_FindRecordLabel	94
5.14.26	逐个获取搜索到的录像标签 NET_DVR_FindNextLabel	94
5.14.27	停止搜索录像标签 NET_DVR_StopFindLabel	95
	回放的其他操作	95
5.14.28	获取录像回放时显示的 OSD 时间 NET_DVR_GetPlayBackOsdTime	95
5.14.29	录像回放时抓图，并保存在文件中 NET_DVR_PlayBackCaptureFile	95
5.14.30	刷新显示回放窗口 NET_DVR_RefreshPlay	95
5.14.31	获取回放时用来解码显示的播放库句柄 NET_DVR_GetPlayBackPlayerIndex	96
	下载录像文件	96
5.14.32	按文件名下载录像文件 NET_DVR_GetFileByName	96
5.14.33	按时间下载录像文件 NET_DVR_GetFileByTime_V40	96
5.14.34	控制录像下载的状态 NET_DVR_PlayBackControl_V40	97
5.14.35	停止下载录像文件 NET_DVR_StopGetFile	98
5.14.36	获取当前下载录像文件的进度 NET_DVR_GetDownloadPos	98
	录像文件锁定和解锁	99
5.14.37	按文件名锁定录像文件 NET_DVR_LockFileByName	99
5.14.38	按文件名解锁录像文件 NET_DVR_UnlockFileByName	99
5.14.39	流 ID 方式对某一时间段录像文件进行加锁 NET_DVR_LockStreamFileByTime	99
5.14.40	流 ID 方式对某一时间段录像文件进行解锁 NET_DVR_UnlockStreamFileByTime	99
	备份文件	100
5.14.41	获取设备磁盘列表 NET_DVR_GetDiskList	100
5.14.42	按文件名备份录像文件 NET_DVR_BackupByName	100
5.14.43	按时间段备份录像文件 NET_DVR_BackupByTime	100
5.14.44	获取备份的进度 NET_DVR_GetBackupProgress	101
5.14.45	停止备份 NET_DVR_StopBackup	101
5.15	图片的查找、回放及备份	102
	查找图片	102

5.15.1	根据类型和时间查找图片 NET_DVR_FindPicture	102
5.15.2	逐个获取查找到的图片 NET_DVR_FindNextPicture_V40	102
5.15.3	关闭图片查找, 释放资源 NET_DVR_CloseFindPicture	103
	图片智能检索 (后检索)	103
5.15.4	开始智能图片检索 NET_DVR_SmartSearchPicture	103
5.15.5	逐个获取搜索到的智能图片信息 NET_DVR_FindNextSmartPicture	103
5.15.6	关闭图片智能检索, 释放资源 NET_DVR_CloseSmartSearchPicture	104
	回放 (下载) 图片	104
5.15.7	图片回放 NET_DVR_GetPicture_V30	104
	备份图片	104
5.15.8	备份图片 NET_DVR_BackupPicture	104
5.15.9	获取备份的进度 NET_DVR_GetBackupProgress	104
5.15.10	停止备份 NET_DVR_StopBackup	105
5.16	布防、撤防	106
	设置报警等信息上传的回调函数	106
5.16.1	注册回调函数, 接收报警消息 NET_DVR_SetDVRMessageCallBack_V30	106
	布防撤防	107
5.16.2	建立报警上传通道, 获取报警等信息 NET_DVR_SetupAlarmChan_V41	107
5.16.3	撤销报警上传通道 NET_DVR_CloseAlarmChan_V30	107
5.17	监听报警	107
5.17.1	启动监听, 接收设备主动上传的报警等信息 NET_DVR_StartListen_V30	107
5.17.2	停止监听 (支持多线程) NET_DVR_StopListen_V30	109
5.18	语音对讲、转发及广播	109
	语音对讲	109
5.18.1	启动语音对讲 NET_DVR_StartVoiceCom_V30	109
5.18.2	设置语音对讲客户端的音量 NET_DVR_SetVoiceComClientVolume	110
5.18.3	停止语音对讲或者语音转发 NET_DVR_StopVoiceCom	110
	语音转发	111
5.18.4	启动语音转发, 获取编码后的音频数据 NET_DVR_StartVoiceCom_MR_V30	111
5.18.5	转发语音数据 NET_DVR_VoiceComSendData	112
5.18.6	停止语音对讲或语音转发 NET_DVR_StopVoiceCom	112
	语音广播	112
5.18.7	启动语音广播的 PC 端声音捕获 NET_DVR_ClientAudioStart_V30	112
5.18.8	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	113
5.18.9	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	113
5.18.10	停止语音广播的 PC 端声音捕获 NET_DVR_ClientAudioStop	113
	音频编解码(Windows 32 位系统支持)	114
	G722 音频编解码	114
5.18.11	初始化音频编码 NET_DVR_InitG722Encoder	114
5.18.12	G722 音频编码 NET_DVR_EncodeG722Frame	114
5.18.13	释放音频编码资源 NET_DVR_ReleaseG722Encoder	114
5.18.14	初始化音频解码 NET_DVR_InitG722Decoder	114
5.18.15	G722 音频解码 NET_DVR_DecodeG722Frame	115
5.18.16	释放音频解码资源 NET_DVR_ReleaseG722Decoder	115

G711 音频编解码	115
5.18.17 G711 音频编码 NET_DVR_EncodeG711Frame	115
5.18.18 G711 音频解码 NET_DVR_DecodeG711Frame	116
5.19 远程参数配置	116
通用参数配置	116
5.19.1 获取设备的配置信息 NET_DVR_GetDVRConfig	116
5.19.2 设置设备的配置信息 NET_DVR_SetDVRConfig	121
5.19.3 获取设备的配置信息(标准协议)NET_DVR_GetSTDConfig	126
5.19.4 设置设备的配置信息(标准协议)NET_DVR_SetSTDConfig	126
批量参数配置	127
5.19.5 批量获取配置信息 NET_DVR_GetDeviceConfig	127
5.19.6 批量设置配置信息 NET_DVR_SetDeviceConfig	128
长连接参数配置	130
5.19.7 启动长连接远程配置 NET_DVR_StartRemoteConfig	130
5.19.8 逐个获取查找到的信息 NET_DVR_GetNextRemoteConfig	131
5.19.9 获取长连接配置的状态 NET_DVR_GetRemoteConfigState	131
5.19.10 关闭长连接配置接口所创建的句柄, 释放资源 NET_DVR_StopRemoteConfig	132
远程控制	132
5.19.11 远程控制 NET_DVR_RemoteControl	132
5.19.12 远程控制(标准协议)NET_DVR_STDControl	133
报警输出配置	133
5.19.13 获取设备报警输出 NET_DVR_GetAlarmOut_V30	133
5.19.14 置设备报警输出 NET_DVR_SetAlarmOut	134
RTSP 协议参数配置	134
5.19.15 获取 RTSP 协议参数 NET_DVR_GetRtspConfig	134
5.19.16 设置 RTSP 协议参数 NET_DVR_SetRtspConfig	134
获取设备支持的云台协议	135
5.19.17 获取设备支持的云台协议 NET_DVR_GetPTZProtocol	135
视频输出缩放参数配置	135
5.19.18 获取视频输出缩放信息 NET_DVR_GetScaleCFG_V30	135
5.19.19 设置视频输出缩放参数 NET_DVR_SetScaleCFG_V30	135
5.20 SMART 参数配置	136
SMART 参数配置	136
5.20.1 批量获取配置信息 NET_DVR_GetDeviceConfig	136
5.20.2 批量设置配置信息 NET_DVR_SetDeviceConfig	137
VQD 视频质量诊断	138
5.20.3 批量获取配置信息 NET_DVR_GetDeviceConfig	138
5.20.4 批量设置配置信息 NET_DVR_SetDeviceConfig	139
5.20.5 启动长连接远程配置 NET_DVR_StartRemoteConfig	139
5.20.6 逐个获取查找到的信息 NET_DVR_GetNextRemoteConfig	140
5.20.7 关闭长连接配置接口所创建的句柄, 释放资源 NET_DVR_StopRemoteConfig	141
5.21 硬盘管理	141
5.21.1 远程格式化设备硬盘 NET_DVR_FormatDisk	141
5.21.2 获取格式化硬盘的进度 NET_DVR_GetFormatProgress	141

5.21.3	关闭格式化硬盘句柄, 释放资源 NET_DVR_CloseFormatHandle	142
5.22	零通道预览和配置	142
	参数配置	142
5.22.1	获取设备的配置信息 NET_DVR_GetDVRConfig	142
5.22.2	设置设备的配置信息 NET_DVR_SetDVRConfig	143
	实时预览	143
5.22.3	开启零通道预览 NET_DVR_ZeroStartPlay	143
5.22.4	停止预览 NET_DVR_ZeroStopPlay	144
	其他功能	144
5.22.5	零通道产生一个关键帧 NET_DVR_ZeroMakeKeyFrame	144
5.22.6	零通道预览画面翻页 NET_DVR_ZeroTurnOver	145
5.23	透明通道	145
5.23.1	建立透明通道 NET_DVR_SerialStart	145
5.23.2	通过透明通道向设备串口发送数据 NET_DVR_SerialSend	145
5.23.3	断开透明通道 NET_DVR_SerialStop	146
5.24	向串口发送数据	146
5.24.1	直接向串口发送数据, 不需要建立透明通道 NET_DVR_SendToSerialPort	146
5.24.2	直接向 232 串口发送数据, 不需要建立透明通道 NET_DVR_SendTo232Port	146
5.25	设备手动录像	147
5.25.1	远程手动启动设备录像 NET_DVR_StartDVRRecord	147
5.25.2	远程手动停止设备录像 NET_DVR_StopDVRRecord	147
5.26	远程面板控制	147
5.26.1	远程控制面板上的按键 NET_DVR_ClickKey	147
5.26.2	禁用设备本地面板控制 NET_DVR_LockPanel	149
5.26.3	恢复设备本地面板控制 NET_DVR_UnLockPanel	149
5.27	码流加密	150
5.27.1	设置设备码流加密密钥 NET_DVR_InquestStreamEncrypt	150
5.27.2	获取设备码流加密状态 NET_DVR_InquestGetEncryptState	150
5.28	获取 IPC 协议列表	150
5.28.1	获取设备支持的 IPC 协议表 NET_DVR_GetIPCProtoList	150
5.29	远程扫描和修改 IPC 信息	151
5.29.1	远程扫描获取 IPC 信息列表 NET_DVR_GetSadpInfoList	151
5.29.2	远程修改 IPC 信息 NET_DVR_UpdateSadpInfo	151
5.30	邮件测试	151
5.30.1	测试按已配置的 EMAIL 参数能否收发成功 NET_DVR_StartEmailTest	151
5.30.2	获取邮件测试的进度 NET_DVR_GetEmailTestProgress	151
5.30.3	停止邮件测试 NET_DVR_StopEmailTest	152
5.31	文件上传下载	152
5.31.1	上传文件 NET_DVR_UploadFile_V40	152
5.31.2	获取文件上传的进度和状态 NET_DVR_GetUploadState	153
5.31.3	停止文件上传 NET_DVR_UploadClose	153
5.31.4	开始下载文件 NET_DVR_StartDownload	153
5.31.5	获取文件下载的进度和状态 NET_DVR_GetDownloadState	154
5.31.6	停止文件下载 NET_DVR_StopDownload	154

5.32	IP SAN 文件目录查找	154
5.32.1	查找 IPSAN 文件目录 NET_DVR_FindIpSanDirectory	154
5.32.2	逐个获取查找到的目录信息 NET_DVR_FindNextDirectory	154
5.32.3	停止 IPSAN 文件目录搜索，释放资源 NET_DVR_FindDirectoryClose	155
5.33	设备维护管理	155
	获取设备工作状态	155
5.33.1	获取设备的工作状态 NET_DVR_GetDVRWorkState_V30	155
5.33.2	获取设备运行状态 NET_DVR_GetDeviceStatus	155
5.33.3	设备在线状态检测 NET_DVR_RemoteControl	156
5.33.4	启动设备状态巡检 NET_DVR_StartGetDevState	157
5.33.5	停止设备状态巡检 NET_DVR_StopGetDevState	157
	远程升级	157
5.33.6	设置远程升级时网络环境 NET_DVR_SetNetworkEnvironment	157
5.33.7	远程升级 NET_DVR_Upgrade	158
5.33.8	获取远程升级的进度 NET_DVR_GetUpgradeProgress	158
5.33.9	获取远程升级的状态 NET_DVR_GetUpgradeState	158
5.33.10	获取远程升级的阶段信息 NET_DVR_GetUpgradeStep	158
5.33.11	关闭远程升级句柄，释放资源 NET_DVR_CloseUpgradeHandle	159
	在线升级	159
5.33.12	获取在线升级相关信息 NET_DVR_GetSTDConfig	159
5.33.13	远程控制在线升级 NET_DVR_STDControl	160
	日志查找	161
5.33.14	查找设备的日志信息（可搜索带 S.M.A.R.T 信息的日志）NET_DVR_FindDVRLog_V30	161
5.33.15	逐条获取查找到的日志信息 NET_DVR_FindNextLog_V30	167
5.33.16	释放查找日志的资源 NET_DVR_FindLogClose_V30	168
	远程备份	168
5.33.17	获取设备磁盘列表 NET_DVR_GetDiskList	168
5.33.18	备份统一接口 NET_DVR_Backup	168
5.33.19	获取备份的进度 NET_DVR_GetBackupProgress	169
5.33.20	停止备份 NET_DVR_StopBackup	169
	恢复设备默认参数	170
5.33.21	恢复设备默认参数 NET_DVR_RestoreConfig	170
	导入/导出配置文件	170
5.33.22	导出配置文件 NET_DVR_GetConfigFile_V30	170
5.33.23	导出配置文件 NET_DVR_GetConfigFile	170
5.33.24	导入配置文件 NET_DVR_SetConfigFile_EX	170
5.33.25	导入配置文件 NET_DVR_SetConfigFile	171
	网流量检测	171
5.33.26	开始网络流量检测 NET_DVR_StartNetworkFlowTest	171
5.33.27	停止网络流量检测 NET_DVR_StopNetworkFlowTest	171
	获取 UPNP 端口映射状态	172
5.33.28	获取 UPNP 端口映射状态 NET_DVR_GetUpnpNatState	172
	关机和重启	172
5.33.29	重启设备 NET_DVR_RebootDVR	172

5.33.30	关闭设备 NET_DVR_ShutDownDVR.....	172
6	错误代码及说明	173
6.1	网络通讯库错误码	173
6.2	RTSP 通讯库错误码	178
6.3	软解码库错误码	179
6.4	语音对讲库错误码	180

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 DVR、NVR 支持的功能及相关接口，相关结构体和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：

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

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

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

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、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	HCAalarm.lib	LIB 库文件	HCNetSDKCom 文件夹
		HCAalarm.dll	DLL 库文件	HCNetSDKCom 文件夹
	显示组件	HCDisplay.dll	DLL 库文件	HCNetSDKCom 文件夹
	行业应用管理配置组件	HCIndustry.dll	DLL 库文件	HCNetSDKCom 文件夹
	维护管理配置组件	HCGeneralCfgMgr.lib	LIB 库文件	HCNetSDKCom 文件夹

		HCGeneralCfgMgr.dll	DLL 库文件	HCNetSDKCom 文件夹
RTSP 通讯库		StreamTransClient.dll	DLL 库文件	HCNetSDKCom 文件夹
转封装库		SystemTransform.dll	DLL 库文件	HCNetSDKCom 文件夹
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	
		libhcnet sdk.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就必须加载该项组件。
- **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 DVR、NVR 设备, 不需要使用该库文件。**

2 SDK 版本更新

Version 5.1.3.25 (build20150916)

- 90/80/81/73/72HQH(HGH)-SH V3.3.1
- 新增设备支持 DDNS 国家能力集（对应接口：[NET_DVR_GetSTDAbility](#)）：
能力集类型：NET_DVR_GET_DDNS_COUNTRY_ABILITY。
- 录像配置和查找增加支持码流类型：
 - 1) NET_DVR_RECORD_V40(通道录像参数)中参数 byStreamType(码流类型)新增取值：2-主码流&子码流。
 - 2) NET_DVR_FILECOND_V40(文件查找条件)和 NET_DVR_FINDDATA_V40(文件查找结果)中参数 byStreamType(码流类型)分别新增取值：0xff-全部。
 - 3) NET_DVR_MRD_SEARCH_PARAM(月历录像分布查询条件)使用 1 个保留字节新增参数：
byStreamType(码流类型)。
- NET_DVR_COMPRESSION_INFO_V30 参数 byResolution 增加分辨率类型：111-960*1080(1080p Lite)，112-640*720(half 720p)。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中设备通道输入能力(ChannelInputAbility)扩展，节点 <recordStreamType>新增取值：2-main_and_sub(主码流&子码流)。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中录像相关能力(RecordAbility)扩展，新增节点：
<MDRCondStreamType>(月历搜索条件支持的码流类型)。

Version 5.1.3.16 (build20150826)

- 基线 NVR V3.4.0
- 新增同步 IPC 密码与 NVR 一致功能（对应接口：[NET_DVR_STDCtrl](#)）：
命令：NET_DVR_SYNC_IPC_PASSWD。
- 新增配置字符编码相关处理回调功能（对应接口：[NET_DVR_SetSDKLocalCfg](#)）：
配置类型：NET_SDK_LOCAL_CFG_TYPE_CHAR_ENCODE。
- 新增允许在线升级控制功能（对应接口：[NET_DVR_STDCtrl](#)）：
命令：NET_DVR_SET_ONLINE_UPGRADE。
- 新增在线升级相关信息、状态查询功能（对应接口：[NET_DVR_GetSTDCfg](#)）：
命令：NET_DVR_GET_ONLINEUPGRADE_PROGRESS(在线升级进度)、
NET_DVR_GET_FIRMWARECODE(获取识别码)、
NET_DVR_GET_ONLINEUPGRADE_SERVER(获取升级服务器状态)、
NET_DVR_GET_ONLINEUPGRADE_VERSION(获取新版本信息)、
NET_DVR_GET_RECOMMEN_VERSION(检测是否推荐升级到此版本)。
- 新增能力集（对应接口：[NET_DVR_GetSTDAbility](#)）：
能力集类型：NET_DVR_GET_TRAFFIC_CAP(抓拍相关能力集)、
NET_DVR_GET_ONLINEUPGRADE_ABILITY(在线升级能力集)。
- NET_DVR_PREVIEWINFO(预览参数)中 dwLinkMode(连接方式)新增取值：6- HRUDP(可靠 UDP 传输)。
- NET_DVR_DEVICEINFO_V40(设备参数)使用 1 个保留字节新增参数：byCharEncodeType(字符编码格式)。
- NET_DVR_DIGITAL_CHANNEL_STATE(设备数字通道状态)中的数字通道状态取值新增：
NET_SDK_DC_STATUS_POE_PORT_DETECTING(POE 通道检测中)。
- NET_DVR_VEHICLE_PARA(车辆检测信息)使用 1 个保留字节新增参数：byCountry(国家索引值)。
- 事件触发能力集(EventTriggersCap)新增节点：<BlackListTriggerCap>(黑名单事件触发能力)、

<WhiteListTriggerCap>(白名单事件触发能力)。

- 设备软硬件能力集(DEVICE_SOFTHARDWARE_ABILITY)新增节点: <isSupportSyncIPCPassword>(是否支持同步 IPC 密码)、<isSupportTransferIPC>(是否支持透传 IPC 协议功能)、<supportPreviewHRUDP>(是否支持 HRUDP 预览取流方式)。
- JPEG 抓图能力集(DEVICE_JPEG_CAP_ABILITY)扩展, <SmartPicSearchInfo>(智能图片检索)中节点 <VehicleCond>(车辆检索条件)新增子节点<country>(国家索引)。
- 设备图像参数能力集(DEVICE_VIDEOPIC_ABILITY)扩展, 新增节点: <Sensitivity>(灵敏度节点)。
- 新增异常日志次类型: MINOR_SYNC_IPC_PASSWD、MINOR_EZVIZ_OFFLINE。

Version 5.1.1.3 (build20150318)

- 更新 V40 登录扩展接口:
[NET_DVR_Login_V40](#)。
- 新增设备激活接口:
[NET_DVR_ActivateDevice](#)。
- 新增通过 NVR 激活前端设备功能 (对应接口: [NET_DVR_SetSTDConfig](#)):
命令: NET_DVR_SET_IPDEVICE_ACTIVATED。
- 新增获取数字通道对应设备安全状态功能 (对应接口: [NET_DVR_GetDVRConfig](#)):
命令: NET_DVR_GET_DIGITAL_CHAN_SECURITY_STATUS。
- 新增 NVR 激活前端设备能力集 (对应接口: [NET_DVR_GetSTDAbility](#))
能力集类型: NET_DVR_GET_ACTIVATE_IPC_ABILITY。
- 新增设备激活相关错误码: 250、251、252。

Version 5.0.3.20 (build20150318)

- NVR V3.3.0
- 新增 GB28181 服务器参数配置功能 (对应接口: [NET_DVR_GetSTDConfig](#) 和 [NET_DVR_SetSTDConfig](#)):
命令: NET_DVR_GET_GBT28181_SERVICE_CFG、NET_DVR_SET_GBT28181_SERVICE_CFG。
- 新增 GB28181 服务器能力集 (对应接口: [NET_DVR_GetSTDAbility](#))
能力集类型: NET_DVR_GET_GBT28181_SERVICE_CAPABILITIES。
- NET_DVR_SADPINFO(SADP 搜索的 IPC 信息)使用 32 个保留字节新增参数:
szGB28181DevID[DEV_ID_LEN](GB28181 协议接入时的设备 ID)。
- NET_DVR_IPDEVINFO_V31(IP 设备信息)使用 32 个保留字节新增参数: szDeviceID[DEV_ID_LEN](设备 ID)。
- NET_DVR_DDNSPARA_V30(DDNS 网络应用参数)使用 5 个保留字节新增参数: wCountryID(国家编号)、byStatus(DDNS 运行状态)。
- 设备软硬件能力集(DEVICE_SOFTHARDWARE_ABILITY)新增节点: <DDNSStatus>(DDNS 运行状态能力)、<supportGB28181Service>(支持 GB28181 服务参数配置)。

Version 5.0.2.15 (build20141212)

- DS_96XXN_FX、DS_86XXN_FX V3.2.0
- 按事件查找录像文件接口扩展:
[NET_DVR_FindFileByEvent_V40](#)、[NET_DVR_FindNextEvent_V40](#)。
- 图片查找接口扩展:
[NET_DVR_FindNextPicture_V40](#)。
- 新增智能图片查找接口:
[NET_DVR_SmartSearchPicture](#)、[NET_DVR_FindNextSmartPicture](#)、[NET_DVR_CloseSmartSearchPicture](#)。

- 新增简易智能假日计划配置功能（对应接口：[NET_DVR_GetDeviceConfig](#) 和 [NET_DVR_SetDeviceConfig](#)）：
命令：NET_DVR_GET_SMD_HOLIDAY_HANDLE、NET_DVR_SET_SMD_HOLIDAY_HANDLE。
- 新增获取所有被锁定信息功能（对应接口：[NET_DVR_StartRemoteConfig](#) 和 [NET_DVR_GetNextRemoteConfig](#)）：
命令：NET_DVR_GET_LOCKED_INFO_LIST。
- 新增用户解锁功能（对应接口：[NET_DVR_RemoteControl](#)）：
命令：NET_DVR_UNLOCK_USER。
- 智能搜索扩展：
 - 1) NET_DVR_SMART_SEARCH_PARAM(智能搜索条件)的 bySearchCondType(智能查找类型)新增取值：3-人脸侦测。
 - 2) NET_DVR_AREA_SMARTSEARCH_COND_UNION(智能查找条件联合体)新增结构体参数：NET_DVR_FACEDETECTION_SEARCHCOND。
 - 3) NET_DVR_TRAVERSE_PLANE_SEARCHCOND(越界侦测录像查找条件)和 NET_DVR_INTRUSION_SEARCHCOND(区域入侵录像查找条件)分别使用 37 个保留字节新增参数：byAdvanceType(组合方式)、uAdvanceCond(组合属性联合体)。
- NET_DVR_DISK_RAID_INFO(磁盘 Raid 信息)使用 1 个保留字节新增参数：bySleepStatus(休眠状态)。
- NET_DVR_CORRIDOR_MODE(旋转功能配置)使用 1 个保留字节新增参数：byMirrorMode(镜像方式)。
- NET_DVR_GUARD_CFG(车牌侦测计划配置)使用 424 个保留字节新增参数：dwMaxRelRecordChanNum(最大支持触发的录像通道数)、dwRelRecordChanNum(报警触发的录像通道数实际个数)、dwRelRecordChan[MAX_CHANNUM_V30](报警触发的录像通道)、struHolidayTime[MAX_TIMESEGMENT_V30](假日布防时间)。
- NET_DVR_PHY_DISK_INFO(物理磁盘信息)中 byStatus(硬盘状态)新增取值：9-休眠。
- NET_DVR_CAPTURE_DAY(全天抓图计划)和 NET_DVR_CAPTURE_SCHED(时间段抓图布防计划)中 byCaptureType(抓图类型)新增取值：8-全部事件。

Version 4.3.0.5 (build20140627)

- Netra 平台设备 V3.1.3
- NET_DVR_INTRUSION_SEARCHCOND(区域入侵录像查找条件)使用 12 字节新增参数：struPTZPosInfo(PTZ 坐标信息)。
- NET_DVR_VCA_DETION_CFG(智能侦测参数)使用 32 字节新增参数：struHolidayTime[MAX_TIMESEGMENT_V30](假日布防时间)。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中事件能力(EventAbility)，<VCADetection>新增子节点 <holidaySched>(是否支持假日计划)。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中录像相关能力(RecordAbility)，<TraversingVirtualPlane>和 <FieldDetection>新增子节点：<supportPTZInfo>(检索时是否支持带 PTZ 信息)。
- 新增操作日志的参数类型：PARAM_DETECTION(侦测配置)。
- 新增支持越界侦测和区域侦测配置功能：NET_DVR_GET_TRAVERSE_PLANE_DETECTION、NET_DVR_SET_TRAVERSE_PLANE_DETECTION、NET_DVR_GET_FIELD_DETECTION、NET_DVR_SET_FIELD_DETECTION。
- 复用按流 ID 方式获取录像状态功能（对应接口：对应接口：[NET_DVR_GetDeviceConfig](#) 和 [NET_DVR_SetDeviceConfig](#)）：
命令：NET_DVR_GET_STREAM_RECORD_STATUS、NET_DVR_SET_STREAM_RECORD_STATUS
- 复用流 ID 模式按时间段锁定录像文件功能：
[NET_DVR_LockStreamFileByTime](#)、[NET_DVR_UnlockStreamFileByTime](#)。

Version 4.3.0.2 (build20140510)

- Netra 平台设备 V3.1.2
- 新增参数配置功能（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_CLOUD_STORAGE_CFG(云存储工作模式配置)。
- 新增录像上传功能（对应接口：[NET_DVR_UploadFile_V40](#)）：
命令：UPLOAD_RECORD_FILE。
- NET_DVR_COMPRESSION_INFO_V30(码流压缩参数)中 byResolution(分辨率)新增取值：67- 2592*1944。
- NET_DVR_PREVIEW_SWITCH_CFG(本地预览切换参数)中 byPreviewNumber(预览画面分割模式)新增取值：9- 64 画面。
- NET_DVR_DISK_QUOTA(磁盘配额信息)，使用 2 个保留字节新增参数：wStoragePeriod(录像存储周期)，byQuotaType(磁盘配额类型)新增取值：2- 按比例、3- 按时间。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中录像相关能力(RecordAbility)扩展，新增节点：
<RecordLockTime>(录像锁定时间)。
- 设备数字通道能力集(DEVICE_DYNCHAN_ABILITY)，新增节点：<ANRCfgReconnect>(ANR 功能配置后设备是否需要重连)。
- 设备软硬件能力集(DEVICE_SOFTWARE_ABILITY)扩展，新增节点：<QuotaByTime>(是否支持磁盘配额按时间分配)、<CloudStorageModeChangeReboot>(云存储模式切换是否需要重启)、<VCADetection>(是否支持智能侦测)。
- 新增异常日志次类型：0x48。
- 新增操作日志次类型：0x211c、0x211d、0x2120~0x2126。
- 新增错误号：822、823。

Version 4.2.8.7 (build20140408)

- 经济型 NVR(DS_77XXN_E4 等) V3.0.3
- 新增设备类型：DS_76XXN_EX、DS_77XXN_E4、DS_86XXN_E8。
- 异常参数配置，NET_DVR_EXCEPTION_V30 的参数 struExceptionHandleType 和 NET_DVR_EXCEPTION_V40 的参数 struExceptionHandle，新增取值定义：数组 15-POE 供电异常。
- 报警信息上传，NET_DVR_ALARMINFO_V30 和 NET_DVR_ALARM_FIXED_HEADER 的参数 dwAlarmType 新增节点取值定义：16-POE 供电异常。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中报警事件处理能力(EventAbility)的节点<exceptionType>新增取值：POEPoweException。

Version 4.2.8.1 (build20140115)

- 高新性能 NVR(256 路 NVR) V3.0.2
- 新增设备类型：DS_96XXX_N_E(高性能 NVR，256 路)
- 参数配置扩展（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_ALARMINCFG_V40(报警输入配置)、NET_DVR_USER_V40(用户参数配置)、
NET_DVR_EXCEPTION_V40(异常参数配置)、NET_DVR_HDGROUP_CFG_V40(盘组管理配置)，分别扩展报警输入、用户、异常类型、盘组个数。
- IP 报警输入输出接入信息获取功能扩展（对应接口：[NET_DVR_GetDVRConfig](#)）：
NET_DVR_IPALARMINCFG_V40、NET_DVR_IPALARMOUTCFG_V40，扩展报警输入输出个数。
- 新增批量配置功能（对应接口：[NET_DVR_GetDeviceConfig](#) 和 [NET_DVR_SetDeviceConfig](#)）：
命令：NET_DVR_GET_WORK_STATUS(获取设备工作状态)、NET_DVR_GET_JPEG_CAPTURE_CFG_V40(获取

设备抓图配置)、NET_DVR_SET_JPEG_CAPTURE_CFG_V40(设置设备抓图配置)。

- 新增报警信息上传（对应接口：[NET_DVR_SetDVRMessageCallBack_V30](#)和[NET_DVR_StartListen_V30](#)）：
NET_DVR_ALARMINFO_V40(移动侦测、视频丢失、遮挡、IO 信号量等报警信息，报警数据可变量)。
- NET_DVR_SEARCH_EVENT_PARAM(按事件搜索的条件参数)中联合体 uSeniorParam 新增结构参数：
struAlarmParamByValue(按值表示的报警输入参数)、struMotionParamByValue(按值表示的移动侦测参数)、struVcaParamByValue(按值表示的行为分析参数)。
- NET_DVR_DEVICECFG_V40(设备参数配置)使用 6 个保留字节新增参数：byStartAlarmInNo(模拟报警输入起始号)、byStartAlarmOutNo(模拟报警输出起始号)、byStartIPAlarmInNo(IP 报警输入起始号)、byStartIPAlarmOutNo(IP 报警输出起始号)、byHighIPChanNum(数字通道个数高 8 位)。
- NET_DVR_PREVIEW_SWITCH_COND(本地预览输出接口参数)中 byVideoOutType(视频输出接口类型)新增取值：7-辅助 HDMI、8-扩展 HDMI1、9-扩展 HDMI2、10-扩展 HDMI3、11-扩展 HDMI4。
- NET_DVR_AUXOUTCFG(辅助输出参数配置)中 dwAlarmOutChan(报警弹出画面的输出通道)新增取值：5-HDMI、6-VGA、7- 辅助 HDMI1、8-扩展 HDMI1、9-扩展 HDMI2、10-扩展 HDMI3、11-扩展 HDMI4。
- NET_DVR_DIGITAL_CHANNEL_STATE(设备数字通道状态)使用 384 字节新增：
byDigitalAudioChanTalkStateEx[192](数字语音通道的对讲状态)、byDigitalChanStateEx[192](数字通道的连接状态)。
- 直接从设备取流配置扩展，支持选择取流协议(TCP、UDP、多播等)、码流类型(主码流、子码流)和设备厂家类型：NET_DVR_STREAM_MODE(取流模式)中 byGetStreamType(取流方式)新增取值：6- 直接从设备取流(扩展)；NET_DVR_GET_STREAM_UNION(不同取流方式联合体)新增结构：
NET_DVR_IPCHANINFO_V40(直接从设备取流扩展)。
- 新增异常日志次类型：0x46、0x47。

Version 4.2.7.6 (build20131231)

- Netra 平台设备 V3.1.0
- 新增参数配置功能（对应接口：[NET_DVR_GetDVRConfig](#)和[NET_DVR_SetDVRConfig](#)）：
NET_DVR_VIDEO_AUDIOIN_CFG(视频的音频输入配置，可绑定语音对讲通道)、NET_DVR_CMS_PARAM(平台参数配置)。
- 新增批量配置功能（对应接口：[NET_DVR_GetDeviceConfig](#)和[NET_DVR_SetDeviceConfig](#)）：
命令：NET_DVR_GET_VCA_DETION_CFG(获取智能侦测参数)、NET_DVR_SET_VCA_DETION_CFG(设置智能侦测参数)。
- 新增回放转码命令（对应接口：[NET_DVR_PlayBackControl_V40](#)）：NET_DVR_PLAY_CONVERT。
- 新增报警信息上传（对应接口：[NET_DVR_SetDVRMessageCallBack_V30](#)和[NET_DVR_StartListen_V30](#)）：
NET_DVR_ALARM_HOT_SPARE(N+1 热备异常报警)。
- NET_DVR_PREVIEW_SWITCH_CFG(本地预览切换参数配置)中 byPreviewNumber(画面分割模式)新增取值：6- 25 画面、7- 32 画面、8- 36 画面。
- NET_DVR_MRD_SEARCH_RESULT(月历录像分布查询结果)使用 31 个保留字节新增参数：
byHasEventRecode(有无事件录像)。
- NET_DVR_BACKUP_LOG_PARAM(日志备份参数)使用 1 个保留字节新增参数：byAllLogBackUp(按磁盘号备份还是全部备份)。
- 获取备份的进度 NET_DVR_GetBackupProgress 中 pState 新增进度值：BACKUP_NO_LOG_FILE(硬盘中无日志)。
- NET_DVR_RECORDDAY(全天录像参数)中 byRecordType(录像类型)新增取值：19-智能侦测（越界侦测|区域入侵|进入区域|离开区域|人脸识别）。
- NET_DVR_CAPTURE_DAY(全天抓图计划参数)、NET_DVR_CAPTURE_SCHED(时间段抓图计划参

数)byCaptureType(抓图类型)新增取值： 7-智能侦测抓图。

- NET_DVR_FILECOND_V40(录像查找条件)中 dwFileType(文件类型)和 NET_DVR_FINDDATA_V40(录像查找结果)中 byFileType(文件类型)分别新增取值： 19-智能侦测（越界侦测|区域入侵|进入区域|离开区域|人脸识别）。
- NET_DVR_FIND_PICTURE_PARAM(图片查找条件)byFileType(图片类型)新增取值： 0x11-设备本地回放时截图、0x12-智能侦测。
- NET_DVR_ALARMINFO_V30(设备报警信息)的 dwAlarmType(报警类型)新增取值： 15-智能侦测。
- NET_DVR_SEARCH_EVENT_PARAM(按事件搜索的条件参数)中，使用 1 个保留字节新增参数： byValue(通道号按值还是按位表示)；联合体 uSeniorParam 新增结构参数： struVCADetectByBit(按位表示的智能侦测参数)、struVCADetectByValue(按值表示的智能侦测参数)； MAIN_EVENT_TYPE(事件查找主类型)新增枚举类型： EVENT_VCA_DETECTION(智能侦测)，对应次类型取值： VCA_DETECTION_MINOR_TYPE。
- NET_DVR_SMART_SEARCH_PARAM(智能搜索条件参数)中，使用 1 个保留字节新增参数： bySearchCondType(智能查找类型)， byMotionScope[64][96]更改为 NET_DVR_AREA_SMARTSEARCH_COND_UNION(智能查找条件联合体)，支持查询移动侦测，越界侦测、区域入侵等智能录像。
- NET_SDK_IPC_CFG_FILE_ERR_CODE(导入 IPC 配置文件错误号)新增枚举类型： NET_SDK_IPC_CFG_FILE_ERR_CODE_TRANSPORT_PROTOCOL_INVALID(传输协议错误)。
- NET_DVR_COMPRESSION_INFO_V30(码流压缩参数)中 byResolution(分辨率)新增取值： 50-720*720、51-1280*1280、52-2048*768、53-2048*2048、54-2560x2048、55-3072x2048、58-1600*600。
- NET_DVR_DIGITAL_CHANNEL_STATE(数字通道状态)新增状态类型： NET_SDK_DC_STATUS_RESOLUTION_NOT_SUPPORT(IPC 分辨率不支持)。
- 获取阵列一键配置的进度(NET_DVR_FastConfigProcess 中 pState)新增取值： PROCESS_QUICK_SETUP_PD_COUNT(一键配置至少需要三块空闲盘)。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中本地预览切换能力 PreviewSwitchAbility，新增节点： <VideoOutList>(视频本地输出能力)。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中录像相关能力 RecordAbility，新增节点： <monthlyDistributionAndRecodeType>(是否支持月历搜索中的录像类型)、<PlayConvert>(回放转码压缩参数能力)、<TraversingVirtualPlane>(越界侦测后检索能力)、<FieldDetection>(区域入侵侦测后检索能力)。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中通道输入能力 ChannelInputAbility，新增节点： <VCAFindFile>(是否支持智能事件查找)。
- 设备通用能力集(DEVICE_ABILITY_INFO)，报警事件处理能力 EventAbility，新增节点： <supportLogNotCfg>(支持日志但不支持配置)、<VCADetection>(智能侦测处理能力)。
- 设备软硬件能力集(DEVICE_SOFTHARDWARE_ABILITY)，新增节点： <AllBackupLog>(是否支持全部日志导出)、<VoiceTalkAsAudioIn>(是否支持语音对讲绑定为音频输入)， <AlarmInTypeChange>(修改报警输入类型)新增能力类型： 2-需要提示重启。
- 设备编码能力集(DEVICE_ENCODE_ALL_ABILITY_V20)中帧率（<VideoFrameRate>、<VideoFrameRateN>）新增取值： 25-3、26-5、27-7、28-9。
- 新增报警日志次类型： 0x29~0x36
- 新增操作日志次类型： 0x2117~0x211b
- 新增附加信息日志次类型： 0xb6~0xbf
- 新增错误号： NET_SDK_ERR_CHAN_AUDIO_BIND(821)

Version 4.2.3.1 (build20130701)

- Netra 平台设备 V3.0.0

- 新增设备类型：DS_96XX_N_RX
- 新增远程配置功能（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_N_PLUS_ONE_WORK_MODE_CFG(N+1 工作模式配置)、NET_DVR_HD_STATUS(硬盘休眠状态配置)、NET_DVR_RAID_BTTS_CFG(RAID 后台任务速度配置)。
- 新增连接远程配置功能（对应接口 [NET_DVR_StartRemoteConfig](#)）：
命令：NET_DVR_IMPORT_IPC_CFG_FILE(导入 IPC 接入配置文件)、NET_DVR_UPGRADE_IPC(通过 NVR 对 IPC 升级)。
- 新增导出 IPC 接入配置文件功能（对应接口：[NET_DVR_StartDownload](#)）：
命令：NET_SDK_DOWNLOAD_IPC_CFG_FILE。
- 新增支持 VQD 功能（对应接口：[NET_DVR_GetDeviceConfig](#) 和 [NET_DVR_SetDeviceConfig](#)）：
命令：NET_DVR_GET_VQD_LOOP_DIAGNOSE_CFG、NET_DVR_SET_VQD_LOOP_DIAGNOSE_CFG。
- 手动获取 VQD 诊断信息（对应接口：[NET_DVR_StartRemoteConfig](#) 和 [NET_DVR_GetNextRemoteConfig](#)）：
命令：NET_DVR_GET_VQD_DIAGNOSE_INFO。
- VQD 诊断报警上传（对应接口：[NET_DVR_SetDVRMessageCallBack_V30](#) 和 [NET_DVR_StartListen_V30](#)）：
NET_DVR_VQD_DIAGNOSE_INFO。
- 设备通用能力集(DEVICE_ABILITY_INFO)新增能力集类型（对应接口：[NET_DVR_GetDeviceAbility](#)）：
N+1 能力 NPlusOneAbility、磁盘相关能力 HardDiskAbility、IPC 配置文件导入导出能力
IPAccessConfigFileAbility、通道输入能力 ChannellInputAbility。
- 设备新增支持转码码流取流：NET_DVR_MULTI_STREAM_COMPRESSIONCFG(多码流压缩参数)、
NET_DVR_MULTI_STREAM_COMPRESSIONCFG_COND(多码流压缩参数配置条件)中 dwStreamType 新增取值 4-转码码流；NET_DVR_PREVIEWINFO(预览参数)中 dwStreamType 新增取值 3-虚拟码流。
- NET_DVR_FILECOND_V40(文件查找条件)使用 16 个保留字节新增参数：byWorkingDeviceGUID[16](工作机 GUID)，其参数 byFindType 增加类型取值：2- 查询 N+1 录像文件。
- NET_DVR_DIGITAL_CHANNEL_STATE(设备数字通道状态)使用 64 个保留字节新增参数：
byDigitalChanState[64](数字通道状态)。
- NET_DVR_MOTION_V30(移动侦测参数)使用 1 个保留字节新增参数：byEnableDisplay(启用移动侦测高亮显示)。
- NET_DVR_DEVICEINFO_V30(设备参数信息)使用 3 个保留字节新增参数：byStartDChan(起始数字通道号)、
byStartDTalkChan(起始数字对讲通道号)、byHighDChanNum(数字通道个数高 8 位)。
- NET_DVR_EXCEPTION_V30(异常参数)中 struExceptionHandleType[MAX_EXCEPTIONNUM_V30]取数组 10、
11 新增异常类型：10- 行车超速（车载专用），11- 热备异常（N+1 使用）。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中录像相关能力集(RecordAbility)，新增节点：
<findWorkingDeviceRecord>。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中报警事件处理能力集(EventAbility)，节点<exceptionType>
新增取值：spareException。
- 设备数字通道能力集(DEVICE_DYNCHAN_ABILITY)，新增节点：<IPCUpgradeAbility>、
<ChannelRecordStatus>、<DigitalChanState>。
- 设备 RAID 能力集(DEVICE_RAID_ABILITY)，新增节点：<RaidInfo>。
- 新增异常日志次类型：
MINOR_ANR_RECORD_FAIED、MINOR_SPARE_WORK_DEVICE_EXCEPT、MINOR_START_IPC_MAS_FAILED。
- 新增 N+1 功能错误号：803~814。

Version 4.2.2.2(2013-5-21)

- Netra 平台 NVR V2.3.1

- 新增批量参数获取功能（对应接口：[NET_DVR_GetDeviceConfig](#)）：
命令：NET_DVR_GET_MONTHLY_RECORD_DISTRIBUTION(获取月历录像分布)、
NET_DVR_GET_ACCESS_DEVICE_CHANNEL_INFO(获取待接入设备通道信息)
- 设备本地预览切换参数配置（对应接口：[NET_DVR_GetDeviceConfig](#) 和 [NET_DVR_SetDeviceConfig](#)）：
命令：NET_DVR_GET_PREVIEW_SWITCH_CFG、NET_DVR_SET_PREVIEW_SWITCH_CFG
- 扩展文件查找接口：[NET_DVR_FindNextFile_V40](#)，支持日历查询功能
- 设备通用能力集(DEVICE_ABILITY_INFO)新增能力集类型（对应接口：[NET_DVR_GetDeviceAbility](#)）：
录像相关能力 RecordAbility、NVR 前端待接入设备通道能力 AccessDeviceChannelAbility、设备本地预览切换能力 PreviewSwitchAbility
- NET_DVR_FILECOND_V40(文件查找条件)使用 6 个保留字节新增参数：byFindType(存储卷类型)、
byQuickSearch(查询类型)、dwVolumeNum(存档卷号)。
- NET_DVR_VOD_PARA(按时间回放参数)使用 6 个保留字节新增参数：byVolumeType(存储卷类型)、
byVolumeNum(存档卷号)、dwFileIndex(存档卷上的录像文件索引)。

Version 4.2.1.6(2012-12-26)

- NETRA 平台设备 V2.3.0
- 新增设备类型：DS_72XXHF_SV、DS_72XXHW_SV。
- 新增配置功能（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_WD1_CFG(WD1 使能配置)、~~错误!超链接引用无效。~~(录像打包参数)。
- 新增批量配置功能（对应接口：[NET_DVR_GetDeviceConfig](#) 和 [NET_DVR_SetDeviceConfig](#)）：
NET_DVR_FTPCFG_V40(FTP 扩展配置)。
- 新增远程控制功能（对应接口：[NET_DVR_RemoteControl](#)）：
NET_DVR_CMD_TRIGGER_PERIOD_RECORD_PARA。
- 新增 NAS 目录查找功能（新增接口：[NET_DVR_StartRemoteConfig](#)、[NET_DVR_GetNextRemoteConfig](#)、
[NET_DVR_GetRemoteConfigState](#)、[NET_DVR_StopRemoteConfig](#)）：
对应命令：NET_DVR_FIND_NAS_DIRECTORY。
- 新增 IPSAN 目录查找接口：[NET_DVR_FindIpSanDirectory](#)、[NET_DVR_FindNextDirectory](#)、
[NET_DVR_FindDirectoryClose](#)。
- 新增日志类型：MINOR_RUN_STATUS_INFO、MINOR_REMOTE_DELETE_HDISK、
MINOR_REMOTE_LOAD_HDISK、MINOR_REMOTE_UNLOAD_HDISK、MINOR_LOCAL_OPERATE_LOCK、
MINOR_LOCAL_OPERATE_UNLOCK。
- NET_DVR_SINGLE_HD(设备硬盘信息)中 dwHdStatus(硬盘状态)新增取值：15-正在删除（网络硬盘）。
- NET_DVR_PU_STREAM_URL(URL 取流配置)使用 4 个保留字节增加参数：byTransPortocol(传输协议类型)、
wIPID(设备 ID 号)和 byChannel(设备通道号)。
- NET_DVR_GET_STREAM_UNION(不同取流方式联合体)新增通过 hkDDNS 连接设备然后从设备取流的方式：
NET_DVR_HKDDNS_STREAM struHkDDNSStream。

Version 4.2.1.6(2012-12-25)

- DS-6700 系列 DVS V1.0.0
- 新增设备类型：DS_67XXHF_SATA、DS_67XXHW、DS_67XXHW_SATA、DS_67XXHF
- 新增配置功能（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_BONJOUR_CFG(Bonjour 参数配置)、NET_DVR SOCKS_CFG(SOCKS 参数配置)、
NET_DVR_QOS_CFG(QoS 参数配置)、NET_DVR_HTTPS_CFG(HTTPS 参数配置)。
- 新增远程控制功能（对应接口：[NET_DVR_RemoteControl](#) 和 [NET_DVR_GetDeviceConfig](#)）：

命令：NET_DVR_CREATE_CERT(创建证书)、NET_DVR_DELETE_CERT(删除证书)。

- 新增获取证书信息的功能（对应接口：[NET_DVR_GetDeviceConfig](#)）：

命令：NET_DVR_GET_CERT

- 新增上传证书功能（对应接口：NET_DVR_UploadFile）：

命令：UPLOAD_CERTIFICATE

- 新增文件下载接口：[NET_DVR_StartDownload](#)、[NET_DVR_GetDownloadState](#)、[NET_DVR_StopDownload](#)，
下载证书命令：NET_SDK_DOWNLOAD_CERT。
- NET_DVR_UPNP_CFG、NET_DVR_UPNP_PORT 复用为：NET_DVR_NAT_CFG、NET_DVR_NAT_PORT。

Version 4.1.9.3（2012-10-30）

- Netra 平台 NVR V2.2.2
- 新增远程控制功能（对应接口：[NET_DVR_RemoteControl](#)）：
命令：NET_DVR_MOUNT_DISK(加载磁盘)、NET_DVR_UNMOUNT_DISK(卸载磁盘)、
NET_DVR_SET_VIDEO_EFFECT(设置通道视频输入图像参数)、NET_DVR_DEL_INVALID_DISK(删除无效磁
盘)。
- 新增日志类型（对应接口：[NET_DVR_FindDVRLog_V30](#)）：
异常日志次类型：MINOR_RECORD_OVERFLOW、MINOR_DSP_ABNORMAL，
操作日志次类型：MINOR_LOCAL_LOAD_HDISK、MINOR_LOCAL_DELETE_HDISK。

Version 4.1.7（2012-8-3）

- 新增设备类型：DS_76XX_N_SE
- 获取视频输入图像效果参数默认值（对应接口：[NET_DVR_GetDeviceConfig](#)）：
命令：NET_DVR_GET_DEFAULT_VIDEO_EFFECT。
- NET_DVR_VIDEO_EFFECT 保留字节数错误修改，并且增加参数 dwHueValue(色度)、dwSharpness(锐度)、
dwDenoising(去噪)。

Version 4.1.6（2012-6-28）

- NETRA 平台 V2.2.0
- 新增设备类型：DS90XXHW_ST、DS72XXHX_SH、DS_92XX_HF_ST、DS_91XX_HF_XT、DS_90XX_HF_XT、
DS_77XXN_ST、DS_95XX_N_ST、DS_85XX_N_ST、DS_96XX_N_XT。
- 新增远程录像文件倒放接口：[NET_DVR_PlayBackReverseByName](#)、[NET_DVR_PlayBackReverseByTime_V40](#)。
- 新增获取通道录像起止时间的接口：[NET_DVR_InquiryRecordTimeSpan](#)。
- 新增即时刷新录像索引的接口：[NET_DVR_UpdateRecordIndex](#)。
- 新增录像文件查找扩展接口：[NET_DVR_FindFile_V40](#)。
- 新增 UPNP 端口映射状态查询接口：[NET_DVR_GetUpnpNatState](#)。
- 新增统一备份接口：[NET_DVR_Backup](#)。
- 新增配置功能（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_DRAWFRAME_DISK_QUOTA_CFG(抽帧通道磁盘配额)、NET_DVR_UPNP_CFG(UPNP 端口映射配
置)、NET_DVR_POE_CFG(POE 参数配置)、NET_DVR_CUSTOM_PROTOCOL(自定义协议配置)、
NET_DVR_STREAM_CABAC(码流压缩性能 CABAC 选项配置)、NET_DVR_POE_CHANNEL_ADD_MODE(POE
通道添加方式配置)、NET_DVR_ESATA_MINISAS_USAGE(eSATA 和 miniSAS 用途配置)。
- 新增配置命令（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_GET_FTPCFG_SECOND、NET_DVR_SET_FTPCFG_SECOND、NET_DVR_GET_HDCFG_V40、
NET_DVR_SET_HDCFG_V40。

- 新增获取设备数字通道状态功能（对应接口：[NET_DVR_GetDVRConfig](#)）：
NET_DVR_DIGITAL_CHANNEL_STATE。
- 新增支持加密配置接口：[NET_DVR_InquestStreamEncrypt](#)、[NET_DVR_InquestGetEncryptState](#)。
- 新增操作日志的次类型：MINOR_LOCAL_TAG_OPT、MINOR_LOCAL_VOUT_SWITCH、
MINOR_STREAM_CABAC。
- [NET_DVR_PlayBackControl V40](#) 的控制命令 dwControlCode 新增：NET_DVR_PLAY_FORWARD(倒放切换为正放)和 NET_DVR_PLAY_REVERSE(正放切换为倒放)。
- NET_DVR_DEVICEINFO_V30(设备参数信息)和 NET_DVR_DEVICECFG_V40(设备参数配置)中 bySupport1 新增能力：bySupport1 & 0x2，表示是否支持下载/回放流量区分。
- NET_DVR_DEVICECFG_V40(设备参数配置)中 byStorageMode(存储模式)新增取值：2- 抽帧模式。
- NET_DVR_BACKUP_TIME_PARAM(按时间备份参数)和 NET_DVR_FIND_LABEL(录像标签查找条件)使用 1 个保留字节增加参数 byDrawFrame(是否抽帧)。

Version 4.0.9 (2011-9-1)

- NETRA 平设备 V2.1.0
- 新增设备类型：DS96XXN_ST、DS86XXN_ST、DS80XXHF_ST、DS90XXHF_ST、DS76XXN_ST。
- 新增参数获取功能（对应接口：[NET_DVR_GetDVRConfig](#)）：
NET_DVR_SYNCHRONOUS_IPC(是否同步前端 IPC 设备参数)、NET_DVR_DEVICE_NET_USING_INFO (获取当前设备网络资源使用情况)
- 新增参数设置功能（对应接口：[NET_DVR_SetDVRConfig](#)）：
NET_DVR_SYNCHRONOUS_IPC(是否同步前端 IPC 设备参数)、NET_DVR_IPC_PASSWD(设置 IPC 用户名密码)、
NET_DVR_IPC_NETCFG(设置前端 IPC 的网络地址)。

Version 4.0.7 (2011-06-13)

- DS-81/91HW-S(T)系列 V2.0.0
- 新增设备类型：DS81XXHW_S、DS81XXHW_ST、DS91XXHW_ST。
- 新增抓图图片分辨率能力集（对应接口：[NET_DVR_GetDeviceAbility](#)）：PIC_CAPTURE_ABILITY。

Version 4.0.5 (2011-06-13)

- NETRA 平台 V2.0.0
- 新增设备类型：DS-91XXHF-ST
- 新增参数配置功能（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_HOLIDAY_PARAM_CFG(假日参数配置)、NET_DVR_HOLIDAY_HANDLE(假日报警处理方式配置)、
NET_DVR_HOLIDAY_RECORD(假日录像配置)、NET_DVR_NETCFG_MULTI(多网卡网络配置)、
NET_DVR_NETWORK_BONDING(网卡 BONDING 配置)、NET_DVR_JPEG_CAPTURE_CFG(抓图参数配置)、
NET_DVR_SCHED_CAPTURECFG(抓图计划配置)、NET_DVR_FTPCFG(FTP 参数配置)、
NET_DVR_DISK_QUOTA_CFG(磁盘配额配置)、NET_DVR_SNMP_CFG(SNMP 协议配置)、
NET_DVR_VIDEO_INPUT_EFFECT(通道视频输入图像参数配置)。
- 新增获取通道设备工作状态的功能（对应接口：[NET_DVR_GetDVRConfig](#)）：NET_DVR_LINK_STATUS。
- 新增网络流量检测功能：[NET_DVR_StartNetworkFlowTest](#)、[NET_DVR_StopNetworkFlowTest](#)。
- 新增加录像标签功能：[NET_DVR_InsertRecordLabel](#)、[NET_DVR_DelRecordLabel](#)、
[NET_DVR_FindRecordLabel](#)、[NET_DVR_FindNextLabel](#)、[NET_DVR_StopFindLabel](#)。
- 新增图片回放备份功能：[NET_DVR_FindPicture](#)、[NET_DVR_FindNextPicture](#)、[NET_DVR_CloseFindPicture](#)、
[NET_DVR_GetPicture V30](#)、[NET_DVR_BackupPicture](#)。

3 函数调用顺序

3.1 SDK 基本调用的主要流程

图 3.1 SDK 基本流程

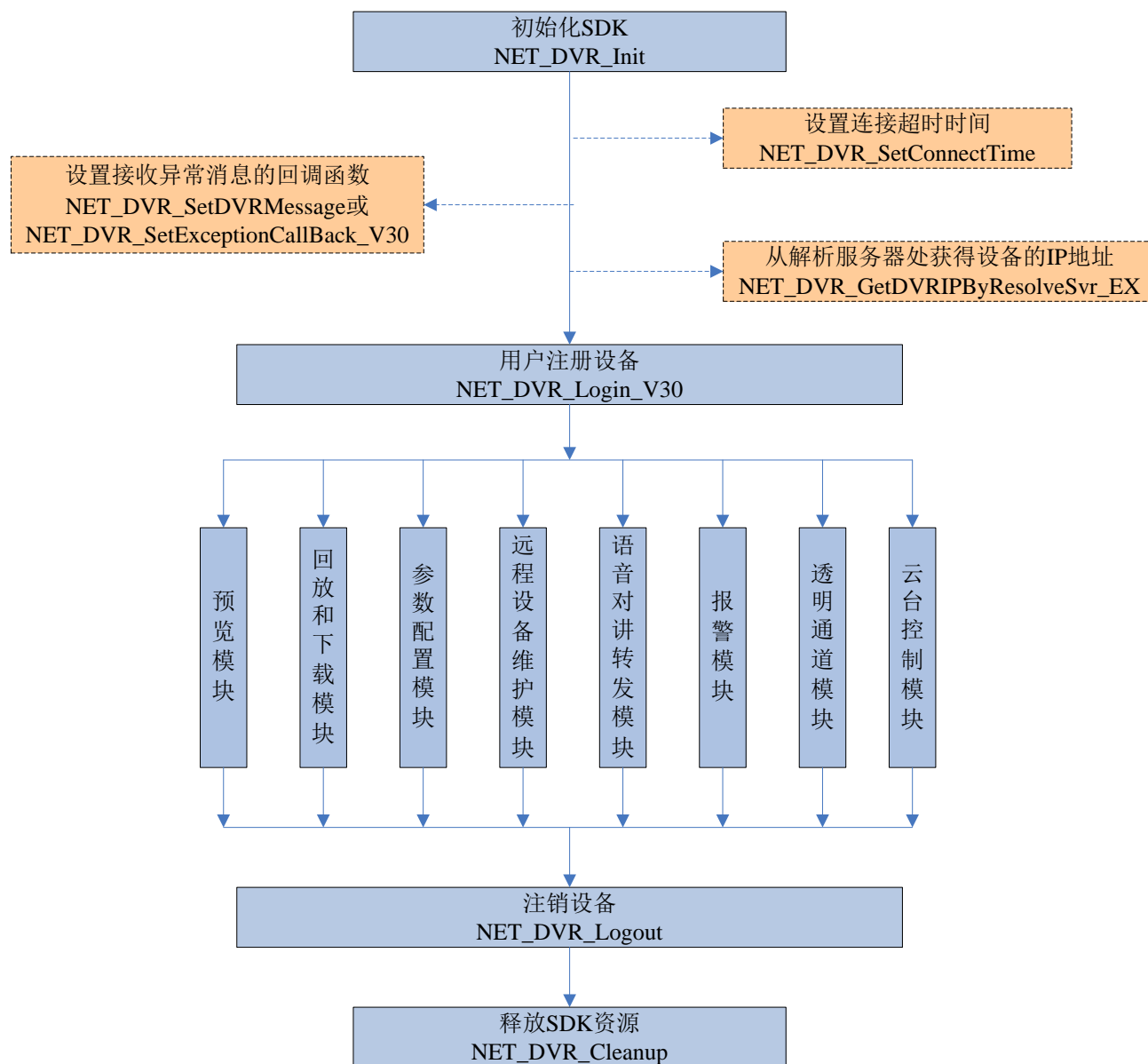


图 3.1 中虚线框的流程是可选部分，不会影响其他流程和模块的功能使用。按实现功能的不同可以分成十个模块，实现每个模块的功能时初始化 SDK、用户注册设备、注销设备和释放 SDK 资源这 4 个流程是必不可少的，解码器功能模块和行为分析功能模块是针对解码器和智能设备的，在该文档里我们不做描述。

- 初始化 SDK ([NET_DVR_Init](#))：对整个网络 SDK 系统的初始化，内存预分配等操作。
- 设置连接超时时间 ([NET_DVR_SetConnectTime](#))：这部分为可选，用于设置 SDK 中的网络连接超时时间，用户可以根据需要设置该值。在不调用此接口设置超时时间的情况下，将采用 SDK 中的默认值。

- 设置接收异常消息的回调函数 ([NET_DVR_SetDVRMessage](#) 或 [NET_DVR_SetExceptionCallBack_V30](#)): 由于 SDK 中大部分模块的功能都是由异步模式实现, 所以我们提供此接口用于接收预览、报警、回放、透明通道和语音对讲等模块发生异常信息。用户可以在初始化 SDK 后就设置该回调函数, 在应用层对各个模块异常消息的接收和处理。
- 从解析服务器获得设备的 IP 地址 ([NET_DVR_GetDVRIPByResolveSvr_EX](#)): 该接口提供一种在仅知道设备名称 (或 DDNS 域名) 或者序列号的情况下, 从解析服务器获得设备 IP 地址的方法。如: 当前设备是通过拨号上网方式获取到动态 IP 地址, 而运行了我公司 IPServer 软件的服务器即为解析服务器或者设备注册到我公司 DDNS 服务器上, 我们可以通过此接口输入服务器的地址、设备的名称或序列号等信息查询该设备的 IP 地址。
- 用户注册设备 ([NET_DVR_Login_V30](#)): 实现用户的注册功能, 注册成功后, 返回的用户 ID 作为其他功能操作的唯一标识, SDK 允许最大注册用户数为 512 个。就设备而言, V3.0 以上版本支持的设备允许有 32 个注册用户名, 而且同时最多允许 128 个用户注册; V3.0 以下版本支持的设备允许有 16 个注册用户名, 而且同时最多允许 128 个用户注册。
- 预览模块: 从设备取实时码流, 解码显示以及播放控制等功能, 同时支持软解码和解码卡解码。具体流程详见[预览模块流程](#)。
- 回放和下载模块: 可以通过按时间和按文件名的方式远程回放或者下载设备的录像文件, 后续可以进行解码或者存储。同时还支持断点续传功能 (需要设备支持)。具体流程详见[回放和下载模块流程](#)。
- 参数配置模块: 设置和获取设备的参数, 主要包括设备参数、网络参数、通道压缩参数、串口参数、报警参数、异常参数、交易信息和用户配置等参数信息。具体流程详见[参数配置模块流程](#)。
- 远程设备维护模块: 实现关闭设备、重启设备、恢复默认值、远程硬盘格式化、远程升级和配置文件导入/导出等维护工作。具体流程详见[远程设备维护模块流程](#)。
- 语音对讲转发模块: 实现和设备的语音数据对讲和语音数据获取, 音频编码格式可以指定。具体流程详见[语音对讲转发模块流程](#)。
- 报警模块: 处理设备上传的各种报警信号。报警分为“布防”和“监听”两种方式, 在采用监听方式并且不需要获取用户 ID 的情况下, 报警模块可以无需进行“用户注册”操作步骤。具体流程详见[报警模块流程](#)。
- 透明通道模块: 透明通道是将 IP 数据报文解析后直接发送到串行口的一种技术。实际上起到了延伸串行设备控制距离的作用。可利用 IP 网络控制多种串行设备, 如控制解码器、矩阵、报警主机、门禁、仪器仪表等串行设备, 对用户来说, 只看到点对点传输, 无须关心网络传输过程, 所以称为串口透明通道。SDK 提供 485 和 232 串口作为透明通道功能, 其中要将 232 串口作为透明通道使用, 首先必须在 232 串口的配置信息 (NET_DVR_RS232CFG) 中将工作模式选为透明通道, 这样 232 串口才可作为透明通道使用。具体流程详见[透明通道模块流程](#)。
- 云台控制模块: 实现对云台的基本操作、预置点、巡航、轨迹和透明云台的控制。SDK 将云台控制分为两种模式: 一种是通过图像预览返回的句柄进行控制; 另一种是无预览限制, 通过用户注册 ID 号进行云台控制。

3.2 IP 通道相关说明

- DVR 视频输入接模拟摄像机，其通道称为模拟通道；混合型 DVR、NVR 等设备支持 IPC 接入，通道称为 IP 通道（或者数字通道），配置相关参数时需调用 IP 接入配置参数来进行资源的获取和重新分配。
- 客户端通过注册设备（[NET_DVR_Login_V30](#)）返回的设备信息 `NET_DVR_DEVICEINFO_V30` 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`）。
- 从 `byStartChan` 到 `byStartChan+byChanNum-1` 对应为模拟通道的通道号，IP 通道的通道号为 `byStartDChan` 到 `byStartDChan+ (byIPChanNum + byHighDChanNum*256) -1`。DVR 只有模拟通道，NVR 只有 IP 通道，混合型 DVR 同时支持模拟通道和 IP 通道。
- 如果设备支持 IP 通道个数大于 0，则可以通过远程参数配置接口 [NET_DVR_GetDVRConfig](#)（配置命令：`NET_DVR_GET_IPPARACFG_V40`）可以获取得到设备详细的 IP 资源信息（`NET_DVR_IPPARACFG_V40`），包括模拟通道是否禁用（`byAnalogChanEnable`）、IP 通道个数（`dwDChanNum`）、IP 通道起始通道号（`dwStartDChan`）、IP 通道取流模式、IP 通道有效状态和在线状态等。通过远程参数配置接口 [NET_DVR_SetDVRConfig](#)（配置命令：`NET_DVR_SET_IPPARACFG_V40`）可对设备进行 IP 资源配置，包括添加、修改、删除 IP 通道等。
- 混合型 DVR 或 NVR 的 IP 报警输入和报警输出的通道是在音视频 IP 通道资源分配好后，由设备自动分配的。如果要对 IP 报警参数进行配置，首先通过命令 `NET_DVR_GET_IPALARMINCFG_V40` 和 `NET_DVR_GET_IPALARMOUTCFG_V40` 获取 IP 报警输入资源（`NET_DVR_IPALARMINCFG_V40`）和 IP 报警输出资源（`NET_DVR_IPALARMOUTCFG_V40`）。然后通过命令 `NET_DVR_SET_ALARMINCFG_V40` 可以配置报警数相关参数（`NET_DVR_ALARMINCFG_V40`），包括报警输入名称、报警器类型、布防时间、联动方式等，通过命令 `NET_DVR_SET_ALARMOUTCFG_V30` 可以配置报警输出相关参数（`NET_DVR_ALARMOUTCFG_V30`），比如报警输出名称、布防时间、输出报警延时等。 相关接口：[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)。

[调用实例代码](#)

3.3 实时预览模块流程

图 3.2 实时预览模块流程

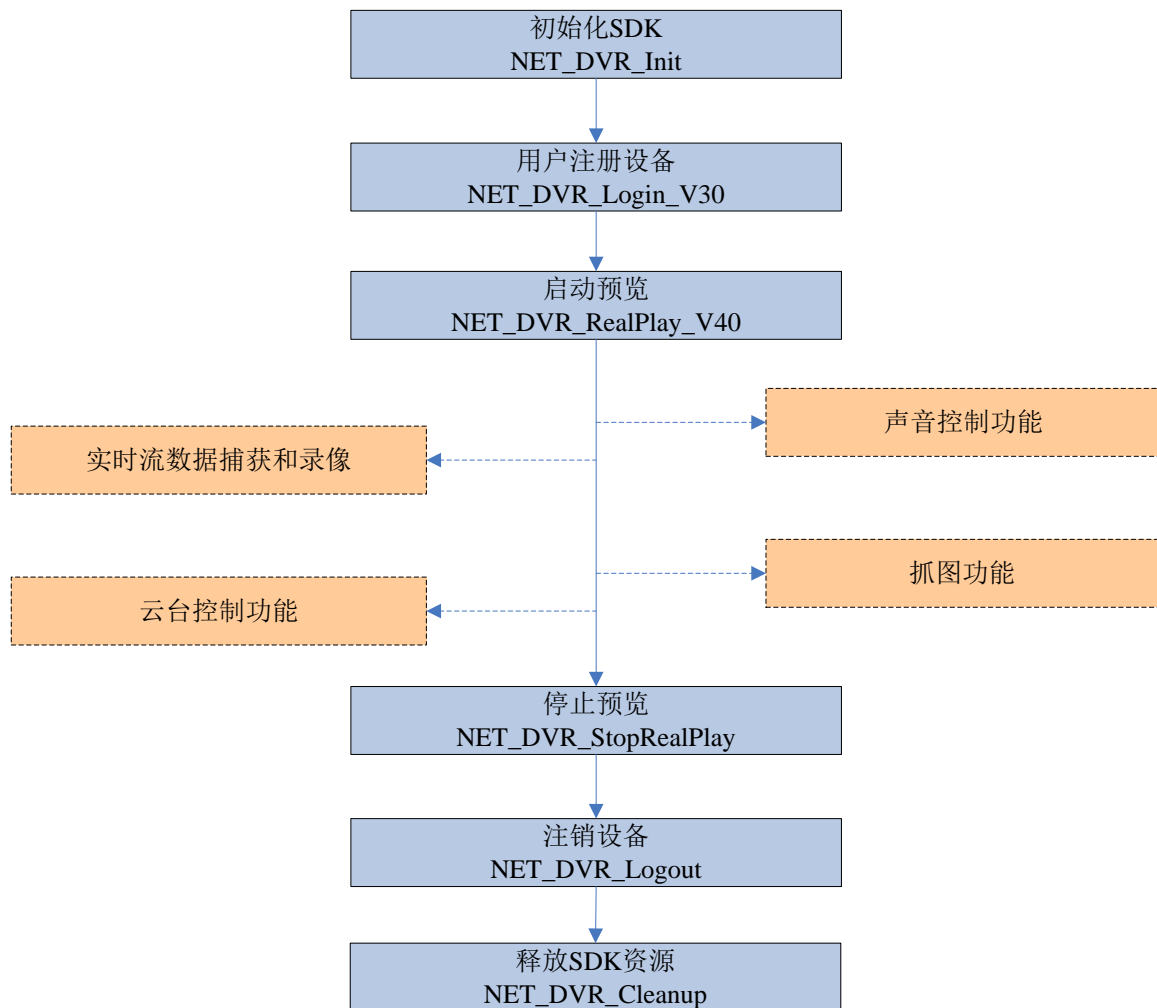


图 3.2 中虚线框部分的模块是与预览模块相关，必须在启动预览后才能调用，这些模块之间是并列的关系，各自完成相应的功能。

- 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。
- 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](#)。

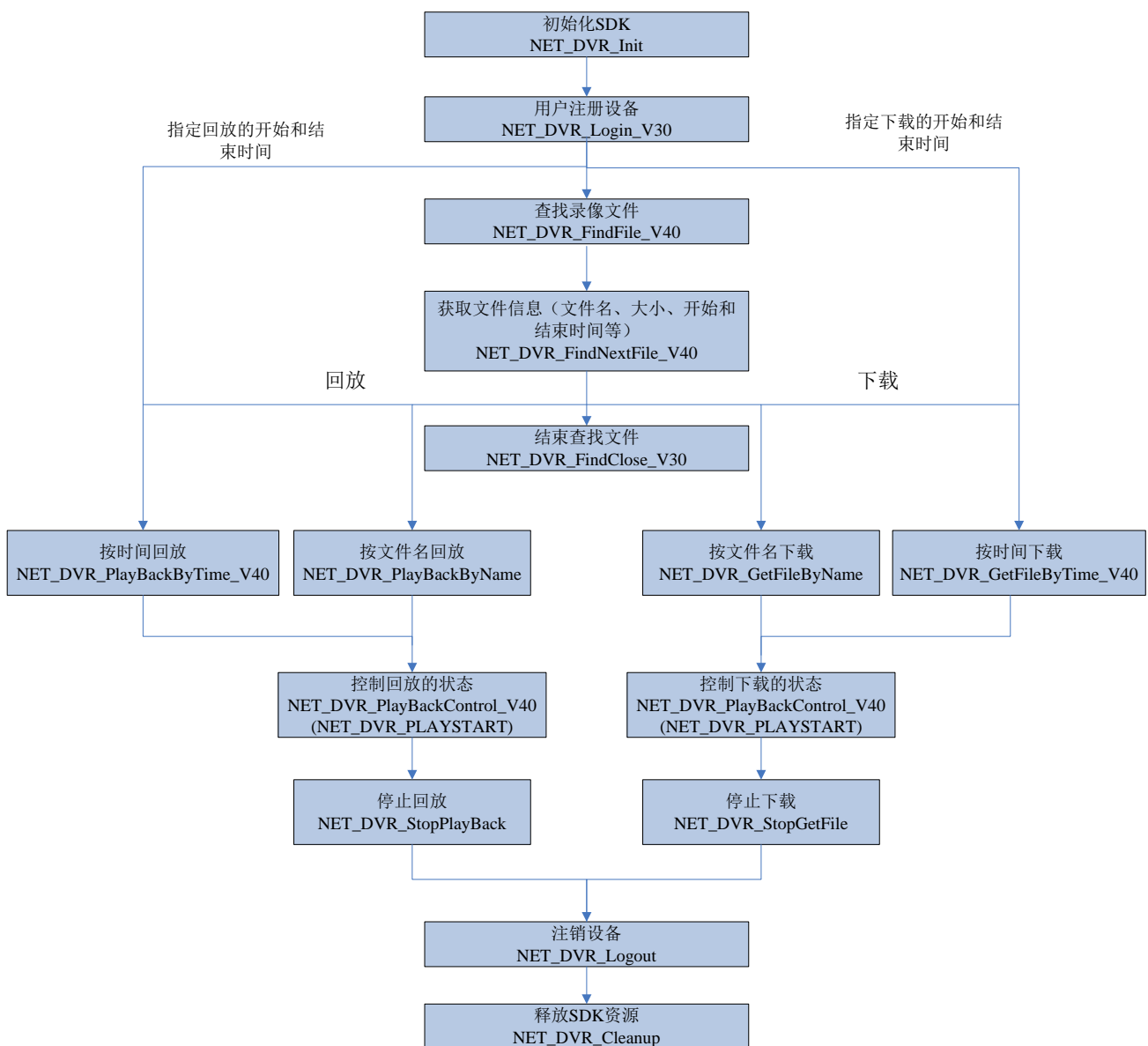
实时流解码方式

- 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。

[调用实例代码](#)

3.4 回放和下载模块流程

图 3.3 回放和下载模块流程



- 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信息（相关接口 [NET_DVR_FindFile_V40](#)、[NET_DVR_FindNextFile_V40](#)），然后根据获取到的文件名开始回放或下载（相关接口

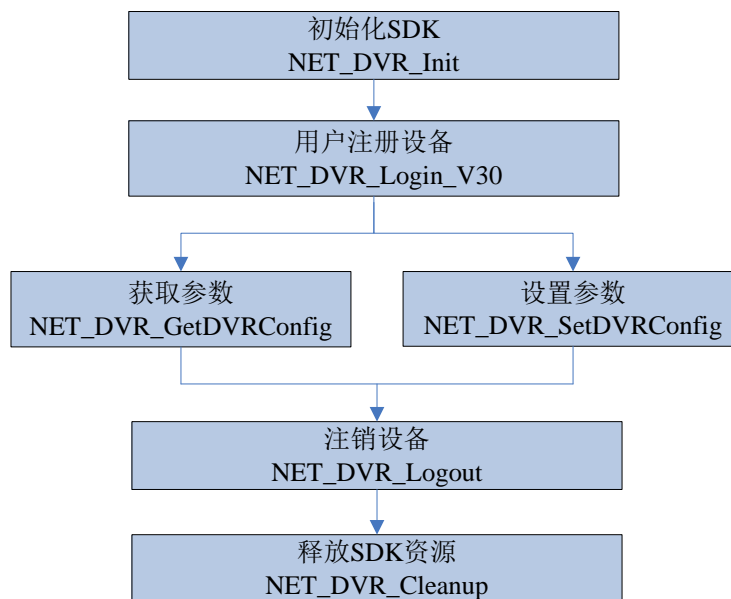
[NET_DVR_PlayBackByName](#)、[NET_DVR_GetFileByName](#)），特别提醒在调用了回放或下载的接口后，还必须调用控制接口（[NET_DVR_PlayBackControl V40](#)）的开始播放命令（[NET_DVR_PLAYSTART](#)）。

- 按时间回放或下载文件时，用户可以无需调用查找录像文件的相关接口，只要在接口中指定开始和结束时间，调用回放或下载接口（相关接口 [NET_DVR_PlayBackByTime_V40](#)、[NET_DVR_GetFileByTime_V40](#)）后，还必须调用控制接口（[NET_DVR_PlayBackControl V40](#)）的开始播放命令（[NET_DVR_PLAYSTART](#)）。此时，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。用户也可以通过调用查找录像文件的相关接口，获取文件的开始和结束时间后，按这个时间范围指定回放或下载接口中的时间参数，最后还必须调用控制接口（[NET_DVR_PlayBackControl V40](#)）的开始播放命令（[NET_DVR_PLAYSTART](#)）。
- 回放或下载文件时，在调用控制接口（[NET_DVR_PlayBackControl V40](#)）的开始播放命令（[NET_DVR_PLAYSTART](#)）之前，可以调用控制接口（[NET_DVR_PlayBackControl V40](#)）的设置转码类型的命令（[NET_DVR_SET_TRANS_TYPE](#)）。这样回放回调函数或下载获取的录像数据为转封装之后的数据，支持的转封装格式有：PS、TS、RTP。

[调用实例代码](#)

3.5 参数配置模块流程

图 3.4 参数配置模块流程

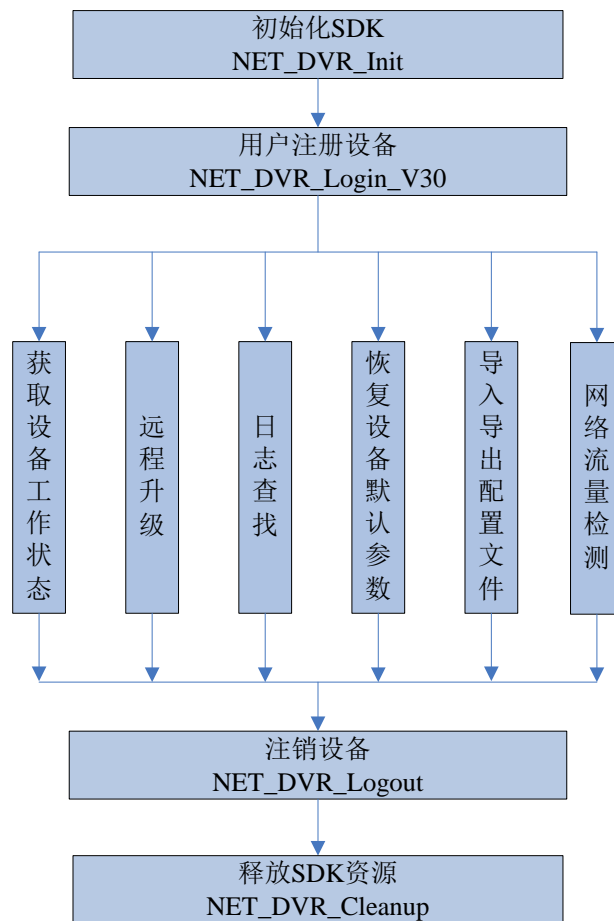


- 实现参数配置首先必须做好初始化 SDK 和用户注册这两个步骤，将用户注册接口返回的 ID 号作为配置接口的首个参数。建议在每次设置某类参数之前，先调用获取参数的接口（[NET_DVR_GetDVRConfig](#)）得到完整的参数结构，修改需要更改的参数，作为设置参数接口中的输入参数，最后调用设置参数接口（[NET_DVR_SetDVRConfig](#)），返回成功即设置成功。

[调用实例代码](#)

3.6 远程设备维护模块流程

图 3.5 设备维护模块流程



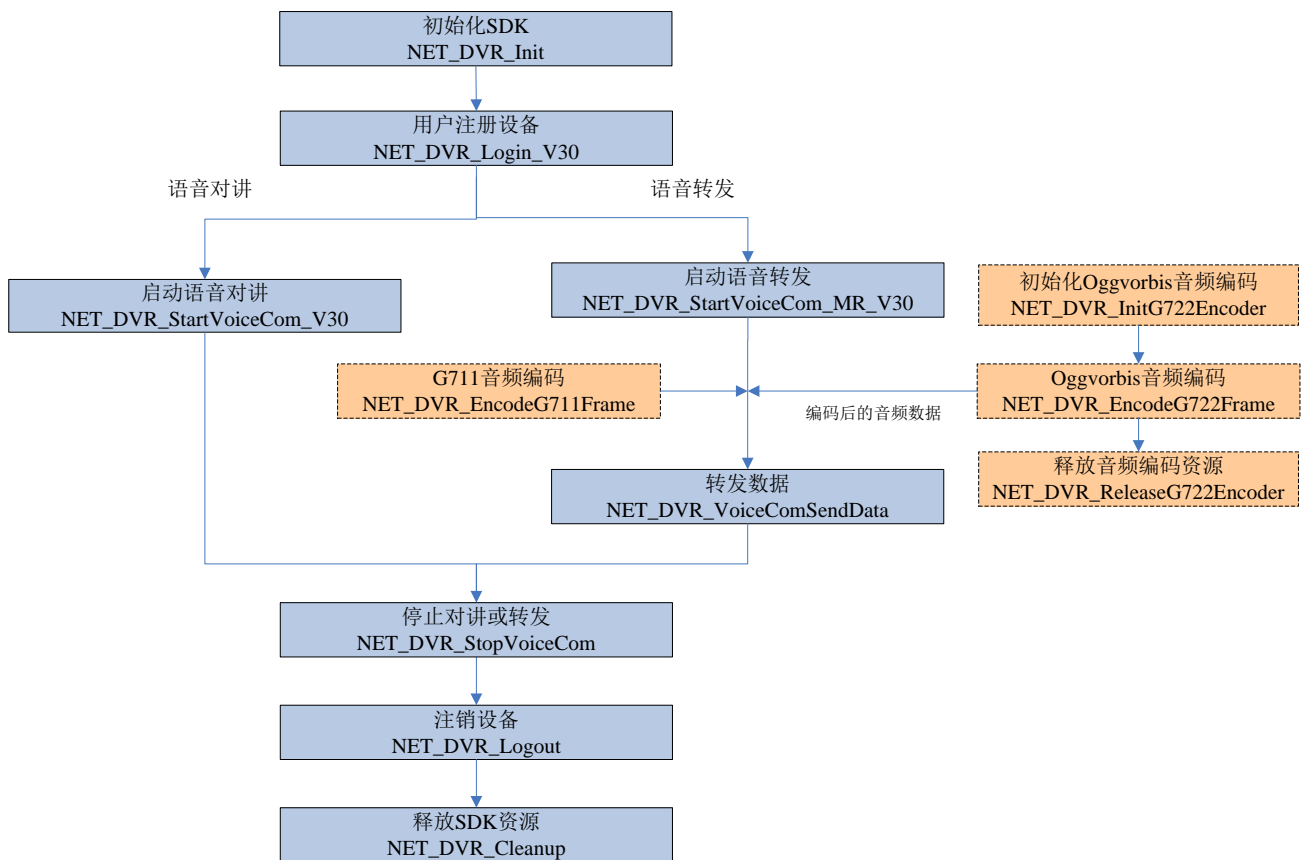
远程设备维护模块包括获取设备工作状态、远程升级、日志查找、恢复设备默认参数和导入、导出配置文件等功能。

- 获取设备工作状态：可以获取到设备当前的硬盘状态、通道状态、报警输入和输出口状态、本地显示状态和语音通道状态等信息。相关接口有：[NET_DVR_GetDVRWorkState_V30](#) 等。
- 远程升级：对设备进行升级，并且可以获取当前升级的进度和状态。相关接口有：[NET_DVR_Upgrade](#)、[NET_DVR_GetUpgradeProgress](#)、[NET_DVR_GetUpgradeState](#) 等。
- 日志查找：可以搜索到当前设备的日志信息，包括报警、异常、操作和带 S.M.A.R.T 信息的日志。相关接口有：[NET_DVR_FindDVRLog_V30](#)、[NET_DVR_FindNextLog_V30](#) 等。
- 恢复设备默认参数：调用接口 [NET_DVR_RestoreConfig](#) 能将设备的所有参数都恢复成默认值。
- 导入、导出配置文件：将设备目前的所有配置信息导出保存或者将指定的配置信息导入到设备。相关接口有：[NET_DVR_GetConfigFile_V30](#)、[NET_DVR_GetConfigFile](#)、[NET_DVR_SetConfigFile_EX](#)、[NET_DVR_SetConfigFile](#) 等。
- 网络流量检测：检测设备当前的发送和接收的网络流量（kbps）。相关接口：[NET_DVR_StartNetworkFlowTest](#)、[NET_DVR_StopNetworkFlowTest](#)

[调用实例代码](#)

3.7 语音对讲转发模块流程

图 3.6 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。

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。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。

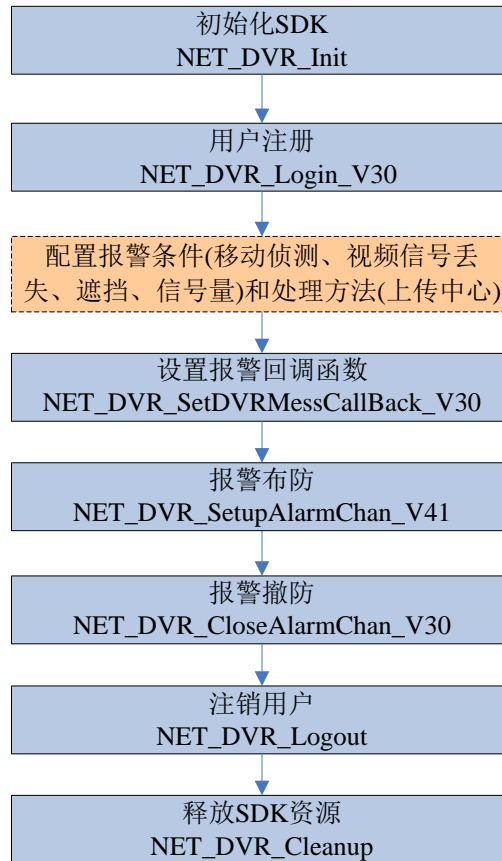
- Windows 64 位或者 Linux 系统下只支持语音转发功能，语音对讲、语音广播、音频编解码接口均不支持

[调用实例代码](#)

3.8 报警模块流程

3.8.1 报警（布防）流程

图 3.7 报警布防流程

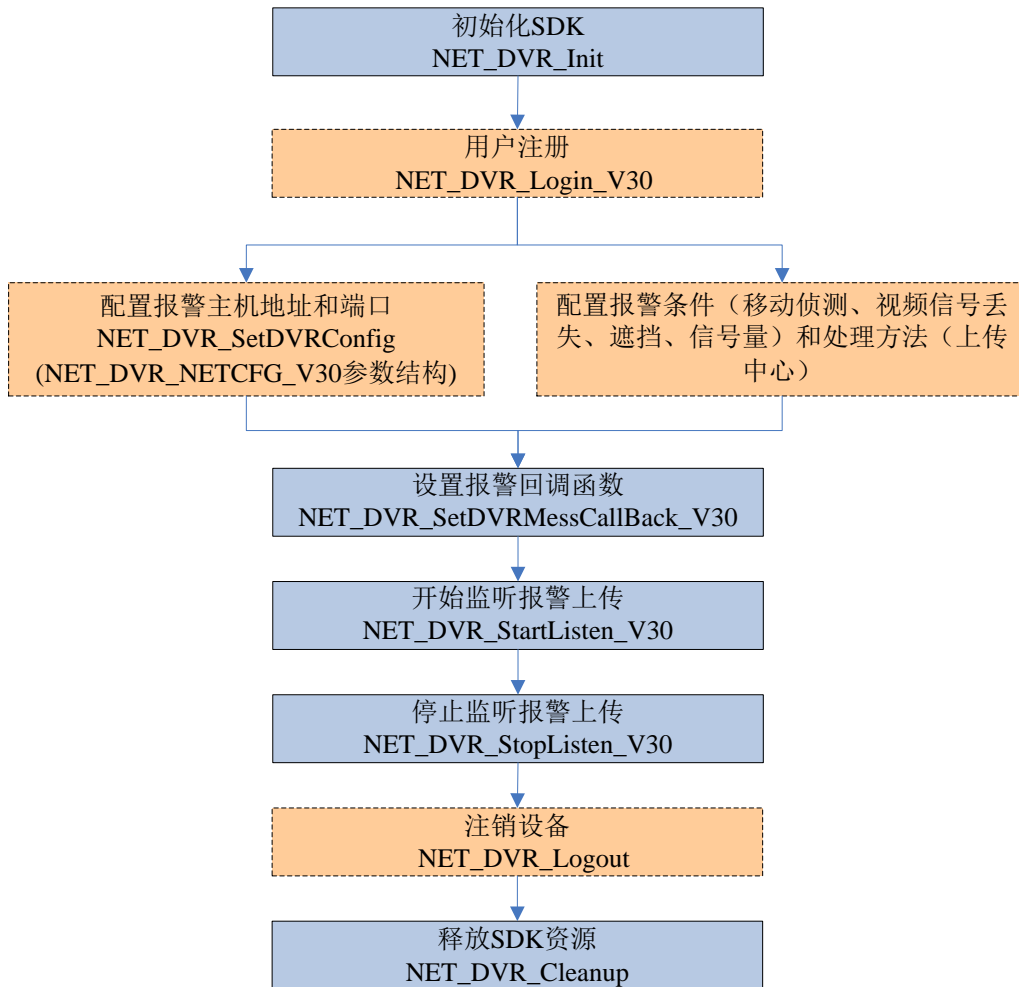


- “布防”报警方式：是指 SDK 主动连接设备，并发起报警上传命令，设备发生报警立即发送给 SDK。
- 由“报警（布防）的流程图”中看出，“布防”方式需要先进行用户注册（[NET_DVR_Login_V30](#)）。虚线框部分是实现报警信息上传的必要条件，主要完成相关的报警条件和处理方法的配置，参数配置的接口为 [NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)。支持的报警类型有移动侦测、视频信号丢失、遮挡和信号量报警，其中前三种报警类型对应的报警条件和处理方法的配置结构体是 [NET_DVR_PICCFG_V40](#)，而信号量报警的配置结构体是 [NET_DVR_ALARMINGCFG_V40](#)。这些参数如果已经配置完成，那么虚线框部分可以省略。接下来就是设置报警回调函数（[NET_DVR_SetDVRMessageCallBack_V30](#) 等），调用成功后还需要设置布防（[NET_DVR_SetupAlarmChan_V41](#)）。整个报警上传过程结束后还需要调用撤防接口等操作。

[调用实例代码](#)

3.8.2 报警（监听）流程

图 3.8 报警监听流程



- “监听”报警方式：是指 SDK 不主动发起连接设备，只是在设定的端口上监听接收设备主动上传的报警信息。
- 这个过程需要远程配置设备的报警主机地址（即 PC 机地址）和报警主机端口（即 PC 的监听端口），报警主机就在该端口上监听接收设备主动上传的报警信息。如果报警主机地址和报警主机端口已配置完成，那么“报警（监听）的流程图”中虚线框“用户注册”和“配置报警主机地址和端口”部分就可以省略，但事先没有配置，就必须调用参数配置接口（[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）对设备的网络参数（NET_DVR_NETCFG_V30）进行配置。而虚线框“配置报警条件和处理方法”部分与“布防”中的一致。对以上需要配置的参数都设置完后，调用 [NET_DVR_StartListen_V30](#) 函数，开启 SDK 的监听端口，准备接收设备上传的报警信息。
- 该方式适用于多个设备向一台客户端上传报警，而且不需要设备登录即可完成，设备重启后不影响报警上传；缺点是设备只支持一个报警主机地址和端口号的配置。

[调用实例代码](#)

3.9 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 IP 通道资源配置示例代码

[相关模块流程](#)

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

void main() {
    int i=0;
    BYTE byIPID,byIPIDHigh;
    int iDevInfoIndex, iGroupNO, iIPCh, iDevID;
    DWORD dwReturned = 0;

    //-----
    // 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("172.6.22.165", 8000, "admin", "12345", &struDeviceInfo);
    if (lUserID < 0)
    {
        printf("Login error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }

    printf("The max number of analog channels: %d\n",struDeviceInfo.byChanNum); //模拟通道个数
    printf("The max number of IP channels: %d\n",struDeviceInfo.byIPChanNum); //IP 通道个数

    //获取 IP 通道参数信息
    NET_DVR_IPPARACFG_V40 IPAccessCfgV40;
    memset(&IPAccessCfgV40, 0, sizeof(NET_DVR_IPPARACFG));
    iGroupNO=0;
```

```

    if (!NET_DVR_GetDVRConfig(IUserID, NET_DVR_GET_IPPARACFG_V40, iGroupNO, &IPAccessCfgV40,
sizeof(NET_DVR_IPPARACFG_V40), &dwReturned))
    {
        printf("NET_DVR_GET_IPPARACFG_V40 error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Logout(IUserID);
        NET_DVR_Cleanup();
        return;
    }
    else
    {
        for (i=0;i<IPAccessCfgV40.dwDChanNum;i++)
        {
            switch (IPAccessCfgV40.struStreamMode[i].byGetStreamType)
            {
                case 0: //直接从设备取流
                    if (IPAccessCfgV40.struStreamMode[i].uGetStream.struChanInfo.byEnable)
                    {
                        byIPID=IPAccessCfgV40.struStreamMode[i].uGetStream.struChanInfo.byIPID;
                        byIPIDHigh=IPAccessCfgV40.struStreamMode[i].uGetStream.struChanInfo.byIPIDHigh;
                        iDevInfoIndex=byIPIDHigh*256 + byIPID-1-iGroupNO*64;
                        printf("IP channel no.%d is online, IP: %s\n", i+1, \
IPAccessCfgV40.struIPDevInfo[iDevInfoIndex].struIP.slIpV4);
                    }
                    break;
                case 1: //从流媒体取流
                    if (IPAccessCfgV40.struStreamMode[i].uGetStream.struPUStream.struStreamMediaSvrCfg.byValid)
                    {
                        printf("IP channel %d connected with the IP device by stream server.\n", i+1);
                        printf("IP of stream server: %s, IP of IP device: %s\n", \
IPAccessCfgV40.struStreamMode[i].uGetStream.struPUStream.struStreamMediaSvrCfg.struDevIP.slIpV4, \
IPAccessCfgV40.struStreamMode[i].uGetStream.struPUStream.struDevChanInfo.struIP.slIpV4);
                    }
                    break;
                default:
                    break;
            }
        }
    }
    //获取 IP 通道相关参数(以获取 IP 通道图像参数为例)
    DWORD iIPChanIndex = IPAccessCfgV40.dwStartDChan; //起始通道，即 IP 通道 01

    NET_DVR_PICCFG_V40 m_struPicCfg; //存放通道图像参数的结构体
    memset(&m_struPicCfg, 0, sizeof(NET_DVR_PICCFG_V40));
    if (!NET_DVR_GetDVRConfig(IUserID, NET_DVR_GET_PICCFG_V40, iIPChanIndex, &m_struPicCfg,

```

```

sizeof(NET_DVR_PICCFG_V40), &dwReturned))
{
    printf("NET_DVR_GET_PICCFG_V40 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}
printf("The channel name of IP channel01: %s\n", m_struPicCfg.sChanName);

//-----
//支持自定义协议
NET_DVR_CUSTOM_PROTOCOL struCustomPro;
//获取自定义协议 1
if (!NET_DVR_GetDVRConfig(IUserID, NET_DVR_GET_CUSTOM_PRO_CFG, 1, &struCustomPro,
sizeof(NET_DVR_CUSTOM_PROTOCOL), &dwReturned))
{
    printf("NET_DVR_GET_CUSTOM_PRO_CFG error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}
struCustomPro.dwEnabled=1;    //启用主码流
struCustomPro.dwEnableSubStream=1; //启用子码流
strcpy((char *)struCustomPro.sProtocolName, "Protocol_RTSP"); //自定义协议名称:Protocol_RTSP,最大 16 字节
struCustomPro.byMainProType=1;    //主码流协议类型: 1- RTSP
struCustomPro.byMainTransType=2; //主码流传输协议: 0- Auto, 1- udp, 2- rtp over rtsp
struCustomPro.wMainPort=554;      //主码流取流端口
strcpy((char *)struCustomPro.sMainPath, "rtsp://192.168.1.65/h264/ch1/main/av_stream"); //主码流取流 URL
struCustomPro.bySubProType=1;     //子码流协议类型: 1- RTSP
struCustomPro.bySubTransType=2;   //子码流传输协议: 0- Auto, 1- udp, 2- rtp over rtsp
struCustomPro.wSubPort=554;       //子码流取流端口
strcpy((char *)struCustomPro.sSubPath, "rtsp://192.168.1.65/h264/ch1/sub/av_stream"); //子码流取流 URL

//设置自定义协议 1
if (!NET_DVR_SetDVRConfig(IUserID, NET_DVR_SET_CUSTOM_PRO_CFG, 1, &struCustomPro,
sizeof(NET_DVR_CUSTOM_PROTOCOL)))
{
    printf("NET_DVR_SET_CUSTOM_PRO_CFG error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}
printf("Set the custom protocol: %s\n", "Protocol_RTSP");
NET_DVR_IPC_PROTO_LIST m_struProtoList;

```

```

if (!NET_DVR_GetIPCProtoList(IUserID, &m_struProtoList)) //获取设备支持的前端协议
{
    printf("NET_DVR_GetIPCProtoList error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//-----
//添加 IP 设备
for(i = 0; i<MAX_IP_DEVICE_V40; i++)
{
    if (IPAccessCfgV40.struIPDevInfo[i].byEnable==0) //选择空闲的设备 ID
    {
        iDevID=i+1;
        break;
    }
}

//添加 IP 通道 5
iIPCh=5;
IPAccessCfgV40.struIPDevInfo[iDevID-1].byEnable=1; //启用
for (i = 0; i<m_struProtoList.dwProtoNum; i++)
{
    if(strcmp((char *)struCustomPro.sProtocalName,(char *)m_struProtoList.struProto[i].byDescribe)==0)
    {
        IPAccessCfgV40.struIPDevInfo[iDevID-1].byProType=m_struProtoList.struProto[i].dwType; //选择自定义协议
        break;
    }
}

//IPAccessCfgV40.struIPDevInfo[iIPCh].byProType=0; //厂家私有协议
strcpy((char *)IPAccessCfgV40.struIPDevInfo[iDevID-1].struIP.slIPV4,"172.6.22.225"); //前端 IP 设备的 IP 地址
IPAccessCfgV40.struIPDevInfo[iDevID-1].wDVRPort=8000; //前端 IP 设备服务端口
strcpy((char *)IPAccessCfgV40.struIPDevInfo[iDevID-1].sUserName,"admin"); //前端 IP 设备登录用户名
strcpy((char *)IPAccessCfgV40.struIPDevInfo[iDevID-1].sPassword,"12345"); //前端 IP 设备登录密码
IPAccessCfgV40.struStreamMode[iIPCh-1].byGetStreamType=0; //直接从设备取流
IPAccessCfgV40.struStreamMode[iIPCh-1].uGetStream.struChanInfo.byChannel=1; //前端 IP 设备的通道号
IPAccessCfgV40.struStreamMode[iIPCh-1].uGetStream.struChanInfo.byIPID=iDevID%256; //通道对应前端 IP 设备 ID 的低 8 位
IPAccessCfgV40.struStreamMode[iIPCh-1].uGetStream.struChanInfo.byIPIDHigh=iDevID/256;
//通道对应前端 IP 设备 ID 的高 8 位

//IP 通道配置，包括添加、删除、修改 IP 通道等
if (!NET_DVR_SetDVRConfig(IUserID, NET_DVR_SET_IPPARACFG_V40, iGroupNO, &IPAccessCfgV40,
sizeof(NET_DVR_IPPARACFG_V40)))
{
    printf("NET_DVR_SET_IPPARACFG_V40 error, %d\n", NET_DVR_GetLastError());
}

```

```
        NET_DVR_Logout(IUserID);
        NET_DVR_Cleanup();
        return;
    }
    else
    {
        printf("Set IP channel no.%d, IP: %s\n", iIPCh, IPAccessCfgV40.strulPDevInfo[iDevID-1].strulP.slPv4);
    }

    //注销用户
    NET_DVR_Logout(IUserID);

    //释放 SDK 资源
    NET_DVR_Cleanup();
    return;
}
```

4.2 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 GetConsoleWindow;

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);

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

    //-----
    // 注册设备
    LONG IUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    IUserID = NET_DVR_Login_V30("192.0.0.64", 8000, "admin", "12345", &struDeviceInfo);
    if (IUserID < 0)
    {
        printf("Login error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }

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

    //-----
    //启动预览
    LONG IRealPlayHandle;
    HWND hWnd = GetConsoleWindow();    //获取窗口句柄
    NET_DVR_PREVIEWINFO struPlayInfo = {0};
    struPlayInfo.hPlayWnd = hWnd;        //需要 SDK 解码时句柄设为有效值，仅取流不解码时可设为空
    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, 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 GetConsoleWindow;

LONG IPort; //全局的播放库 port 号

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

    switch (dwDataType)
    {
        {
            case NET_DVR_SYSHEAD: //系统头

                if (!IPlayM4_GetPort(&IPort)) //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);

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

    //-----
    // 注册设备
    LONG IUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    IUserID = NET_DVR_Login_V30("172.0.0.100", 8000, "admin", "12345", &struDeviceInfo);
    if (IUserID < 0)
    {
        printf("Login error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }

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

    //-----
    //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

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

Sleep(10000);
//-----
//关闭预览
NET_DVR_StopRealPlay(lRealPlayHandle);
//注销用户
NET_DVR_Logout(lUserID);
NET_DVR_Cleanup();

return;
}
```

4.3 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;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    IUserID = NET_DVR_Login_V30("192.0.0.64", 8000, "admin", "12345", &struDeviceInfo);
    if (IUserID < 0)
    {
        printf("Login error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }

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

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

    //-----
    // 注册设备
    LONG IUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    IUserID = NET_DVR_Login_V30("192.0.0.64", 8000, "admin", "12345", &struDeviceInfo);
    if (IUserID < 0)
    {
        printf("Login error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Cleanup();
        return;
    }

    HWND hWnd = GetConsoleWindow();    //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 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;  
    }  
  
    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(lUserID, "./test.mp4",&struDownloadCond);
if(hPlayback < 0)
{
    printf("NET_DVR_GetFileByTime_V40 fail,last error %d\n",NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    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(lUserID);
    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(lUserID);
    NET_DVR_Cleanup();
    return;
}
if(nPos<0 || nPos>100)
{
    printf("download err [%d]\n",NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    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;
}
```

4.4 参数配置模块的示例代码

[相关模块流程图](#)

配置压缩参数（NET_DVR_COMPRESSIONCFG_V30）

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

void main() {

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

//-----
// 注册设备
LONG IUserID;
NET_DVR_DEVICEINFO_V30 struDeviceInfo;
IUserID = NET_DVR_Login_V30("192.0.0.64", 8000, "admin", "12345", &struDeviceInfo);
if (IUserID < 0)
{
    printf("Login error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

int iRet;
//获取压缩参数
```

```
DWORD dwReturnLen;
NET_DVR_COMPRESSIONCFG_V30 struParams = {0};
iRet = NET_DVR_GetDVRConfig(IUserID, NET_DVR_GET_COMPRESSCFG_V30, struDeviceInfo.byStartChan, &struParams, \
sizeof(NET_DVR_COMPRESSIONCFG_V30), &dwReturnLen);
if (!iRet)
{
    printf("NET_DVR_GetDVRConfig NET_DVR_GET_COMPRESSCFG_V30 error.\n");
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//设置压缩参数
struParams.struNormHighRecordPara.dwVideoBitrate = 22;
iRet = NET_DVR_SetDVRConfig(IUserID, NET_DVR_SET_COMPRESSCFG_V30, struDeviceInfo.byStartChan, \
    &struParams, sizeof(NET_DVR_COMPRESSIONCFG_V30));
if (!iRet)
{
    printf("NET_DVR_GetDVRConfig NET_DVR_SET_COMPRESSCFG_V30 error.\n");
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//获取压缩参数
iRet = NET_DVR_GetDVRConfig(IUserID, NET_DVR_GET_COMPRESSCFG_V30, struDeviceInfo.byStartChan, \
    &struParams, sizeof(NET_DVR_COMPRESSIONCFG_V30), &dwReturnLen);
if (!iRet)
{
    printf("NET_DVR_GetDVRConfig NET_DVR_GET_COMPRESSCFG_V30 error.\n");
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

printf("Video Bitrate is %d\n", struParams.struNormHighRecordPara.dwVideoBitrate);
//注销用户
NET_DVR_Logout(IUserID);
//释放 SDK 资源
NET_DVR_Cleanup();
return;
}
```

4.5 远程设备维护模块的示例代码

[相关模块流程图](#)

日志查询

```
#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 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;
    }

    NET_DVR_TIME struStartTime, struStopTime;
    struStartTime.dwYear    = 2011;
    struStartTime.dwMonth   = 3;
    struStartTime.dwDay     = 2;
    struStartTime.dwHour    = 9;
    struStartTime.dwMinute  = 0;
    struStartTime.dwSecond  = 0;

    struStopTime.dwYear     = 2011;
    struStopTime.dwMonth    = 3;
    struStopTime.dwDay      = 2;
```

```

struStopTime.dwHour    = 9;
struStopTime.dwMinute  = 10;
struStopTime.dwSecond  = 0;

//-----
//查找日志
int IFindHandle = NET_DVR_FindDVRLog_V30(IUserID, 0, 0, 0, &struStartTime, &struStopTime, FALSE);
if(IFindHandle < 0)
{
    printf("find log fail,last error %d\n",NET_DVR_GetLastError());
    return;
}
NET_DVR_LOG_V30 struLog;
while(true)
{
    int result = NET_DVR_FindNextLog_V30(IFindHandle, &struLog);
    if(result == NET_DVR_ISFINDING)
    {
        printf("finding\n");
        continue;
    }
    else if(result == NET_DVR_FILE_SUCCESS)
    {
        char strLog[256] = {0};
        printf("log:%04d-%02d-%02d %02d:%02d:%02d\n", struLog.strLogTime.dwYear, struLog.strLogTime.dwMonth,
struLog.strLogTime.dwDay, \
            struLog.strLogTime.dwHour,struLog.strLogTime.dwMinute, struLog.strLogTime.dwSecond);
    }
    else if(result == NET_DVR_FILE_NOFIND || result == NET_DVR_NOMOREFILE)
    {
        printf("find ending\n");
        break;
    }
    else
    {
        printf("find log fail for illegal get file state\n");
        break;
    }
}

//停止日志查询
if(IFindHandle > 0)
{
    NET_DVR_FindLogClose_V30(IFindHandle);
}

```



```
}

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

4.6 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;
    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;
    }

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

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

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

4.7 报警模块的示例代码

[相关模块流程图](#)

布防报警的示例代码:

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

void CALLBACK MessageCallback(LONG lCommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD dwBufLen, void*
```

```
pUser)
{
    int i;
    NET_DVR_ALARMINFO struAlarmInfo;
    memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO));
    switch(ICommand)
    {
    case COMM_ALARM:
        {
            switch (struAlarmInfo.dwAlarmType)
            {
            case 3: //移动侦测报警
                for (i=0; i<16; i++)    //define MAX_CHANNUM    16    //最大通道数
                {
                    if (struAlarmInfo.dwChannel[i] == 1)
                    {
                        printf("发生移动侦测报警的通道号 %d\n", i+1);
                    }
                }
                break;
            default:
                break;
            }
        }
        break;
    default:
        break;
    }
}

void main() {

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

    //-----
    // 注册设备
    LONG IUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    IUserID = NET_DVR_Login_V30("192.0.0.64", 8000, "admin", "12345", &struDeviceInfo);
```

```
if (IUserID < 0)
{
    printf("Login error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

//设置报警回调函数
NET_DVR_SetDVRMessageCallBack_V30(MessageCallback, NULL);

//启用布防
LONG IHandle;
NET_DVR_SETUPALARM_PARAM struAlarmParam={0};
struAlarmParam.dwSize=sizeof(struAlarmParam);
struAlarmParam.byAlarmInfoType=0;
//设备是否支持新的报警信息通过登录返回的 NET_DVR_DEVICEINFO_V30 中参数 bySupport1 & 0x80 判断

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(5000);
//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;
}
```

监听报警的示例代码:

```

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

void CALLBACK MessageCallback(LONG ICommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD dwBufLen, void*
pUser)
{
    int i;
    NET_DVR_ALARMINFO struAlarmInfo;
    memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO));
    switch(ICommand)
    {
    case COMM_ALARM:
        {
            switch (struAlarmInfo.dwAlarmType)
            {
            case 3: //移动侦测报警
                for (i=0; i<16; i++)    //define MAX_CHANNUM    16    //最大通道数
                {
                    if (struAlarmInfo.dwChannel[i] == 1)
                    {
                        printf("发生移动侦测报警的通道号 %d\n", i+1);
                    }
                }
                break;
            default:
                break;
            }
        }
        break;
    default:
        break;
    }
}

void main() {

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

```

```
NET_DVR_SetReconnect(10000, true);

//-----
// 注册设备
LONG IUserID;
NET_DVR_DEVICEINFO_V30 struDeviceInfo;
IUserID = NET_DVR_Login_V30("172.0.0.100", 8000, "admin", "12345", &struDeviceInfo);
if (IUserID < 0)
{
    printf("Login error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

//设置报警回调函数
NET_DVR_SetDVRMessageCallBack_V30(MessageCallback, NULL);

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

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

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

4.8 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 pSDKAbl)

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 lUserID, LONG lHandle, void *pUser)
```

[out]dwType 异常或重连等消息的类型, 详见表 5.3
[out]lUserID 登录 ID
[out]lHandle 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
EXCEPTION_PICPREVIEWRECONNECT	0x8019	图片预览重连
PICPREVIEW_RECONNECTSUCCESS	0x8020	图片预览重连成功
EXCEPTION_PICPREVIEW	0x8021	图片预览异常
EXCEPTION_MAX_ALARM_INFO	0x8022	报警信息缓存已达上限
EXCEPTION_LOST_ALARM	0x8023	报警丢失
EXCEPTION_RELOGIN	0x8040	用户重登陆
RELOGIN_SUCCESS	0x8041	用户重登陆成功

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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 IP Server 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: 建议使用此接口实现注销功能。

[返回目录](#)

5.4 获取设备能力集

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

函 数： BOOL NET_DVR_GetDeviceAbility(LONG 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 能力集类型，具体定义见表 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，可以是 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，可以是 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_SOFTHARDWARE_ABILITY	0x001	设备软硬件能力
DEVICE_NETWORK_ABILITY	0x002	设备网络能力
DEVICE_ENCODE_ALL_ABILITY_V20	0x008	设备所有编码能力
DEVICE_RAID_ABILITY	0x007	设备 RAID 能力
DEVICE_ALARM_ABILITY	0x00a	设备报警能力
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
PIC_CAPTURE_ABILITY	0x402	抓图图片分辨率能力集

返回值： TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [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

dwAbilityType 宏定义	pInBuf	pOutBuf
DEVICE_SOFTHARDWARE_ABILITY	无	设备软硬件能力 XML 描述
DEVICE_NETWORK_ABILITY	无	设备网络能力 XML 描述
DEVICE_ENCODE_ALL_ABILITY_V20	编码能力获取输入描述	设备所有编码能力 XML 描述
DEVICE_RAID_ABILITY	无	设备 RAID 能力 XML 描述

DEVICE_ALARM_ABILITY	报警能力获取输入描述	设备报警能力 XML 描述
DEVICE_USER_ABILITY	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	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	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	JPEG 抓图能力获取输入描述	设备 JPEG 抓图能力 XML 描述
DEVICE_SERIAL_ABILITY	串口能力获取输入描述	设备串口能力 XML 描述
DEVICE_ABILITY_INFO	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 ROI 能力集	ROI 能力 XML 描述(ROIAbility)
	获取录像相关能力集	录像相关能力 XML 描述(RecordAbility)
	NVR 前端待接入设备通道能力集	NVR 前端待接入设备通道能力 XML 描述 (GetAccessDeviceChannelAbility)
	获取设备本地预览切换能力集	设备本地预览切换能力 XML 描述(PreviewSwitchAbility)
	获取设备 N+1 能力集	设备 N+1 能力 XML 描述(NPlusOneAbility)
	获取设备磁盘相关能力集	设备磁盘相关能力 XML 描述(HardDiskAbility)
	获取 IPC 配置文件导入导出能力集	IPC 配置文件导入导出能力 XML 描述 (IPAccessConfigFileAbility)
	获取设备通道输入能力集	设备通道输入能力 XML 描述(ChannellInputAbility)
	获取报警触发录像能力集	设备报警触发录像能力 XML 描述 (AlarmTriggerRecordAbility)
	获取 GB/T28181 能力集	设备 GB/T28181 能力 XML 描述(GBT28181AccessAbility)
PIC_CAPTURE_ABILITY	通道号（4 个字节）	NET_DVR_COMPRESSIONCFG_ABILITY

注：能力集结构和 XML 描述请参见《设备网络 SDK 使用手册.chm》

[返回目录](#)

5.4.2 获取设备能力集(标准协议) NET_DVR_GetSTDAbility

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

参 数： [in]lUserID 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）

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

说 明：

表 5.6 获取设备能力集

宏定义	宏定义值	含义	IpCondBuffer	IpOutBuffer
NET_DVR_GET_ACTIVATE_IPC_ABILITY	13003	获取 NVR 激活前端设备能力	NULL	VideoSourceActivationCapability

		集		
NET_DVR_GET_EVENT_TRIGGERS_CAPABILITIES	3501	获取事件触发能力集	NULL	EventTriggerCap
NET_DVR_GET_GBT28181_SERVICE_CAPABILITIES	6505	获取 GB28181 服务器能力集	NULL	GB28181ServiceCap
NET_DVR_GET_TRAFFIC_CAP	6630	获取抓拍相关能力集	NULL	TrafficCap
NET_DVR_GET_ONLINEUPGRADE_ABILITY	9309	获取在线升级能力集	NULL	OnlineUpgradeCap
NET_DVR_GET_DDNS_COUNTRY_ABILITY	3800	获取设备支持的 DDNS 国家能力列表	NULL	DDNSCountry

[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，包括通道号、码流类型、取流协议
[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

5.6.1 强制 I 帧 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 5.8

[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 lUserID, LONG lChannel,DWORD *pBrightValue, DWORD *pContrastValue,DWORD *pSaturationValue,DWORD *pHueValue)

参 数： [in]lRealHandle NET_DVR_Login_V40 的返回值
[in]lChannel 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 lRealHandle,DWORD pBrightValue, DWORD pContrastValue,DWORD pSaturationValue,DWORD pHueValue)

参 数： [in]lRealHandle 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 IRealHandle, fDrawFun cbDrawFun, DWORD dwUser)

参 数： [in]IRealHandle 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 IRealHandle, HDC hDc, DWORD dwUser)
```

[out]IRealHandle 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。如果调用接口 NET_DVR_RealPlay_V40 进行预览，参数 bBlocked 必须置 1（TRUE），否则该接口调用会失败，获取错误号为 12（调用次序错误）。

[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 IRealHandle)

参 数: [in]IRealHandle NET_DVR_RealPlay_V40 的返回值

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

说 明:

[返回目录](#)

5.9.6 调节播放音量 **NET_DVR_Volume**

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

参 数: [in]IRealHandle NET_DVR_RealPlay_V40 的返回值

[in]wVolume 音量, 取值范围[0,0xffff]

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

说 明:

[返回目录](#)

5.10 实时预览数据捕获

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 IRealHandle, fRealDataCallBack
cbRealDataCallBack,DWORD dwUser)

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

[in]cbRealDataCallBack 码流数据回调函数

[in]dwUser 用户数据

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

[out]IRealHandle 当前的预览句柄

[out]dwDataType 数据类型, 详见表 5.9

[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]lRealHandle` 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
<code>NET_DVR_SYSHEAD</code>	1	系统头数据
<code>NET_DVR_STD_VIDEODATA</code>	4	标准视频流数据
<code>NET_DVR_STD_AUDIODATA</code>	5	标准音频流数据
<code>NET_DVR_PRIVATE_DATA</code>	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 协议取流的设备的标准码流回调。**

[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

返回值: `TRUE` 表示成功, `FALSE` 表示失败。接口返回失败请调用 [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 IRealHandle, char *sPicFileName)`

参 数: `[in] IRealHandle` `NET_DVR_RealPlay_V40` 的返回值
`[in] sPicFileName` 保存图象的文件路径。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，实际是播放库解码抓图，要求在调用 NET_DVR_RealPlay_V40 等接口时传入非空的播放句柄（播放库解码显示），否则时接口会返回失败（调用次序错误）。

[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 IUserID, LONG IChannel, LPNET_DVR_JPEGPARA lpJpegPara, char *sPicFileName)

参 数：

[in]IUserID	NET_DVR_Login_V40 的返回值
[in]IChannel	通道号
[in]lpJpegPara	JPEG 图像参数
[in]sPicFileName	保存 JPEG 图的文件路径

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

说 明： 该接口用于设备的单帧数据捕获：SDK 发送命令给设备，设备抓图之后返回客户端，然后 SDK 将接收到的 JPEG 图片数据保存成文件。JPEG 抓图功能或者抓图分辨率需要设备支持，如果不支持接口返回失败，错误号 23 或者 29。

[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 IUserID, LONG IChannel, LPNET_DVR_JPEGPARA lpJpegPara, char *sJpegPicBuffer, DWORD dwPicSize, LPDWORD lpSizeReturned)

参 数：

[in]IUserID	NET_DVR_Login_V40 的返回值
[in]IChannel	通道号
[in]lpJpegPara	JPEG 图像参数
[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：SDK 发送命令给设备，设备抓图之后返回客户端，然后 SDK 将接收到的 JPEG 图片数据保存在 `sJpegPicBuffer` 缓冲区里。JPEG 抓图功能或者抓图分辨率需要设备支持，如果不支持接口返回失败，错误号 23 或者 29。

[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

wPTZCommand 宏定义	宏定义值	含义
LIGHT_PWRON	2	接通灯光电源
WIPER_PWRON	3	接通雨刷开关
FAN_PWRON	4	接通风扇开关
HEATER_PWRON	5	接通加热器开关
AUX_PWRON1	6	接通辅助设备开关
AUX_PWRON2	7	接通辅助设备开关
ZOOM_IN	11	焦距变大(倍率变大)
ZOOM_OUT	12	焦距变小(倍率变小)
FOCUS_NEAR	13	焦点前调
FOCUS_FAR	14	焦点后调
IRIS_OPEN	15	光圈扩大
IRIS_CLOSE	16	光圈缩小

TILT_UP	21	云台上仰
TILT_DOWN	22	云台下俯
PAN_LEFT	23	云台左转
PAN_RIGHT	24	云台右转
UP_LEFT	25	云台上仰和左转
UP_RIGHT	26	云台上仰和右转
DOWN_LEFT	27	云台下俯和左转
DOWN_RIGHT	28	云台下俯和右转
PAN_AUTO	29	云台左右自动扫描

[返回目录](#)

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。

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。

[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 lUserID, LONG lChannel, DWORD dwPTZCommand, DWORD dwStop, DWORD dwSpeed)

参数: [in]lUserID NET_DVR_Login_V40 的返回值
[in]lChannel 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 lRealHandle, DWORD dwPTZPresetCmd, DWORD dwPresetIndex)

参数: [in]lRealHandle 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]IChannel 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_PTZCruise**

- 函 数： BOOL NET_DVR_PTZCruise(LONG IRealHandle, DWORD dwPTZCruiseCmd, BYTE byCruiseRoute, BYTE byCruisePoint, WORD wInput)
- 参 数： [in]IRealHandle NET_DVR_RealPlay_V40 的返回值
 [in]dwPTZCruiseCmd 操作云台巡航命令，详见表 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）
 巡航点，最多支持 32 个点（序号从 1 开始）
[in]byCruisePoint 不同巡航命令时的值不同，预置点(最大 255)、时间(最大 255)、
[in]wInput 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](#)

云台轨迹操作

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

函数： BOOL NET_DVR_PTZTrack(LONG IRealHandle, DWORD dwPTZTrackCmd)

参数： [in]IRealHandle NET_DVR_RealPlay_V40 的返回值
[in]dwPTZTrackCmd 操作云台巡航命令，详见表 5.14

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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 5.14 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](#)

5.13.10 云台轨迹操作 **NET_DVR_PTZTrack_Other**

函数: BOOL NET_DVR_PTZTrack_Other(LONG IUserID, LONG IChannel, DWORD dwPTZTrackCmd)

参数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IChannel 通道号
[in]dwPTZTrackCmd 操作云台巡航命令, 详见表 5.14

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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

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

函数: BOOL NET_DVR_TransPTZ(LONG IRealHandle, char *pPTZCodeBuf, DWORD dwBufSize)

参数: [in]IRealHandle 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](#)

5.13.12 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

5.13.13 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` 云台图象区域位置信息
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

5.13.14 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
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

5.14 录像文件回放、下载、锁定及备份

获取通道录像起止时间

5.14.1 获取通道录像起止时间 **NET_DVR_InquiryRecordTimeSpan**

函 数: `BOOL NET_DVR_InquiryRecordTimeSpan(LONG IUserID, DWORD dwChannel, LPNET_DVR_RECORD_TIME_SPAN_INQUIRY lpInquiry, LPNET_DVR_RECORD_TIME_SPAN lpResult)`

参 数:

[in]IUserID	NET_DVR_Login_V40 的返回值
[in]dwChannel	通道号
[in]lpInquiry	通道录像起止时间查询条件
[in]lpResult	通道录像起止时间查询结果

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

即时刷新录像索引

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

函 数: `BOOL NET_DVR_UpdateRecordIndex(LONG IUserID, DWORD dwChannel)`

参 数:

[in]IUserID	NET_DVR_Login_V40 的返回值
[in]dwChannel	通道号

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

说 明: 录像索引刷新后, 即可回放刷新前的录像文件。

[返回目录](#)

月历录像查询

5.14.3 获取月历录像分布 **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 5.15
 - [in] dwCount 一次要获取配置参数的个数, 0 和 1 都表示 1 个信息, 2 表示 2 个信息, 最大 64 个
 - [in] lpInBuffer 配置条件缓冲区, 详见表 5.16
 - [in] dwInBufferSize 缓冲区长度
 - [out] lpStatusList 错误信息列表, 和要查询的配置一一对应, 例如 lpStatusList[2]就对应 lpInBuffer[2], 由用户分配内存, 每个错误信息为 4 个字节, 参数值: 0 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 5.16), 和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 0 值, 对应 lpOutBuffer 的内容就是无效的
 - [in] dwOutBufferSize 输出缓冲区大小
- 返回值:** TRUE 表示成功, 但不代表每一个配置都成功, 哪一个成功, 对应查看 lpStatusList[n]值; FALSE 表示全部失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
- 说 明:** 该接口是带有发送数据的批量获取配置信息的通用接口。lpInBuffer 指定需要获取的信息, lpOutBuffer 保存获取得到的 dwCount 个配置信息。不同的 dwCommand 对应不同的结构体和命令号, 如表 5.16 所示。

表 5.15 月历录像查询命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_MONTHLY_RECORD_DISTRIBUTION	获取月历录像分布, dwCount 为 0	6164

表 5.16 月历录像查询

dwCommand	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_GET_MONTHLY_RECORD_DISTRIBUTION	NET_DVR_MRD_SEARCH_PARAM	NET_DVR_MRD_SEARCH_RESULT

[返回目录](#)

录像文件的查找

5.14.4 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 文件查找条件, 包括设备通道号、文件类型、查找起止时间等
- 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](#)

5.14.5 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 保存文件信息的指针, 包括查找到的文件名、文件起止时间、文件类型等

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

表 5.17 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](#)

5.14.6 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](#)

按事件查找录像文件

5.14.7 根据事件查找录像文件 **NET_DVR_FindFileByEvent_V40**

函 数: LONG NET_DVR_FindFileByEvent_V40(LONG IUserID, LPNET_DVR_SEARCH_EVENT_PARAM_V40 lpSearchEventParam)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]lpSearchEventParam 待查找的文件信息结构, 包括事件类型、查找时间、通道等

返回值: -1 表示失败, 其他值作为 NET_DVR_FindNextEvent 等函数的参数。获取错误码调用 NET_DVR_GetLastError。接口返回失败请调用 [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` 接口来获取文件信息。按事件查找出的录像文件针对开始时间和停止时间，因此只支持按时间回放 `NET_DVR_PlayBackByTime`。
该功能需要设备的支持，若设备不支持则接口返回失败，错误号为 23。

[返回目录](#)

5.14.8 逐个获取查找到事件录像信息 `NET_DVR_FindNextEvent_V40`

函数： `LONG NET_DVR_FindNextEvent_V40(LONG lFindHandle, LPNET_DVR_SEARCH_EVENT_RET_V40 lpSearchEventRet)`
参数： `[in]lSearchHandle` `NET_DVR_Login_V40` 的返回值
`[out]lpSearchEventRet` 查找到的事件录像结果信息，包括事件类型、录像时间等
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 5.18。

表 5.18 查找状态信息

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

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

说明： 在调用该接口获取查找文件之前，必须先调用 `NET_DVR_FindFileByEvent` 得到当前的查找句柄。按事件查找出的录像文件针对开始时间和停止时间，因此只支持按时间回放 `NET_DVR_PlayBackByTime()`。

[返回目录](#)

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

函数： `BOOL NET_DVR_FindClose_V30(LONG lFindHandle)`
参数： `[in]lFindHandle` 文件查找句柄，`NET_DVR_FindFileByEvent` 的返回值
返回值： `TRUE` 表示成功，`FALSE` 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说明：

[返回目录](#)

区域移动侦测智能搜索

5.14.10 开始智能搜索 `NET_DVR_SmartSearch_V40`

函数： `LONG NET_DVR_SmartSearch_V40(LONG lUserID, LPNET_DVR_SMART_SEARCH_PARAM_V40`

IpSmartSearchParam)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值

[in]IpSmartSearchParam 智能搜索参数结构

返回值: -1 表示失败, 其他值作为 NET_DVR_SearchNextInfo、NET_DVR_StopSearch 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: 该接口指定了要搜索条件信息, 调用成功后, 就可以调用 [NET_DVR_SearchNextInfo](#) 接口来获取搜索结果。搜索出的录像文件针对开始时间和停止时间, 因此只支持按时间回放 [NET_DVR_PlayBackByTime_V40](#)。该功能需要设备的支持, 若设备不支持则接口返回失败, 错误号为 23。

[返回目录](#)

5.14.11 逐个获取查找到的智能录像信息 NET_DVR_SearchNextInfo

函 数: LONG NET_DVR_SearchNextInfo(LONG ISearchHandle, LPNET_DVR_SMART_SEARCH_RET IpSmartSearchRet)

参 数: [in]ISearchHandle NET_DVR_SmartSearch_V40 的返回值

[out]IpSmartSearchRet 保存搜索结果信息的指针

返回值: -1 表示失败, 其他值表示当前的获取状态等信息, 详见表 5.19。

表 5.19 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_NOFOUND	1001	未查找到文件
NET_DVR_ISFINDING	1002	正在查找请等待
NET_DVR_NOMOREFILE	1003	没有更多的文件, 查找结束
NET_DVR_FILE_EXCEPTION	1004	查找文件时异常

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

说 明: 在调用该接口前, 必须先调用 [NET_DVR_SmartSearch_V40](#) 得到当前的搜索句柄。按搜索出的录像文件针对开始时间和停止时间, 因此只支持按时间回放 [NET_DVR_PlayBackByTime_V40](#)。

[返回目录](#)

5.14.12 停止智能搜索 NET_DVR_StopSearch

函 数: BOOL NET_DVR_StopSearch(LONG ISearchHandle)

参 数: [in]ISearchHandle NET_DVR_SmartSearch_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（正放或倒放）录像文件

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

函数: `LONG NET_DVR_PlayBackByName(LONG lUserID, char *sPlayBackFileName, HWND hWnd)`
 参数: `[in] lUserID` `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:

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](#)

5.14.14 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

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。

Linux 下：

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](#)

5.14.15 按文件名倒放录像文件 NET_DVR_PlayBackReverseByName

函 数： LONG NET_DVR_PlayBackReverseByName(LONG lUserID, char *sPlayBackFileName, HWND hWnd)

参 数： [in]lUserID 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：

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();
```

[返回目录](#)

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

函 数： LONG NET_DVR_PlayBackReverseByTime_V40(LONG IUserID,HWND
hWnd,LPNET_DVR_PLAYCOND pPlayCond)

参 数： [in]IUserID NET_DVR_Login_V40 的返回值
[in]hWnd 回放文件的窗口句柄
[in]pPlayCond 回放条件参数

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。

Linux 下：

HWND 表示播放窗口的句柄，定义为：

```
typedef unsigned int HWND;
```

如果使用 Qt 进行界面开发，示例如下：

```
NET_DVR_CLIENTINFO tmpclientinfo;
```

```
QWidget m_framePlayWnd;
```

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

[返回目录](#)

5.14.17 控制录像回放的状态 **NET_DVR_PlayBackControl_V40**

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

参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName、
NET_DVR_PlayBackReverseByName 或
NET_DVR_PlayBackByTime_V40、
NET_DVR_PlayBackReverseByTime_V40 的返回值
[in]dwControlCode 控制录像回放状态命令，详见表 5.20
[in]lpInBuffer 指向输入参数的指针，详见表 5.21
[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 5.21

[out]lpOutLen

输出参数的长度

表 5.20 回放控制命令

dwControlCode 宏定义	宏定义值	含义
NET_DVR_PLAYSTART	1	开始播放
NET_DVR_PLAYPAUSE	3	暂停播放
NET_DVR_PLAYRESTART	4	恢复播放
NET_DVR_PLAYFAST	5	快放
NET_DVR_PLAYSLOW	6	慢放
NET_DVR_PLAYNORMAL	7	正常速度播放（在暂停后调用将恢复暂停前的速度播放）
NET_DVR_PLAYFRAME	8	单帧放（恢复正常回放使用 NET_DVR_PLAYNORMAL 命令）
NET_DVR_PLAYSTARTAUDIO	9	打开声音
NET_DVR_PLAYSTOPAUDIO	10	关闭声音
NET_DVR_PLAYAUDIOVOLUME	11	调节音量，取值范围[0,0xffff]
NET_DVR_PLAYSETPOS	12	改变文件回放的进度
NET_DVR_PLAYGETPOS	13	获取按文件或者按时间回放的进度
NET_DVR_PLAYGETTIME	14	获取当前已经播放的时间(按文件回放的时候有效)
NET_DVR_PLAYGETFRAME	15	获取当前已经播放的帧数(按文件回放的时候有效)
NET_DVR_GETTOTALFRAMES	16	获取当前播放文件总的帧数(按文件回放的时候有效)
NET_DVR_GETTOTALTIME	17	获取当前播放文件总的时间(按文件回放的时候有效)
NET_DVR_THROWBFRAME	20	丢 B 帧
NET_DVR_SETSPEED	24	设置码流速度
NET_DVR_KEEPAIVE	25	保持与设备的心跳(如果回调阻塞，建议 2 秒发送一次)
NET_DVR_PLAYSETTIME	26	按绝对时间定位
NET_DVR_PLAYGETTOTALLEN	27	获取按时间回放对应时间段内的所有文件的总长度
NET_DVR_PLAY_FORWARD	29	倒放切换为正放
NET_DVR_PLAY_REVERSE	30	正放切换为倒放
NET_DVR_SET_TRANS_TYPE	32	设置转封装类型
NET_DVR_PLAY_CONVERT	33	回放转码

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： 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 5.21 所示。

- 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、NET_DVR_GETTOTALTIME、NET_DVR_PLAYSETTIME 和 NET_DVR_PLAYGETTOTALLEN 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。倒放的进度为 100~0。
- 按时间回放或下载时，只能获取进度值：0、100（结束）、200（异常）。
- NET_DVR_SETSPEED 设置码流速率、NET_DVR_PLAYSETTIME 按绝对时间定位等命令需要设备支持，如果接口返回失败获取错误号为 23 则说明设备不支持该命令。
- NET_DVR_SET_TRANS_TYPE 设置转封装类型和 NET_DVR_PLAY_CONVERT 回放转码需要在开始播放（即 NET_DVR_PLAYSTART）之前调用。

表 5.21 回放控制参数

dwControlCode 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_THROWBFRAME	丢 B 帧	一个 4 字节整型的 B 帧个数	无
NET_DVR_SETSPEED	设置码流速度	一个 4 字节整型的速度值	无
NET_DVR_PLAYSETTIME	按绝对时间定位回放	NET_DVR_TIME	无
NET_DVR_PLAYGETTOTALLEN	获取按时间回放对应时间段内的所有文件的总长度	无	一个 8 字节整型的长度值（单位字节）
NET_DVR_PLAY_FORWARD	倒放切换为正放	应用层解码时需在 IpInBuffer 输入 NET_DVR_TIME 表示当前的播放时间,SDK 解码可将 IpInBuffer 置为 NULL	无
NET_DVR_PLAY_REVERSE	正放切换为倒放	应用层解码时需在 IpInBuffer 输入 NET_DVR_TIME 表示当前的播放时间,SDK 解码可将 IpInBuffer 置为 NULL	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
NET_DVR_PLAY_CONVERT	回放转码	NET_DVR_COMPRESSION_INFO_V30	无

[返回目录](#)

5.14.18 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、
NET_DVR_PlayBackReverseByName 或
NET_DVR_PlayBackByTime_V40、
NET_DVR_PlayBackReverseByTime_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

5.14.19 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 (包括文件名的绝对路径)

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](#)

5.14.20 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](#)

5.14.21 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_PlayBackByTime_V40 或 NET_DVR_PlayBackReverseByTime_V40 的返回值
[in]cbPlayDataCallBack 录像数据回调函数
[in]dwUser 用户数据

```
typedef void(CALLBACK *fPlayDataCallBack)(LONG IPlayHandle, DWORD dwDataType, BYTE *pBuffer,
DWORD dwBufSize, DWORD dwUser)
```

[out]IPlayHandle 当前的录像播放句柄
[out]dwDataType 数据类型, 详见表 5.22
[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 5.22 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)
NET_DVR_CHANGE_FORWARD	10	码流改变为正放
NET_DVR_CHANGE_REVERSE	11	码流改变为倒放

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

说 明:

- 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](#)

录像标签的添加和删除

5.14.22 添加录像标签 NET_DVR_InsertRecordLabel

函 数: BOOL NET_DVR_InsertRecordLabel(LONG IPlayHandle, NET_DVR_RECORD_LABEL* lpRecordLabel, NET_DVR_LABEL_IDENTITY *lpLabelIdentify)

参 数: [in]IPlayHandle NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime_V40 的返回值
[in]lpRecordLabel 录像标签
[out]lpLabelIdentify 添加录像标签后的标识

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

说 明: 录像标签功能可帮助用户在回放录像时记录下某一时间点的相关人员或现场等信息, 以便后续随时根据标签信息, 进行搜索和定位录像资料。

[返回目录](#)

5.14.23 修改录像标签 **NET_DVR_ModifyRecordLabel**

函 数: BOOL NET_DVR_ModifyRecordLabel(LONG IUserID, NET_DVR_MOD_LABEL_PARAM * IpModLabelParam)

参 数: [in]IUserID NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime_V40 的返回值
[in]IpModLabelParam 标签修改结构体

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

5.14.24 删除录像标签 **NET_DVR_DelRecordLabel**

函 数: BOOL NET_DVR_DelRecordLabel(LONG IUserID, NET_DVR_DEL_LABEL_PARAM* IpDelLabelParam)

参 数: [in]IUserID NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime_V40 的返回值
[in]IpDelLabelParam 要删除的标签信息

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

录像标签的查找

5.14.25 搜索录像标签 **NET_DVR_FindRecordLabel**

函 数: LONG NET_DVR_FindRecordLabel(LONG IUserID, LPNET_DVR_FIND_LABEL IpFindLabel)

参 数: [in]IUserID NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime_V40 的返回值
[in]IpFindLabel 欲查找的标签信息结构

返回值: -1 表示失败, 其他值作为 NET_DVR_FindNextLabel 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: 回放时, 可根据查找到的标签 (时间) 信息通过 NET_DVR_PlayBackControl_V40 实现定位操作。每次最多搜索 4000 条标签条。

[返回目录](#)

5.14.26 逐个获取搜索到的录像标签 **NET_DVR_FindNextLabel**

函 数: LONG NET_DVR_FindNextLabel(LONG IFindHandle, LPNET_DVR_FINDLABEL_DATA IpFindData)

参 数: [in]IFindHandle NET_DVR_FindRecordLabel 的返回值
[in]IpFindData 标签信息

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录像标签之前, 必须先调用 `NET_DVR_FindRecordLabel` 得到当前的查找句柄。此接口用于获取一条已查找到的标签信息, 若要获取全部的已查找到的标签信息, 需要循环调用此接口。

[返回目录](#)

5.14.27 停止搜索录像标签 `NET_DVR_StopFindLabel`

函数: `BOOL NET_DVR_StopFindLabel(LONG IFindHandle)`

参数: `[in]IFindHandle` `NET_DVR_FindRecordLabel` 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](#)

回放的其他操作

5.14.28 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

返回值: `TRUE` 表示成功, `FALSE` 表示失败。接口返回失败请调用 [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](#)

5.14.29 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](#)

5.14.30 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](#)

5.14.31 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

5.14.32 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](#)

5.14.33 按时间下载录像文件 NET_DVR_GetFileByTime_V40

- 函 数:** LONG NET_DVR_GetFileByTime_V40(LONG IUserID, char *sSavedFileName, LPNET_DVR_PLAYCOND pDownloadCond)

- 参 数:**
- [in]IUserID NET_DVR_Login_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), 比如: "C:\\test.mp4"
 - [in]pDownloadCond 下载条件, 包括通道号、开始时间、结束时间等
- 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](#)

5.14.34 控制录像下载的状态 NET_DVR_PlayBackControl_V40

- 函 数:** BOOL NET_DVR_PlayBackControl_V40(LONG IPlayHandle, DWORD dwControlCode, LPVOID lpInBuffer, DWORD dwInLen, LPVOID lpOutBuffer, DWORD *lpOutLen)
- 参 数:**
- [in]IPlayHandle 播放句柄, NET_DVR_GetFileByName 或 NET_DVR_GetFileByTime_V40 的返回值
 - [in]dwControlCode 控制录像回放状态命令, 详见表 5.23
 - [in]lpInBuffer 指向输入参数的指针, 详见表 5.24
 - [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 5.24
 - [out]lpOutLen 输出参数的长度

表 5.23 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, 如表 5.24 所示。
- 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)。
 - 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 5.24 下载控制参数

状态命令宏定义	状态命令说明	lpInBuf	lpOutBuf
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](#)

5.14.35 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](#)

5.14.36 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，返回 200 表明出现网络异常。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说 明： 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

5.14.37 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](#)

5.14.38 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](#)

5.14.39 流 ID 方式对某一时间段录像文件进行加锁

NET_DVR_LockStreamFileByTime

函 数: BOOL NET_DVR_LockStreamFileByTime(LONG IUserID, LPNET_DVR_STREAM_TIME_LOCK lpLockPara, LPNET_DVR_LOCK_RETURN lpLockReturn)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]lpLockPara 录像加锁参数, 包括流信息 (ID 或通道号)、录像类型、开始和结束时间、锁定持续时间等
[out]lpLockReturn 录像加锁结果, 包括实际锁定的开始时间和结束时间

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

5.14.40 流 ID 方式对某一时间段录像文件进行解锁

NET_DVR_UnlockStreamFileByTime

函 数: BOOL NET_DVR_UnlockStreamFileByTime(LONG IUserID, LPNET_DVR_STREAM_TIME_LOCK lpLockPara, LPNET_DVR_LOCK_RETURN lpLockReturn)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IpLockPara 录像解锁参数, 包括流信息 (ID 或通道号)、录像类型、开始和结束时间、锁定持续时间等
[in]IpLockReturn 录像解锁结果, 包括实际解锁的开始时间和结束时间
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

备份文件

5.14.41 获取设备磁盘列表 **NET_DVR_GetDiskList**

函 数: BOOL NET_DVR_GetDiskList(LONG IUserID, LPNET_DVR_DISKABILITY_LIST IpDiskList)
参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[out]IpDiskList 设备可用备份磁盘信息结构
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
说 明: 该接口用于获取设备可用备份文件的磁盘资源信息, 在备份文件功能接口的输入参数中需要用到。

[返回目录](#)

5.14.42 按文件名备份录像文件 **NET_DVR_BackupByName**

函 数: LONG NET_DVR_BackupByName(LONG IUserID, LPNET_DVR_BACKUP_NAME_PARAM IpBackupByName)
参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IpBackupByName 备份文件参数结构
返回值: -1 表示失败, 其他值作为 NET_DVR_GetBackupProgress, NET_DVR_StopBackup 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: 在调用该接口开始文件备份前必须调用接口 NET_DVR_GetDiskList 获取当前设备可用的磁盘列表信息, 返回的磁盘描述字段用于指定此接口中 IpBackupByName 参数中的备份磁盘描述字段。其中参数 byDiskDes 是由接口 NET_DVR_GetDiskList 获取的参数得到。

[返回目录](#)

5.14.43 按时间段备份录像文件 **NET_DVR_BackupByTime**

函 数: LONG NET_DVR_BackupByTime(LONG IUserID, LPNET_DVR_BACKUP_TIME_PARAM IpBackupBytime)
参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IpBackupByName 备份文件参数结构

返回值: -1 表示失败, 其他值作为 NET_DVR_GetBackupProgress, NET_DVR_StopBackup 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: 在调用该接口开始文件备份前必须调用接口 NET_DVR_GetDiskList 获取当前设备可用的磁盘列表信息, 返回的磁盘描述字段用于指定此接口中 lpBackupByTime 参数中的备份磁盘描述字段。
对于审讯机设备 byDiskDes 和 byWithPlayer 参数无效, 其他设备参数 byDiskDes 是由接口 NET_DVR_GetDiskList 获取

[返回目录](#)

5.14.44 获取备份的进度 **NET_DVR_GetBackupProgress**

函数: BOOL NET_DVR_GetBackupProgress(LONG lHandle, DWORD* pState)

参数: [in]lHandle NET_DVR_BackupByName 或 NET_DVR_BackupByTime 的返回值
[out] pState 当前备份的进度, 进度值的取值范围为[0,100), 其他值的定义见表 5.25

表 5.25 备份进度

宏定义	宏定义值	含义
BACKUP_SUCCESS	100	备份完成
BACKUP_CHANGE_DEVICE	101	备份设备已满, 更换设备继续备份
BACKUP_SEARCH_DEVICE	300	正在搜索备份设备
BACKUP_SEARCH_FILE	301	正在搜索录像文件或者图片
BACKUP_EXCEPTION	400	备份异常
BACKUP_FAIL	500	备份失败
BACKUP_TIME_SEG_NO_FILE	501	时间段内无录像文件或者图片
BACKUP_NO_RESOURCE	502	申请不到资源
BACKUP_DEVICE_LOW_SPACE	503	备份设备容量不足
BACKUP_DISK_FINALIZED	504	刻录光盘封盘
BACKUP_DISK_EXCEPTION	505	刻录光盘异常
BACKUP_DEVICE_NOT_EXIST	506	备份设备不存在
BACKUP_OTNER_BACKUP_WORK	507	有其他备份操作在进行
BACKUP_USER_NO_RIGHT	508	用户没有操作权限
BACKUP_OPERATE_FAIL	509	操作失败

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。

说明: 在进度为 100 或者备份出错时, 需调用 NET_DVR_StopBackup()停止备份。

[返回目录](#)

5.14.45 停止备份 **NET_DVR_StopBackup**

函数: BOOL NET_DVR_StopBackup(LONG lHandle)

参 数: [in]IHandle NET_DVR_BackupByName 或 NET_DVR_BackupByTime 的返回值，
或者 NET_DVR_BackupPicture 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](#)

5.15 图片的查找、回放及备份

查找图片

5.15.1 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，包括查找的通道号、图片类型和起止时间等

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](#)

5.15.2 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 查找到的图片结果信息，包括图片名称、时间、类型等

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 5.26 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 5.26 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](#)

5.15.3 关闭图片查找，释放资源 **NET_DVR_CloseFindPicture**

函 数： BOOL NET_DVR_CloseFindPicture(LONG IFindHandle)

参 数： [in]IFindHandle 图片查找句柄，NET_DVR_FindPicture 的返回值

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

说 明：

[返回目录](#)

图片智能检索（后检索）

5.15.4 开始智能图片检索 **NET_DVR_SmartSearchPicture**

函 数： LONG NET_DVR_SmartSearchPicture(LONG IUserID, NET_DVR_SMART_SEARCH_PIC_PARA *pFindParam)

参 数： [in]IUserID NET_DVR_Login_V40 的返回值

[in]pFindParam 智能图片搜索信息，包括智能查找类型、通道号和起止时间等

返回值： -1 表示失败，其他值作为 NET_DVR_CloseSmartSearchPicture 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_FindNextSmartPicture 接口来获取图片信息。

[返回目录](#)

5.15.5 逐个获取搜索到的智能图片信息 **NET_DVR_FindNextSmartPicture**

函 数： LONG NET_DVR_FindNextSmartPicture(LONG IFindHandle, LPNET_DVR_SMART_SEARCH_PIC_RET lpFindData)

参 数： [in]IFindHandle 图片智能检索句柄，NET_DVR_SmartSearchPicture 的返回值

[out]lpFindData 查找到的智能图片结果信息，包括图片名称、时间、事件类型等

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 5.27 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 5.27 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_NOFOUND	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](#)

5.15.6 关闭图片智能检索，释放资源 **NET_DVR_CloseSmartSearchPicture**

函 数: BOOL NET_DVR_CloseSmartSearchPicture(LONG IFindHandle)

参 数: [in]IFindHandle 图片智能检索句柄，NET_DVR_SmartSearchPicture 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

5.15.7 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。

[返回目录](#)

备份图片

5.15.8 备份图片 **NET_DVR_BackupPicture**

函 数: LONG NET_DVR_BackupPicture(LONG IUserID, NET_DVR_BACKUP_PICTURE_PARAM *lpBackupPicture)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值

[in]lpBackupPicture 图片备份参数

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](#)

5.15.9 获取备份的进度 **NET_DVR_GetBackupProgress**

函 数: BOOL NET_DVR_GetBackupProgress(LONG IHandle, DWORD* pState)

参 数: [in]IHandle NET_DVR_BackupPicture 的返回值

[out]pState

当前备份的进度，进度值的取值范围为[0,100)，其他值的定义见表 5.28

表 5.28 备份进度

pState 宏定义	宏定义值	含义
BACKUP_SUCCESS	100	备份完成
BACKUP_CHANGE_DEVICE	101	备份设备已满，更换设备继续备份
BACKUP_SEARCH_DEVICE	300	正在搜索备份设备
BACKUP_SEARCH_FILE	301	正在搜索录像文件或者图片
BACKUP_EXCEPTION	400	备份异常
BACKUP_FAIL	500	备份失败
BACKUP_TIME_SEG_NO_FILE	501	时间段内无录像文件或者图片
BACKUP_NO_RESOURCE	502	申请不到资源
BACKUP_DEVICE_LOW_SPACE	503	备份设备容量不足
BACKUP_DISK_FINALIZED	504	刻录光盘封盘
BACKUP_DISK_EXCEPTION	505	刻录光盘异常
BACKUP_DEVICE_NOT_EXIST	506	备份设备不存在
BACKUP_OTNER_BACKUP_WORK	507	有其他备份操作在进行
BACKUP_USER_NO_RIGHT	508	用户没有操作权限
BACKUP_OPERATE_FAIL	509	操作失败

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。

说 明： 在进度为 100 或者备份出错时，需调用 NET_DVR_StopBackup() 停止备份。

[返回目录](#)

5.15.10 停止备份 **NET_DVR_StopBackup**

函 数： BOOL NET_DVR_StopBackup(LONG lHandle)

参 数： [in]lHandle NET_DVR_BackupPicture 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 布防、撤防

设置报警等信息上传的回调函数

5.16.1 注册回调函数，接收报警消息 **NET_DVR_SetDVRMessageCallBack_V30**

函 数： BOOL NET_DVR_SetDVRMessageCallBack_V30(MSGCallBack fMessageCallBack, void* pUser)

参 数： [in]fMessageCallBack 报警信息回调函数

[in]pUser 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 5.29

[out]pAlarmer 报警设备信息

[out]pAlarmInfo 报警信息，详见表 5.30

[out]dwBufLen 报警信息缓存大小

[out]pUser 用户数据

表 5.29 报警信息类型

ICommand 宏定义	宏定义值	含义
COMM_ALARM	0x1100	V3.0 以下版本支持的设备的报警信息上传
COMM_ALARM_V30	0x4000	V3.0 以上版本支持的设备的报警信息上传
COMM_IPCCFG	0x4001	混合型 DVR 在 IPC 接入配置改变时的报警信息上传
COMM_IPCCFG_V31	0x4002	混合型 DVR 在 IPC 接入配置改变时的报警信息上传（扩展）
COMM_ALARM_HOT_SPARE	0x4006	热备异常报警（N+1 模式异常报警）
COMM_ALARM_V40	0x4007	移动侦测、视频丢失、遮挡、IO 信号量等报警信息主动上传，报警数据为可变长
COMM_ALARM_VQD	0x6000	VQD 诊断报警信息上传
COMM_VEHICLE_CONTROL_ALARM	0x3059	黑白名单车辆报警上传

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

说 明： 该接口中回调函数的第一个参数（ICommand）和第三个参数（pAlarmInfo）是密切关联的，其关系见表 5.30。

表 5.30 报警信息结构

消息类型 ICommand	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM	V3.0 以下版本支持的设备的报警信息	NET_DVR_ALARMINFO
COMM_ALARM_V30	V3.0 以上版本支持的设备的报警信息	NET_DVR_ALARMINFO_V30
COMM_IPCCFG	混合型 DVR 在 IPC 接入配置改变时的报警信息	NET_DVR_IPALARMINFO
COMM_IPCCFG_V31	混合型 DVR 在 IPC 接入配置改变时的报警信息（扩展）	NET_DVR_IPALARMINFO_V31
COMM_ALARM_HOT_SPARE	热备异常报警（N+1 模式异常报警）信息	NET_DVR_ALARM_HOT_SPARE

COMM_ALARM_V40	移动侦测、视频丢失、遮挡、IO 信号量等报警信息	NET_DVR_ALARMINFO_V40
COMM_ALARM_VQD	VQD 诊断报警信息	NET_DVR_VQD_DIAGNOSE_INFO
COMM_VEHICLE_CONTROL_ALARM	黑白名单车辆报警上传	NET_DVR_VEHICLE_CONTROL_ALARM

[返回目录](#)

布防撤防

5.16.2 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

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： 使用该接口支持上传 V3.0 以上版本支持的设备的报警结构。启动布防前，需要调用注册回调函数的接口（如 [NET_DVR_SetDVRMessageCallBack_V30](#)）才能获取到上传的报警等信息。

[返回目录](#)

5.16.3 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](#)

5.17 监听报警

5.17.1 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 5.31
[out]pAlarmer	报警设备信息
[out]pAlarmInfo	报警信息，详见表 5.31
[out]dwBufLen	报警信息缓存大小
[out]pUser	用户数据

表 5.31 报警信息类型

ICommand 宏定义	宏定义值	含义
COMM_ALARM	0x1100	V3.0 以下版本支持的设备的报警信息上传
COMM_ALARM_V30	0x4000	V3.0 以上版本支持的设备的报警信息上传
COMM_IPCCFG	0x4001	混合型 DVR 在 IPC 接入配置改变时的报警信息上传
COMM_IPCCFG_V31	0x4002	混合型 DVR 在 IPC 接入配置改变时的报警信息上传（扩展）
COMM_ALARM_HOT_SPARE	0x4006	热备异常报警（N+1 模式异常报警）
COMM_ALARM_V40	0x4007	移动侦测、视频丢失、遮挡、IO 信号量等报警信息主动上传，报警数据为可变长
COMM_ALARM_VQD	0x6000	VQD 诊断报警信息上传

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 5.32。

表 5.32 报警信息结构

消息类型 ICommand	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM	V3.0 以下版本支持的设备的报警信息	NET_DVR_ALARMINFO
COMM_ALARM_V30	V3.0 以上版本支持的设备的报警信息	NET_DVR_ALARMINFO_V30
COMM_IPCCFG	混合型 DVR 在 IPC 接入配置改变时的报警信息	NET_DVR_IPALARMINFO
COMM_IPCCFG_V31	混合型 DVR 在 IPC 接入配置改变时的报警信息（扩展）	NET_DVR_IPALARMINFO_V31
COMM_ALARM_HOT_SPARE	热备异常报警（N+1 模式异常报警）信息	NET_DVR_ALARM_HOT_SPARE
COMM_ALARM_V40	移动侦测、视频丢失、遮挡、IO 信号量等报警信息	NET_DVR_ALARMINFO_V40
COMM_ALARM_VQD	VQD 诊断报警信息	NET_DVR_VQD_DIAGNOSE_INFO

- 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。

[返回目录](#)

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

函 数: BOOL NET_DVR_StopListen_V30(LONG lListenHandle)

参 数: [in]lListenHandle 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](#)

5.18 语音对讲、转发及广播

语音对讲

5.18.1 启动语音对讲 **NET_DVR_StartVoiceCom_V30**

函 数: LONG NET_DVR_StartVoiceCom_V30(LONG lUserID, DWORD dwVoiceChan, BOOL bNeedCBNoEncData, 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]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 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](#)

5.18.2 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](#)

5.18.3 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

5.18.4 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 IUserID, DWORD dwVoiceChan, 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]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 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](#)

5.18.5 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](#)

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

函 数： BOOL NET_DVR_StopVoiceCom (LONG IVoiceComHandle)

参 数： [in]IVoiceComHandle 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。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明：

[返回目录](#)

语音广播

5.18.7 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](#)

5.18.8 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 IUserID, DWORD dwVoiceChan)

参数: [in]IUserID 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](#)

5.18.9 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](#)

5.18.10 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](#)

音频编解码(Windows 32 位系统支持)

G722 音频编解码

5.18.11 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](#)

5.18.12 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](#)

5.18.13 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](#)

5.18.14 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](#)

5.18.15 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。接口返回失败请调用 [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_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](#)

5.18.16 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

5.18.17 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_DecodeG711Frame` 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](#)

5.18.18 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：

<code>[in]iType</code>	编码类型：0-Mu law 编码，非 0-A law 编码
<code>[in]pInBuffer</code>	输入缓冲区，编码数据大小为 160 字节
<code>[out]pOutBuffer</code>	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](#)

5.19 远程参数配置

通用参数配置

5.19.1 获取设备的配置信息 `NET_DVR_GetDVRConfig`

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

参数：

<code>[in]IUserID</code>	用户 ID 号，NET_DVR_Login_V40 的返回值
<code>[in]dwCommand</code>	设备配置命令，详见表 5.33
<code>[in]IChannel</code>	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 5.34
<code>[out]lpOutBuffer</code>	接收数据的缓冲指针，详见表 5.34
<code>[in]dwOutBufferSize</code>	接收数据的缓冲长度(以字节为单位)，不能为 0
<code>[out]lpBytesReturned</code>	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 5.34 所示。

- 通道号是指设备视频通道号，通过注册设备（[NET DVR Login V30](#)）返回的设备信息（NET_DVR_DEVICEINFO_V30）获取模拟通道个数（byChanNum）、模拟通道起始通道号（byStartChan）和设备支持的最大 IP 通道数（byIPChanNum+ byHighDChanNum*256）、数字通道起始通道号（byStartDChan）。

表 5.33 参数获取命令

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_GET_DEVICECFG_V40	获取设备参数(扩展)	1100
NET_DVR_GET_TIMECFG	获取时间参数	118
NET_DVR_GET_NETCFG_V30	获取网络参数	1000
NET_DVR_GET_PICCFG_V40	获取图像参数	6179
NET_DVR_GET_COMPRESSCFG_V30	获取压缩参数	1040
NET_DVR_GET_RECORDCFG_V30	获取录像参数	1004
NET_DVR_GET_DECODERCFG_V30	获取（云台）解码器参数	1042
NET_DVR_GET_RS232CFG_V30	获取 232 串口参数	1036
NET_DVR_GET_ALARMINCFG_V40	获取报警输入参数	6181
NET_DVR_GET_ALARMOUTCFG_V30	获取报警输出参数	1026
NET_DVR_GET_PREVIEWCFG_V30	获取本地预览参数	1044
NET_DVR_GET_PREVIEWCFG_AUX_V30	获取辅助预览参数	1046
NET_DVR_GET_VIDEOOUTCFG_V30	获取视频输出参数	1028
NET_DVR_GET_USERCFG_V40	获取用户参数	6187
NET_DVR_GET_EXCEPTIONCFG_V40	获取异常参数	6177
NET_DVR_GET_ZONEANDDST	获取时区和夏时制参数	128
NET_DVR_GET_SHOWSTRING_V30	获取叠加字符参数	1030
NET_DVR_GET_AUXOUTCFG	获取报警触发辅助输出参数	140
NET_DVR_GET_NETAPPCFG	获取网络应用参数(NTP/DDNS)	222
NET_DVR_GET_NTPCFG	获取网络应用参数(NTP)	224
NET_DVR_GET_DDNSCFG_V30	获取网络应用参数(DDNS)	1010
NET_DVR_GET_EMAILCFG_V30	获取网络应用参数(EMAIL)	1012
NET_DVR_GET_NFSCFG	获取 NFS（网络文件系统）参数	230
NET_DVR_GET_IPPARACFG_V40	获取 IP 接入配置参数	1062
NET_DVR_GET_IPALARMINCFG_V40	获取 IP 报警输入接入配置参数	6183
NET_DVR_GET_IPALARMOUTCFG_V40	获取 IP 报警输出接入配置参数	6185
NET_DVR_GET_DIGITAL_CHAN_SECURITY_STATUS	获取数字通道对应设备安全状态	13001
NET_DVR_GET_HDCFG	获取硬盘管理参数	1054
NET_DVR_GET_HDCFG_V40	获取硬盘管理参数	6122
NET_DVR_GET_HDGROUP_CFG_V40	获取盘组管理参数	6192

NET_DVR_GET_COMPRESSCFG_AUD	获取语音对讲音频参数	1058
NET_DVR_GET_NET_DISKCFG	获取网络硬盘接入参数	1038
NET_DVR_GET_SNMPCFG	获取 SNMP 参数	1112
NET_DVR_GET_NETCFG_MULTI	获取多网卡配置参数	1161
NET_DVR_GET_NETWORK_BONDING	获取 BONDING 网卡参数	1254
NET_DVR_GET_HOLIDAY_PARAM_CFG	获取节假日参数	1240
NET_DVR_GET_MOTION_HOLIDAY_HANDLE	获取移动侦测假日报警处理方式	1242
NET_DVR_GET_VILOST_HOLIDAY_HANDLE	获取视频信号丢失假日报警处理方式	1244
NET_DVR_GET_HIDE_HOLIDAY_HANDLE	获取遮盖假日报警处理方式	1246
NET_DVR_GET_ALARMIN_HOLIDAY_HANDLE	获取报警输入假日报警处理方式	1248
NET_DVR_GET_ALARMOUT_HOLIDAY_HANDLE	获取报警输出假日报警处理方式	1250
NET_DVR_GET_HOLIDAY_RECORD	获取假日录像参数	1252
NET_DVR_GET_LINK_STATUS	获取通道的工作状态	1256
NET_DVR_GET_DISK_QUOTA_CFG	获取磁盘配额信息	1278
NET_DVR_GET_JPEG_CAPTURE_CFG	获取设备抓图配置	1280
NET_DVR_GET_SCHED_CAPTURECFG	获取抓图计划	1282
NET_DVR_GET_FTPCFG	获取抓图 FTP 上传参数	134
NET_DVR_GET_FTPCFG_SECOND	默认的第一个 FTP 有问题则该 FTP 生效	6103
NET_DVR_GET_VIDEO_INPUT_EFFECT	获取通道视频输入图像参数	1286
NET_DVR_GET_DVR_SYNCHRONOUS_IPC	获取是否为前端 IPC 同步设备参数	6005
NET_DVR_GET_DEVICE_NET_USING_INFO	获取当前设备网络资源使用情况	6009
NET_DVR_GET_DISK_RAID_INFO	获取磁盘 Raid 信息	6001
NET_DVR_GET_ACCESS_CAMERA_INFO	获取前端相机信息	6201
NET_DVR_GET_DRAWFRAME_DISK_QUOTA_CFG	获取抽帧通道磁盘配额	6109
NET_DVR_GET_NAT_CFG	获取 NAT 映射参数	6111
NET_DVR_GET_AES_KEY	获取设备 AES 加密密钥	6113
NET_DVR_GET_POE_CFG	获取 POE 参数	6114
NET_DVR_GET_CUSTOM_PRO_CFG	获取自定义协议参数	6116
NET_DVR_GET_STREAM_CABAC	获取码流压缩性能选项	6118
NET_DVR_GET_ESATA_MINISAS_USAGE_CFG	获取 eSATA 和 miniSAS 用途	6120
NET_DVR_GET_POE_CHANNEL_ADD_MODE	获取 POE 通道添加方式	6124
NET_DVR_GET_DIGITAL_CHANNEL_STATE	获取设备数字通道状态	6126
NET_DVR_GET_WD1_CFG	获取 WD1 使能开关状态	6136
NET_DVR_GET_N_PLUS_ONE_WORK_MODE	获取 N+1 工作模式	6168
NET_DVR_GET_HD_STATUS	获取硬盘状态	6170

NET_DVR_GET_RAID_BACKGROUND_TASK_SPEED	获取 RAID 后台任务速度	6175
NET_DVR_GET_VIDEO_AUDIOIN_CFG	获取视频的音频输入参数	9118
NET_DVR_GET_CMS_CFG	获取平台参数	2070
NET_DVR_GET_DVR_SYNCHRONOUS_IPC	获取是否为前端 IPC 同步设备参数	6005
NET_DVR_GET_DEVICE_NET_USING_INFO	获取当前设备网络资源使用情况	6009
NET_DVR_GET_BONJOUR_CFG	获取 Bonjour 信息	6127
NET_DVR_GET_SOCKS_CFG	获取 SOCKS	6130
NET_DVR_GET_QOS_CFG	获取 QoS 信息	6132
NET_DVR_GET_HTTPS_CFG	获取 HTTPS 信息	6134
NET_DVR_GET_RECORD_PACK	获取录像打包参数	6301
NET_DVR_GET_CLOUD_STORAGE_CFG	获取云存储工作模式	6303

表 5.34 获取设备参数

dwCommand 宏定义	IChannel 取值	lpOutBuffer 对应结构体
NET_DVR_GET_DEVICECFG_V40	无效	NET_DVR_DEVICECFG_V40
NET_DVR_GET_TIMECFG	无效	NET_DVR_TIME
NET_DVR_GET_NETCFG_V30	无效	NET_DVR_NETCFG_V30
NET_DVR_GET_PICCFG_V40	通道号	NET_DVR_PICCFG_V40
NET_DVR_GET_COMPRESSCFG_V30	通道号	NET_DVR_COMPRESSIONCFG_V30
NET_DVR_GET_RECORDCFG_V30	通道号	NET_DVR_RECORD_V30
NET_DVR_GET_DECODERCFG_V30	通道号	NET_DVR_DECODERCFG_V30
NET_DVR_GET_RS232CFG_V30	无效	NET_DVR_RS232CFG_V30
NET_DVR_GET_ALARMINCFG_V40	报警输入号, 从 0 开始	NET_DVR_ALARMINCFG_V40
NET_DVR_GET_ALARMOUTCFG_V30	报警输出号, 从 0 开始	NET_DVR_ALARMOUTCFG_V30
NET_DVR_GET_PREVIEWCFG_V30	无效	NET_DVR_PREVIEWCFG_V30
NET_DVR_GET_PREVIEWCFG_AUX_V30	无效	NET_DVR_PREVIEWCFG_V30
NET_DVR_GET_VIDEOOUTCFG_V30	无效	NET_DVR_VIDEOOUT_V30
NET_DVR_GET_USERCFG_V40	组号, 从 0 开始, 每组 32 个用户	NET_DVR_USER_V40
NET_DVR_GET_EXCEPTIONCFG_V40	组号, 从 0 开始, 每组 32 种异常	NET_DVR_EXCEPTION_V40
NET_DVR_GET_ZONEANDDST	无效	NET_DVR_ZONEANDDST
NET_DVR_GET_SHOWSTRING_V30	通道号	NET_DVR_SHOWSTRING_V30
NET_DVR_GET_AUXOUTCFG	无效	NET_DVR_AUXOUTCFG
NET_DVR_GET_NETAPPCFG	无效	NET_DVR_NETAPPCFG
NET_DVR_GET_NTPCFG	无效	NET_DVR_NTTPARA

NET_DVR_GET_DDNSCFG_V30	无效	NET_DVR_DDNSPARA_V30
NET_DVR_GET_EMAILCFG_V30	无效	NET_DVR_EMAILCFG_V30
NET_DVR_GET_NFSCFG	无效	NET_DVR_NFSCFG
NET_DVR_GET_IPPARACFG_V40	无效	NET_DVR_IPPARACFG_V40
NET_DVR_GET_IPALARMINCFG_V40	无效	NET_DVR_IPALARMINCFG_V40
NET_DVR_GET_IPALARMOUTCFG_V40	无效	NET_DVR_IPALARMOUTCFG_V40
NET_DVR_GET_DIGITAL_CHAN_SECURITY_STATUS	通道号	NET_DVR_DIGITAL_CHANNEL_SECURITY_STATUS
NET_DVR_GET_HDCFG	无效	NET_DVR_HDCFG
NET_DVR_GET_HDCFG_V40	组号，从 0 开始，每组 33 个硬盘	NET_DVR_HDCFG
NET_DVR_GET_HDGROUP_CFG_V40	组号，从 0 开始，每组 16 个硬盘组	NET_DVR_HDGROUP_CFG_V40
NET_DVR_GET_COMPRESSCFG_AUD	无效	NET_DVR_COMPRESSION_AUDIO
NET_DVR_GET_NET_DISKCFG	无效	NET_DVR_NET_DISKCFG
NET_DVR_GET_SNMPCFG	无效	NET_DVR_SNMPCFG
NET_DVR_GET_NETCFG_MULTI	无效	NET_DVR_NETCFG_MULTI
NET_DVR_GET_NETWORK_BONDING	无效	NET_DVR_NETWORK_BONDING
NET_DVR_GET_HOLIDAY_PARAM_CFG	无效	NET_DVR_HOLIDAY_PARAM_CFG
NET_DVR_GET_MOTION_HOLIDAY_HANDLE	通道号	NET_DVR_HOLIDAY_HANDLE
NET_DVR_GET_VIOLST_HOLIDAY_HANDLE	通道号	NET_DVR_HOLIDAY_HANDLE
NET_DVR_GET_HIDE_HOLIDAY_HANDLE	通道号	NET_DVR_HOLIDAY_HANDLE
NET_DVR_GET_ALARMIN_HOLIDAY_HANDLE	报警输入号，从 0 开始	NET_DVR_HOLIDAY_HANDLE
NET_DVR_GET_ALARMOUT_HOLIDAY_HANDLE	报警输出号，从 0 开始	NET_DVR_HOLIDAY_HANDLE
NET_DVR_GET_HOLIDAY_RECORD	通道号	NET_DVR_HOLIDAY_RECORD
NET_DVR_GET_LINK_STATUS	组号	NET_DVR_LINK_STATUS
NET_DVR_GET_DISK_QUOTA_CFG	通道号	NET_DVR_DISK_QUOTA_CFG
NET_DVR_GET_JPEG_CAPTURE_CFG	通道号	NET_DVR_JPEG_CAPTURE_CFG
NET_DVR_GET_SCHED_CAPTURECFG	通道号	NET_DVR_SCHED_CAPTURECFG
NET_DVR_GET_FTPCFG	无效	NET_DVR_FTPCFG
NET_DVR_GET_FTPCFG_SECOND	无效	NET_DVR_FTPCFG
NET_DVR_GET_VIDEO_INPUT_EFFECT	通道号	NET_DVR_VIDEO_INPUT_EFFECT
NET_DVR_GET_DVR_SYNCHRONOUS_IPC	无效	NET_DVR_SYNCHRONOUS_IPC
NET_DVR_GET_DEVICE_NET_USING_INFO	无效	NET_DVR_DEVICE_NET_USING_INFO
NET_DVR_GET_DISK_RAID_INFO	无效	NET_DVR_DISK_RAID_INFO
NET_DVR_GET_ACCESS_CAMERA_INFO	通道号	NET_DVR_ACCESS_CAMERA_INFO

NET_DVR_GET_DRAWFRAME_DISK_QUOTA_CFG	无效	NET_DVR_DRAWFRAME_DISK_QUOTA_CFG
NET_DVR_GET_NAT_CFG	无效	NET_DVR_NAT_CFG
NET_DVR_GET_AES_KEY	无效	NET_DVR_AES_KEY_INFO
NET_DVR_GET_POE_CFG	无效	NET_DVR_POE_CFG
NET_DVR_GET_CUSTOM_PRO_CFG	协议序号, 从 1 开始	NET_DVR_CUSTOM_PROTOCOL
NET_DVR_GET_STREAM_CABAC	通道号	NET_DVR_STREAM_CABAC
NET_DVR_GET_ESATA_MINISAS_USAGE_CFG	无效	NET_DVR_ESATA_MINISAS_USAGE
NET_DVR_GET_POE_CHANNEL_ADD_MODE	无效	NET_DVR_POE_CHANNEL_ADD_MODE
NET_DVR_GET_DIGITAL_CHANNEL_STATE	无效	NET_DVR_DIGITAL_CHANNEL_STATE
NET_DVR_GET_WD1_CFG	通道号	NET_DVR_WD1_CFG
NET_DVR_GET_N_PLUS_ONE_WORK_MODE	无效	NET_DVR_N_PLUS_ONE_WORK_MODE_CFG
NET_DVR_GET_HD_STATUS	无效	NET_DVR_HD_STATUS
NET_DVR_GET_RAID_BACKGROUND_TASK_SPEED	无效	NET_DVR_RAID_BTS_CFG
NET_DVR_GET_VIDEO_AUDIOIN_CFG	语音对讲通道号, 从 0 开始	NET_DVR_VIDEO_AUDIOIN_CFG
NET_DVR_GET_CMS_CFG	无效	NET_DVR_CMS_PARAM
NET_DVR_GET_DVR_SYNCHRONOUS_IPC	通道号	NET_DVR_SYNCHRONOUS_IPC
NET_DVR_GET_DEVICE_NET_USING_INFO	无效	NET_DVR_DEVICE_NET_USING_INFO
NET_DVR_GET_BONJOUR_CFG	无效	NET_DVR_BONJOUR_CFG
NET_DVR_GET_SOCKS_CFG	无效	NET_DVR_SOCKS_CFG
NET_DVR_GET_QOS_CFG	无效	NET_DVR_QOS_CFG
NET_DVR_GET_HTTPS_CFG	无效	NET_DVR_HTTPS_CFG
NET_DVR_GET_RECORD_PACK	无效	NET_DVR_RECORD_PACK
NET_DVR_GET_CLOUD_STORAGE_CFG	无效	NET_DVR_CLOUD_STORAGE_CFG

[返回目录](#)

5.19.2 设置设备的配置信息 **NET_DVR_SetDVRConfig**

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

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 5.35
[in]lChannel 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 5.36
[in]lpInBuffer 输入数据的缓冲指针, 详见表 5.36
[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 5.36 所示。

- 通道号是指设备视频通道号，通过注册设备（[NET DVR Login V30](#)）返回的设备信息（NET_DVR_DEVICEINFO_V30）获取模拟通道个数（byChanNum）、模拟通道起始通道号（byStartChan）和设备支持的最大 IP 通道数（byIPChanNum+ byHighDChanNum*256）、数字通道起始通道号（byStartDChan）。

表 5.35 参数设置命令

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_SET_DEVICECFG_V40	设置设备参数(扩展)	1101
NET_DVR_SET_TIMECFG	设置时间参数	119
NET_DVR_SET_NETCFG_V30	设置网络参数	1001
NET_DVR_SET_PICCFG_V40	设置图像参数	6180
NET_DVR_SET_COMPRESSCFG_V30	设置压缩参数	1041
NET_DVR_SET_RECORDCFG_V30	设置录像参数	1005
NET_DVR_SET_DECODERCFG_V30	设置（云台）解码器参数	1043
NET_DVR_SET_RS232CFG_V30	设置 232 串口参数	1037
NET_DVR_SET_ALARMINCFG_V40	设置报警输入参数	6182
NET_DVR_SET_ALARMOUTCFG_V30	设置报警输出参数	1027
NET_DVR_SET_PREVIEWCFG_V30	设置本地预览参数	1045
NET_DVR_SET_PREVIEWCFG_AUX_V30	设置辅助预览参数	1047
NET_DVR_SET_VIDEOOUTCFG_V30	设置视频输出参数	1029
NET_DVR_SET_USERCFG_V40	设置用户参数	6188
NET_DVR_SET_EXCEPTIONCFG_V40	设置异常参数	6178
NET_DVR_SET_ZONEANDDST	设置时区和夏时制参数	129
NET_DVR_SET_SHOWSTRING_V30	设置叠加字符参数	1031
NET_DVR_SET_AUXOUTCFG	设置报警触发辅助输出参数	141
NET_DVR_SET_NETAPPCFG	设置网络应用参数(NTP/DDNS)	223
NET_DVR_SET_NTPCFG	设置网络应用参数(NTP)	225
NET_DVR_SET_DDNSCFG_V30	设置网络应用参数(DDNS)	1011
NET_DVR_SET_EMAILCFG_V30	设置网络应用参数(EMAIL)	1013
NET_DVR_SET_NFSCFG	设置 NFS（网络文件系统）参数	231
NET_DVR_SET_IPPARACFG_V40	设置 IP 接入配置参数	1063
NET_DVR_SET_HDCFG	设置硬盘管理参数	1055
NET_DVR_SET_HDCFG_V40	设置硬盘管理参数	6123
NET_DVR_SET_HDGROUP_CFG_V40	设置盘组管理参数	6193
NET_DVR_SET_COMPRESSCFG_AUD	设置语音对讲音频参数	1059
NET_DVR_SET_NET_DISKCFG	设置网络硬盘接入参数	1039
NET_DVR_SET_SNMPCFG	设置 SNMP 参数	1113

NET_DVR_SET_NETCFG_MULTI	设置多网卡配置参数	1162
NET_DVR_SET_NETWORK_BONDING	设置 BONDING 网卡参数	1255
NET_DVR_SET_HOLIDAY_PARAM_CFG	设置节假日参数	1241
NET_DVR_SET_MOTION_HOLIDAY_HANDLE	设置移动侦测假日报警处理方式	1243
NET_DVR_SET_VIOLST_HOLIDAY_HANDLE	设置视频信号丢失假日报警处理方式	1245
NET_DVR_SET_HIDE_HOLIDAY_HANDLE	设置遮盖假日报警处理方式	1247
NET_DVR_SET_ALARMIN_HOLIDAY_HANDLE	设置报警输入假日报警处理方式	1249
NET_DVR_SET_ALARMOUT_HOLIDAY_HANDLE	设置报警输出假日报警处理方式	1251
NET_DVR_SET_HOLIDAY_RECORD	设置假日录像参数	1253
NET_DVR_SET_DISK_QUOTA_CFG	设置磁盘配额信息	1279
NET_DVR_SET_JPEG_CAPTURE_CFG	设置设备抓图配置	1281
NET_DVR_SET_SCHED_CAPTURECFG	设置抓图计划	1283
NET_DVR_SET_FTPCFG	设置抓图 FTP 上传参数	135
NET_DVR_SET_FTPCFG_SECOND	默认的的第一个 FTP 有问题则该 FTP 生效	6104
NET_DVR_SET_VIDEO_INPUT_EFFECT	设置通道视频输入图像参数	1287
NET_DVR_SET_DVR_SYNCHRONOUS_IPC	设置是否为前端 IPC 同步设备参数	6006
NET_DVR_SET_DVR_IPC_PASSWD	设置 IPC 用户名密码	6008
NET_DVR_SET_DVR_IPC_NET	设置前端 IPC 的网络地址	6012
NET_DVR_SET_DISK_RAID_INFO	设置磁盘 Raid 信息	6002
NET_DVR_SET_ACCESS_CAMERA_INFO	设置前端相机信息	6202
NET_DVR_SET_DRAWFRAME_DISK_QUOTA_CFG	设置抽帧通道磁盘配额	6110
NET_DVR_SET_NAT_CFG	设置 NAT 映射参数	6112
NET_DVR_SET_POE_CFG	设置 POE 参数	6115
NET_DVR_SET_CUSTOM_PRO_CFG	设置自定义协议参数	6117
NET_DVR_SET_STREAM_CABAC	设置码流压缩性能选项	6119
NET_DVR_SET_ESATA_MINISAS_USAGE_CFG	设置 eSATA 和 miniSAS 用途	6121
NET_DVR_SET_POE_CHANNEL_ADD_MODE	设置 POE 通道添加方式	6125
NET_DVR_SET_WD1_CFG	设置 WD1 使能开关	6137
NET_DVR_SET_N_PLUS_ONE_WORK_MODE	设置 N+1 工作模式	6169
NET_DVR_SET_HD_STATUS	设置硬盘状态	6171
NET_DVR_SET_RAID_BACKGROUND_TASK_SPEED	设置 RAID 后台任务速度	6176
NET_DVR_SET_VIDEO_AUDIOIN_CFG	设置视频的音频输入参数	9119
NET_DVR_SET_CMS_CFG	设置平台参数	2071
NET_DVR_SET_DVR_SYNCHRONOUS_IPC	设置是否为前端 IPC 同步设备参数	6006
NET_DVR_SET_DVR_IPC_PASSWD	设置 IPC 用户名密码	6008

NET_DVR_SET_DVR_IPC_NET	设置前端 IPC 的网络地址	6012
NET_DVR_SET_BONJOUR_CFG	设置 Bonjour 信息	6128
NET_DVR_SET_SOCKS_CFG	设置 SOCKS	6131
NET_DVR_SET_QOS_CFG	设置 QoS 信息	6133
NET_DVR_SET_HTTPS_CFG	设置 HTTPS 信息	6135
NET_DVR_SET_RECORD_PACK	设置录像打包参数	6302
NET_DVR_SET_CLOUD_STORAGE_CFG	设置云存储工作模式	6304

表 5.36 设置设备参数

dwCommand 宏定义	IChannel 取值	IpInBuffer 对应结构体
NET_DVR_SET_DEVICECFG_V40	无效	NET_DVR_DEVICECFG_V40
NET_DVR_SET_TIMECFG	无效	NET_DVR_TIME
NET_DVR_SET_NETCFG_V30	无效	NET_DVR_NETCFG_V30
NET_DVR_SET_PICCFG_V40	通道号	NET_DVR_PICCFG_V40
NET_DVR_SET_COMPRESSCFG_V30	通道号	NET_DVR_COMPRESSIONCFG_V30
NET_DVR_SET_RECORDCFG_V30	通道号	NET_DVR_RECORD_V30
NET_DVR_SET_DECODERCFG_V30	通道号	NET_DVR_DECODERCFG_V30
NET_DVR_SET_RS232CFG_V30	无效	NET_DVR_RS232CFG_V30
NET_DVR_SET_ALARMINGCFG_V40	报警输入号，从 0 开始	NET_DVR_ALARMINGCFG_V40
NET_DVR_SET_ALARMOUTCFG_V30	报警输出号，从 0 开始	NET_DVR_ALARMOUTCFG_V30
NET_DVR_SET_PREVIEWCFG_V30	无效	NET_DVR_PREVIEWCFG_V30
NET_DVR_SET_PREVIEWCFG_AUX_V30	无效	NET_DVR_PREVIEWCFG_V30
NET_DVR_SET_VIDEOOUTCFG_V30	无效	NET_DVR_VIDEOOUT_V30
NET_DVR_SET_USERCFG_V40	无效	NET_DVR_USER_V40
NET_DVR_SET_EXCEPTIONCFG_V40	无效	NET_DVR_EXCEPTION_V40
NET_DVR_SET_ZONEANDDST	无效	NET_DVR_ZONEANDDST
NET_DVR_SET_SHOWSTRING_V30	通道号	NET_DVR_SHOWSTRING_V30
NET_DVR_SET_AUXOUTCFG	无效	NET_DVR_AUXOUTCFG
NET_DVR_SET_NETAPPCFG	无效	NET_DVR_NETAPPCFG
NET_DVR_SET_NTPCFG	无效	NET_DVR_NTPPARA
NET_DVR_SET_DDNSCFG_V30	无效	NET_DVR_DDNSPARA_V30
NET_DVR_SET_EMAILCFG_V30	无效	NET_DVR_EMAILCFG_V30
NET_DVR_SET_NFSCFG	无效	NET_DVR_NFSCFG
NET_DVR_SET_IPPARACFG_V40	无效	NET_DVR_IPPARACFG_V40
NET_DVR_SET_HDCFG	无效	NET_DVR_HDCFG

NET_DVR_SET_HDCFG_V40	组号，从 0 开始，每组 33 个硬盘	NET_DVR_HDCFG
NET_DVR_SET_HDGROUP_CFG_V40	组号，从 0 开始，每组 16 个硬盘组	NET_DVR_HDGROUP_CFG_V40
NET_DVR_SET_COMPRESSCFG_AUD	无效	NET_DVR_COMPRESSION_AUDIO
NET_DVR_SET_NET_DISKCFG	无效	NET_DVR_NET_DISKCFG
NET_DVR_SET_SNMPCFG	无效	NET_DVR_SNMPCFG
NET_DVR_SET_NETCFG_MULTI	无效	NET_DVR_NETCFG_MULTI
NET_DVR_SET_NETWORK_BONDING	无效	NET_DVR_NETWORK_BONDING
NET_DVR_SET_HOLIDAY_PARAM_CFG	无效	NET_DVR_HOLIDAY_PARAM_CFG
NET_DVR_SET_MOTION_HOLIDAY_HANDLE	通道号	NET_DVR_HOLIDAY_HANDLE
NET_DVR_SET_VIOLST_HOLIDAY_HANDLE	通道号	NET_DVR_HOLIDAY_HANDLE
NET_DVR_SET_HIDE_HOLIDAY_HANDLE	通道号	NET_DVR_HOLIDAY_HANDLE
NET_DVR_SET_ALARMIN_HOLIDAY_HANDLE	报警输入号，从 0 开始	NET_DVR_HOLIDAY_HANDLE
NET_DVR_SET_ALARMOUT_HOLIDAY_HANDLE	报警输出号，从 0 开始	NET_DVR_HOLIDAY_HANDLE
NET_DVR_SET_HOLIDAY_RECORD	通道号	NET_DVR_HOLIDAY_RECORD
NET_DVR_SET_DISK_QUOTA_CFG	通道号	NET_DVR_DISK_QUOTA_CFG
NET_DVR_SET_JPEG_CAPTURE_CFG	通道号	NET_DVR_JPEG_CAPTURE_CFG
NET_DVR_SET_SCHED_CAPTURECFG	通道号	NET_DVR_SCHED_CAPTURECFG
NET_DVR_SET_FTPCFG	无效	NET_DVR_FTPCFG
NET_DVR_SET_FTPCFG_SECOND	无效	NET_DVR_FTPCFG
NET_DVR_SET_VIDEO_INPUT_EFFECT	通道号	NET_DVR_VIDEO_INPUT_EFFECT
NET_DVR_SET_DVR_SYNCHRONOUS_IPC	无效	NET_DVR_SYNCHRONOUS_IPC
NET_DVR_SET_DVR_IPC_PASSWD	通道号	NET_DVR_IPC_PASSWD
NET_DVR_SET_DVR_IPC_NET	通道号	NET_DVR_IPC_NETCFG
NET_DVR_SET_DISK_RAID_INFO	无效	NET_DVR_DISK_RAID_INFO
NET_DVR_SET_ACCESS_CAMERA_INFO	通道号	NET_DVR_ACCESS_CAMERA_INFO
NET_DVR_SET_DRAWFRAME_DISK_QUOTA_CFG	无效	NET_DVR_DRAWFRAME_DISK_QUOTA_CFG
NET_DVR_SET_NAT_CFG	无效	NET_DVR_NAT_CFG
NET_DVR_SET_POE_CFG	无效	NET_DVR_POE_CFG
NET_DVR_SET_CUSTOM_PRO_CFG	协议序号，从 1 开始	NET_DVR_CUSTOM_PROTOCOL
NET_DVR_SET_STREAM_CABAC	通道号	NET_DVR_STREAM_CABAC
NET_DVR_SET_ESATA_MINISAS_USAGE_CFG	无效	NET_DVR_ESATA_MINISAS_USAGE
NET_DVR_SET_POE_CHANNEL_ADD_MODE	通道号	NET_DVR_POE_CHANNEL_ADD_MODE
NET_DVR_SET_WD1_CFG	通道号	NET_DVR_WD1_CFG

NET_DVR_SET_N_PLUS_ONE_WORK_MODE	无效	NET_DVR_N_PLUS_ONE_WORK_MODE_CFG
NET_DVR_SET_HD_STATUS	无效	NET_DVR_HD_STATUS
NET_DVR_SET_RAID_BACKGROUND_TASK_SPEED	无效	NET_DVR_RAID_BTS_CFG
NET_DVR_SET_VIDEO_AUDIOIN_CFG	语音对讲通道号, 从 0 开始	NET_DVR_VIDEO_AUDIOIN_CFG
NET_DVR_SET_CMS_CFG	无效	NET_DVR_CMS_PARAM
NET_DVR_SET_DVR_SYNCHRONOUS_IPC	通道号	NET_DVR_SYNCHRONOUS_IPC
NET_DVR_SET_DVR_IPC_PASSWD	通道号	NET_DVR_IPC_PASSWD
NET_DVR_SET_DVR_IPC_NET	通道号	NET_DVR_IPC_NETCFG
NET_DVR_SET_BONJOUR_CFG	无效	NET_DVR_BONJOUR_CFG
NET_DVR_SET_SOCKS_CFG	无效	NET_DVR_SOCKS_CFG
NET_DVR_SET_QOS_CFG	无效	NET_DVR_QOS_CFG
NET_DVR_SET_HTTPS_CFG	无效	NET_DVR_HTTPS_CFG
NET_DVR_SET_RECORD_PACK	无效	NET_DVR_RECORD_PACK
NET_DVR_SET_CLOUD_STORAGE_CFG	无效	NET_DVR_CLOUD_STORAGE_CFG

[返回目录](#)

5.19.3 获取设备的配置信息(标准协议)NET_DVR_GetSTDConfig

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

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 5.37
[in&out]lpConfigParam 配置输入输出参数, 不同的配置功能对应不同的输入输出参数, 详见表 5.37

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

说 明: 获取配置参数时, lpConfigParam 结构体中的 lpInBuffer 无效, 设为 NULL。对于不同的配置功能 (dwCommand), lpConfigParam 中 lpCondBuffer、lpOutBuffer 分别对应不同的内容, 具体如表 5.37 所示。

表 5.37 获取设备参数

dwCommand 宏定义	含义	lpCondBuffer	lpOutBuffer	宏定义值
NET_DVR_GET_GBT28181_SERVICE_CFG	获取 GB28181 服务器参数	NULL	NET_DVR_GB28181_SERVICE_CFG	6503

[返回目录](#)

5.19.4 设置设备的配置信息(标准协议)NET_DVR_SetSTDConfig

函 数: BOOL NET_DVR_SetSTDConfig(LONG lUserID, 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 5.38
[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 5.38
- 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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 5.38 所示。

表 5.38 获取设备参数

dwCommand 宏定义	含义	IpCondBuffer	IpOutBuffer	宏定义值
NET_DVR_SET_IPDEVICE_ACTIVATED	通过 NVR 激活前端设备	NULL	NET_DVR_IPDEVICE_ACTIVATE_CFG	13000
NET_DVR_SET_GBT28181_SERVICE_CFG	设置 GB28181 服务器参数	NULL	NET_DVR_GB28181_SERVICE_CFG	6504

[返回目录](#)

批量参数配置

5.19.5 批量获取配置信息 NET_DVR_GetDeviceConfig

- 函 数: BOOL NET_DVR_GetDeviceConfig(LONG IUserID, DWORD dwCommand, DWORD dwCount, LPVOID IpInBuffer, DWORD dwInBufferSize, LPVOID IpStatusList, LPVOID IpOutBuffer, 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 5.39
[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] IpInBuffer 配置条件缓冲区, 详见表 5.40
[in] dwInBufferSize 缓冲区长度
[out] IpStatusList 错误信息列表, 和要查询的配置一一对应, 例如 IpStatusList[2]就对应 IpInBuffer[2], 由用户分配内存, 每个错误信息为 4 个字节, 参数值: 0 或者 1 表示成功, 其他值为失败对应的错误号
[out] IpOutBuffer 设备返回的参数内容 (详见表 5.40), 和要查询的监控点一一对应。如果某个监控点对应的 IpStatusList 信息为大于 0 值, 对应 IpOutBuffer 的内容就是无效的
[in] dwOutBufferSize 输出缓冲区大小
- 返回值: TRUE 表示成功, 但不代表每一个配置都成功, 哪一个成功, 对应查看 IpStatusList[n]值; FALSE 表示全部失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
- 说 明: 该接口是带有发送数据的批量获取配置信息的通用接口。IpInBuffer 指定需要获取的信息, IpOutBuffer 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 5.40 所示。

表 5.39 参数批量获取命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_MULTI_STREAM_COMPRESSIONCFG	获取多码流压缩参数	3216
NET_DVR_GET_DEFAULT_VIDEO_EFFECT	获取默认视频效果	6136
NET_DVR_GET_CERT	获取证书信息	6142
NET_DVR_GET_FTPCFG_V40	获取 FTP 信息	6162
NET_DVR_GET_ACCESS_DEVICE_CHANNEL_INFO	获取待接入设备通道信息，dwCount 为 0	6165
NET_DVR_GET_PREVIEW_SWITCH_CFG	获取设备本地预览切换参数，dwCount 为 0	6166
NET_DVR_GET_WORK_STATUS	获取设备工作状态，dwCount 为 0	6189
NET_DVR_GET_JPEG_CAPTURE_CFG_V40	获取设备抓图配置，dwCount 为 0	6190
NET_DVR_GET_USER_RIGHT_CFG	获取用户权限，dwCount 为 1	6210
NET_DVR_GET_STREAM_RECORD_STATUS	获取流录像状态信	6021

表 5.40 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
NET_DVR_GET_DEFAULT_VIDEO_EFFECT	dwCount 个 NET_DVR_DEFAULT_VIDEO_COND	dwCount 个 NET_DVR_VIDEO_EFFECT
NET_DVR_GET_CERT	dwCount 个 NET_DVR_CERT_PARAM	dwCount 个 NET_DVR_CERT_INFO
NET_DVR_GET_FTPCFG_V40	dwCount 个 NET_DVR_FTP_TYPE	dwCount 个 NET_DVR_FTPCFG_V40
NET_DVR_GET_ACCESS_DEVICE_CHANNEL_INFO	NET_DVR_ACCESS_DEVICE_INFO	NET_DVR_ACCESS_DEVICE_CHANNEL_INFO
NET_DVR_GET_PREVIEW_SWITCH_CFG	NET_DVR_PREVIEW_SWITCH_COND	NET_DVR_PREVIEW_SWITCH_CFG
NET_DVR_GET_WORK_STATUS	NET_DVR_GETWORKSTATE_COND	NET_DVR_WORKSTATE_V40
NET_DVR_GET_JPEG_CAPTURE_CFG_V40	NET_DVR_CHANNEL_GROUP	NET_DVR_JPEG_CAPTURE_CFG_V40
NET_DVR_GET_USER_RIGHT_CFG	NET_DVR_USER_COND	NET_DVR_USER_RIGHT_CFG
NET_DVR_GET_STREAM_RECORD_STATUS	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_STREAM_RECORD_STATUS

[返回目录](#)

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

函 数： BOOL NET_DVR_SetDeviceConfig(LONG IUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpInParamBuffer, DWORD dwInParamBufferSize)

参 数： [in] IUserID 用户 ID 号，NET_DVR_Login_V40 的返回值
[in] dwCommand 设备配置命令，详见表 5.41
[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 5.42

[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 5.42），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。不同的设置功能对应不同的结构体和命令号，如表 5.42 所示。

表 5.41 参数批量设置命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_SET_MULTI_STREAM_COMPRESSIONCFG	设置多码流压缩参数	3217
NET_DVR_SET_FTPCFG_V40	设置 FTP 信息	6163
NET_DVR_SET_PREVIEW_SWITCH_CFG	设置设备本地预览切换参数，dwCount 为 0	6167
NET_DVR_SET_JPEG_CAPTURE_CFG_V40	设置设备抓图配置，dwCount 为 0	6191
NET_DVR_SET_USER_RIGHT_CFG	设置用户权限，dwCount 为 1	6211
NET_DVR_SET_FTPCFG_V40	设置 FTP 信息	6163
NET_DVR_SET_STREAM_RECORD_STATUS	设置流录像状态信息	6022

表 5.42 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
NET_DVR_SET_FTPCFG_V40	dwCount 个 NET_DVR_FTP_TYPE	dwCount 个 NET_DVR_FTPCFG_V40
NET_DVR_SET_PREVIEW_SWITCH_CFG	NET_DVR_PREVIEW_SWITCH_COND	NET_DVR_PREVIEW_SWITCH_CFG
NET_DVR_SET_JPEG_CAPTURE_CFG_V40	NET_DVR_CHANNEL_GROUP	NET_DVR_JPEG_CAPTURE_CFG_V40
NET_DVR_SET_USER_RIGHT_CFG	NET_DVR_USER_COND	NET_DVR_USER_RIGHT_CFG
NET_DVR_SET_STREAM_RECORD_STATUS	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_STREAM_RECORD_STATUS

[返回目录](#)

长连接参数配置

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

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

参 数: [in] lUserID 用户 ID 号, NET_DVR_Login_V30 返回值
 [in] dwCommand 配置命令, 详见表 5.43
 [in] lpInBuffer 输入参数, 具体内容跟配置命令相关, 详见表 5.43
 [in] dwInBufferLen 输入缓冲的大小
 [in] cbStateCallback 状态回调函数
 [in] pUserData 用户数据

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

[out] dwType 配置状态

```
enum _NET_SDK_CALLBACK_STATUS_NORMAL_{
    NET_SDK_CALLBACK_STATUS_SUCCESS = 1000, //成功
    NET_SDK_CALLBACK_STATUS_PROCESSING, //处理中
    NET_SDK_CALLBACK_STATUS_FAILED, //失败
    NET_SDK_CALLBACK_STATUS_EXCEPTION, //异常
    NET_SDK_CALLBACK_STATUS_LANGUAGE_MISMATCH,
    // (IPC 配置文件导入) 语言不匹配
    NET_SDK_CALLBACK_STATUS_DEV_TYPE_MISMATCH
    // (IPC 配置文件导入) 设备类型不匹配
}NET_SDK_CALLBACK_STATUS_NORMAL
```

[out] lpBuffer dwType 状态为失败时的错误信息, 不同的配置命令对应不同的结构体, 详见表 5.44

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

说 明: 调用该接口实现长连接远程配置, 如果 dwCommand 对应参数获取或者信息查询功能, 该接口调用成功后, 还需要调用 NET_DVR_GetNextRemoteConfig 接口来逐个获取对应内容。不同的控制功能对应不同的命令号, 同时 lpInBuffer 对应不同的结构体, 如表 5.43 所示。

表 5.43 长连接参数配置

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer 对应结构	cbStateCallback
NET_DVR_FIND_NAS_DIRECTORY	6161	查找 NAS 目录	NET_DVR_NET_DISK_SERACH_PARAM	NULL, 获取状态请调用 NET_DVR_GetRemoteConfigState
NET_DVR_IMPORT_IPC_CFG_FILE	6172	导入 IPC 配置文件	NET_DVR_IPC_CFG_FILE_PARAM	返回成功或失败相应的错误信息
NET_DVR_UPGRADE_IPC	6174	升级 IP 通道	NET_DVR_UPGRADE_IPC_PARAM	返回成功或失败相应的错误信息
NET_DVR_GET_LOCKED_INFO_LIST	149	获取所有被锁定信息	NULL	NULL

表 5.44 长连接回调数据

dwCommand 宏定义	回调 fRemoteConfigCallback 中 lpBuffer 对应结构
NET_DVR_IMPORT_IPC_CFG_FILE	NET_DVR_IPC_CFG_FILE_ERR_INFO
NET_DVR_UPGRADE_IPC	NET_DVR_UPGRADE_IPC_ERR_INFO

[返回目录](#)

5.19.8 逐个获取查找到的信息 NET_DVR_GetNextRemoteConfig

函 数: LONG NET_DVR_GetNextRemoteConfig(LONG lHandle, void *lpOutBuff, DWORD dwOutBuffSize)

参 数: [in] lHandle 查找句柄, NET_DVR_StartRemoteConfig 的返回值
 [out] lpOutBuff 输出数据缓冲区, 与 NET_DVR_StartRemoteConfig 的命令 (dwCommand) 有关, 详见表 5.46
 [out] dwOutBuffSize 缓冲区长度

返回值: -1 表示失败, 其他值表示当前的获取状态等信息, 详见表 5.45。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

表 5.45 长连接参数获取状态

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

说 明: 调用 NET_DVR_StartRemoteConfig 时传入不同的命令号(dwCommand), lpOutBuff 对应不同的结构体, 如表 5.46 所示。

在调用该接口获取查找之前, 必须先调用 NET_DVR_StartRemoteConfig 得到当前的查找句柄。此接口用于获取一条已查找到的信息, 若要获取全部的已查找到的信息, 需要循环调用此接口。

表 5.46 长连接参数获取

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer
NET_DVR_FIND_NAS_DIRECTORY	6161	查找 NAS 目录	NET_DVR_NET_DISK_SERACH_RET
NET_DVR_GET_LOCKED_INFO_LIST	149	获取所有被锁定信息	NET_DVR_LOCKED_INFO

[返回目录](#)

5.19.9 获取长连接配置的状态 NET_DVR_GetRemoteConfigState

函 数: BOOL NET_DVR_GetRemoteConfigState(LONG lHandle, void *pState)

参 数: [in] lHandle 句柄, NET_DVR_StartRemoteConfig 的返回值

[out] pState 返回的状态值，不同的配置命令对应不同的状态取值，详见表 5.47

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

说明: 调用 NET_DVR_StartRemoteConfig 时传入不同的命令号(dwCommand)，pState 对应不同的取值，如表 5.47 所示。

表 5.47 长连接配置状态

dwCommand 宏定义	宏定义值	含义
NET_DVR_FIND_NAS_DIRECTORY	6161	查找 NAS 目录
pState 状态宏定义	宏定义值	含义
NET_SDK_CALLBACK_STATUS_SUCCESS	1000	查找成功
NET_SDK_CALLBACK_STATUS_PROCESSING	1001	处理中
NET_SDK_CALLBACK_STATUS_FAILED	1002	查找失败

[返回目录](#)

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

说明:

[返回目录](#)

远程控制

5.19.11 远程控制 NET_DVR_RemoteControl

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

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

[in] dwCommand 控制命令，详见表 5.48

[in] lpInBuffer 输入参数，具体内容跟控制命令相关，详见表 5.48

[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，详见表 5.48。

表 5.48 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_MOUNT_DISK	6015	加载磁盘	NET_DVR_MOUNT_DISK_PARAM
NET_DVR_UNMOUNT_DISK	6016	卸载磁盘	NET_DVR_MOUNT_DISK_PARAM
NET_DVR_SET_VIDEO_EFFECT	6106	设置通道视频输入图像参数	NET_DVR_VIDEOPARA_V40
NET_DVR_DEL_INVALID_DISK	6107	删除无效磁盘	NET_DVR_INVALID_DISK_PARAM
NET_DVR_CMD_TRIGGER_PERIOD_RECORD	6144	外部命令触发录像	NET_DVR_CMD_TRIGGER_PERIOD_RECORD_PARA
NET_DVR_UNLOCK_USER	146	用户解锁	NET_DVR_UNLOCK_INFO

[返回目录](#)

5.19.12 远程控制(标准协议)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 5.49
[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 5.49

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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 5.49 所示。

表 5.49 远程控制命令

dwCommand 宏定义	宏定义值	含义	lpCondBuffer 对应结构体
NET_DVR_SYNC_IPC_PASSWD	6621	同步 IPC 密码与 NVR 一致	NULL

- 设备是否支持同步 IPC 密码功能, 对应设备软硬件能力集(BasicCapability)中节点<isSupportSyncIPPassword>, 接口: [NET_DVR_GetDeviceAbility](#)(能力集类型: DEVICE_SOFTWARE_ABILITY)。

[返回目录](#)

报警输出配置

5.19.13 获取设备报警输出 NET_DVR_GetAlarmOut_V30

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

参 数: [in]IUserID 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](#)

5.19.14 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` 报警输出口。初始输出口从 0 开始, `0x00ff` 表示全部模拟输出, `0xff00` 表示全部数字输出。DS-90xx 系列等设备支持对 IP 接入的报警输出进行处理, 对应 32-95 为数字报警输出
- `[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

5.19.15 获取 RTSP 协议参数 **NET_DVR_GetRtspConfig**

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

参 数:

- `[in]lUserID` 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](#)

5.19.16 设置 RTSP 协议参数 **NET_DVR_SetRtspConfig**

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

参 数:

- `[in]lUserID` 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](#)

获取设备支持的云台协议

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

函 数: BOOL NET_DVR_GetPTZProtocol(LONG IUserID, NET_DVR_PTZCFG *pPtzcfg)
参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[out]pPtzcfg 设备的云台协议结构
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
说 明: 在早前的设备中规定了一系列云台协议, 但在后期的设备 (如 DS-90xx、DS-91xx、DS-81xx 等) 仅保留一部分常用的协议, 所以在配置前端云台协议时必须调用该接口获取当前设备支持的云台协议。

[返回目录](#)

视频输出缩放参数配置

5.19.18 获取视频输出缩放信息 **NET_DVR_GetScaleCFG_V30**

函 数: BOOL NET_DVR_GetScaleCFG(LONG IUserID, LPNET_DVR_SCALECFG pScalecfg)
参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[out]pScalecfg 缩放参数信息
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
说 明:

[返回目录](#)

5.19.19 设置视频输出缩放参数 **NET_DVR_SetScaleCFG_V30**

函 数: BOOL NET_DVR_SetScaleCFG_V30(LONG IUserID, LPNET_DVR_SCALECFG pScalecfg)
参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]pScalecfg 缩放参数配置结构体
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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 SMART 参数配置

SMART 参数配置

5.20.1 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 5.50
 - [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 5.51
 - [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 5.51），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 5.51 所示。

表 5.50 参数批量获取命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_VCA_DETION_CFG	获取智能移动参数配置	5041
NET_DVR_GET_GUARDCFG	获取车牌识别检测计划	3134
NET_DVR_GET_TRAVERSE_PLANE_DETECTION	获取越界侦测配置	3360
NET_DVR_GET_FIELD_DETECTION	获取区域侦测配置	3362
NET_DVR_GET_SMD_HOLIDAY_HANDLE	获取简易智能假日计划	6194

表 5.51 批量获取设备参数

dwCommand	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_GET_VCA_DETION_CFG	dwCount 个 NET_DVR_CHANNEL_GROUP	dwCount 个 NET_DVR_VCA_DETION_CFG
NET_DVR_GET_GUARDCFG	dwCount 个 NET_DVR_GUARD_COND	dwCount 个 NET_DVR_GUARD_CFG

NET_DVR_GET_SCENECCHANGE_DETECTIONCFG	dwCount 个 NET_DVR_CHANNEL_GROUP	dwCount 个 NET_VCA_TRAVERSE_PLANE_DETECTION
NET_DVR_GET_TRAVERSE_PLANE_DETECTION	dwCount 个 NET_DVR_CHANNEL_GROUP	dwCount 个 NET_VCA_FIELDDETECCION
NET_DVR_GET_SMD_HOLIDAY_HANDLE	dwCount 个 NET_DVR_HOLIDAY_HANDLE_COND	dwCount 个 NET_DVR_HOLIDAY_HANDLE

[返回目录](#)

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

- 函 数：** BOOL NET_DVR_SetDeviceConfig(LONG IUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpInParamBuffer, DWORD dwInParamBufferSize)
- 参 数：**
- [in] IUserID 用户 ID 号，NET_DVR_Login_V40 的返回值
 - [in] dwCommand 设备配置命令，详见表 5.52
 - [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 5.53
 - [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 5.53），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。不同的设置功能对应不同的结构体和命令号，如表 5.53 所示。

表 5.52 参数批量设置命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_SET_VCA_DETION_CFG	设置智能移动参数配置	5040
NET_DVR_SET_GUARDCFG	设置车牌识别检测计划	3135
NET_DVR_SET_TRAVERSE_PLANE_DETECTION	设置越界侦测配置	3361
NET_DVR_SET_FIELD_DETECTION	设置区域侦测配置	3363
NET_DVR_SET_SMD_HOLIDAY_HANDLE	设置简易智能假日计划	6195

表 5.53 批量设置设备参数

dwCommand	lpInBuffer 对应结构体	lpInParamBuffer 对应结构体
NET_DVR_SET_VCA_DETION_CFG	dwCount 个 NET_DVR_CHANNEL_GROUP	dwCount 个 NET_DVR_VCA_DETION_CFG
NET_DVR_SET_GUARDCFG	dwCount 个 NET_DVR_GUARD_COND	dwCount 个 NET_DVR_GUARD_CFG

NET_DVR_SET_SCENECHANGE_DETECTIONCFG	dwCount 个 NET_DVR_CHANNEL_GROUP	dwCount 个 NET_VCA_TRAVERSE_PLANE_DETECTION
NET_DVR_SET_TRAVERSE_PLANE_DETECTION	dwCount 个 NET_DVR_CHANNEL_GROUP	dwCount 个 NET_VCA_FIELDDETECTION
NET_DVR_SET_SMD_HOLIDAY_HANDLE	dwCount 个 NET_DVR_HOLIDAY_HANDLE_COND	dwCount 个 NET_DVR_HOLIDAY_HANDLE

[返回目录](#)

VQD 视频质量诊断

5.20.3 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 5.54
 - [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 5.55
 - [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 5.55），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 5.55 所示。

表 5.54 VQD 参数获取命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_VQD_LOOP_DIAGNOSE_CFG	获取 VQD 循环诊断配置参数	6406

表 5.55 VQD 参数获取结构

dwCommand	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_GET_VQD_LOOP_DIAGNOSE_CFG	dwCount 个 NET_DVR_CHANNEL_GROUP	dwCount 个 NET_DVR_VQD_LOOP_DIAGNOSE_CFG

[返回目录](#)

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

- 函 数：** BOOL NET_DVR_SetDeviceConfig(LONG IUserID,DWORD dwCommand,DWORD dwCount,LPVOID lpInBuffer,DWORD dwInBufferSize,LPVOID lpStatusList, LPVOID lpInParamBuffer, DWORD dwInParamBufferSize)
- 参 数：**
- [in] IUserID 用户 ID 号，NET_DVR_Login_V40 的返回值
 - [in] dwCommand 设备配置命令，参见表 5.56
 - [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 5.57
 - [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 5.57），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 5.57 所示。

表 5.56 VQD 参数设置命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_SET_VQD_LOOP_DIAGNOSE_CFG	设置 VQD 循环诊断配置参数	6407

表 5.57 VQD 参数设置结构

dwCommand	lpInBuffer 对应结构体	lpInParamBuffer 对应结构体
NET_DVR_SET_VQD_LOOP_DIAGNOSE_CFG	dwCount 个 NET_DVR_CHANNEL_GROUP	dwCount 个 NET_DVR_VQD_LOOP_DIAGNOSE_CFG

[返回目录](#)

5.20.5 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 pData)
- 参 数：**
- [in] IUserID 用户 ID 号，NET_DVR_Login_V30 返回值
 - [in] dwCommand 配置命令，详见表 5.58
 - [in] lpInBuffer 输入参数，具体内容跟配置命令相关，详见表 5.58
 - [in] dwInBufferLen 输入缓冲的大小
 - [in] cbStateCallback 状态回调函数

[in] pUserData 用户数据

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

[out] dwType 配置状态

[out] lpBuffer 存放数据的缓冲区指针，具体内容跟 dwType 相关，详见表 5.59

[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: 接口中不同的控制功能对应不同的命令号(dwCommand)，同时 lpInBuffer 对应不同的结构体，如表 5.59 所示。

表 5.58 VQD 长连接配置

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer 对应结构体
NET_DVR_GET_VQD_DIAGNOSE_INFO	6408	手动获取 VQD 诊断信息	通道号(DWORD 类型)数组

表 5.59 长连接回调数据

dwType	值	含义	lpBuffer 对应内容
NET_SDK_CALLBACK_TYPE_STATUS	0	状态值	typedef enum { NET_SDK_CALLBACK_STATUS_SUCCESS = 1000, //成功 NET_SDK_CALLBACK_STATUS_PROCESSING, //处理中 NET_SDK_CALLBACK_STATUS_FAILED //失败 }NET_SDK_CALLBACK_STATUS_NORMAL;
NET_SDK_CALLBACK_TYPE_PROGRESS	1	进度值	lpBuffer 的值表示进度
NET_SDK_CALLBACK_TYPE_DATA	2	信息数据	lpBuffer 的值表示信息数据

调用该接口实现长连接远程配置，如果 dwCommand 对应参数获取或者信息查询功能，该接口调用成功后，还需要调用 NET_DVR_GetNextRemoteConfig 接口来逐个获取对应内容。

[返回目录](#)

5.20.6 逐个获取查找到的信息 [NET_DVR_GetNextRemoteConfig](#)

函 数: LONG NET_DVR_GetNextRemoteConfig(LONG lHandle, void *lpOutBuff, DWORD dwOutBuffSize)

参 数: [in] lHandle 查找句柄，NET_DVR_StartRemoteConfig 的返回值

[out] lpOutBuff 输出数据缓冲区，与 NET_DVR_StartRemoteConfig 的命令（dwCommand）有关，详见表 5.61

[out] dwOutBuffSize 缓冲区长度

返回值: -1 表示失败，其他值表示当前的获取状态等信息，详见表 5.60。获取错误码调用 [NET_DVR_GetLastError](#)。

表 5.60 长连接参数获取状态

宏定义	宏定义值	含义
NET_SDK_GET_NEXT_STATUS_SUCCESS	1000	成功读取到数据，处理完本次数据后需要再次调用 NET_DVR_GetNextRemoteConfig 获取下一条数据
NET_SDK_GET_NETX_STATUS_NEED_WAIT	1001	需等待设备发送数据，继续调用

		NET_DVR_GetNextRemoteConfig
NET_SDK_GET_NEXT_STATUS_FINISH	1002	数据全部取完, 可调用 NET_DVR_StopRemoteConfig 结束长连接
NET_SDK_GET_NEXT_STATUS_FAILED	1003	出现异常, 可调用 NET_DVR_StopRemoteConfig 结束长连接

说明: 调用 NET_DVR_StartRemoteConfig 时传入不同的命令号(dwCommand), lpOutBuff 对应不同的结构体, 如表 5.61 所示。在调用该接口获取查找之前, 必须先调用 NET_DVR_StartRemoteConfig 得到当前的查找句柄。此接口用于获取一条已查找到的信息, 若要获取全部的已查找到的信息, 需要循环调用此接口。

表 5.61 长连接参数获取

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer
NET_DVR_GET_VQD_DIAGNOSE_INFO	6408	手动获取 VQD 诊断信息	NET_DVR_VQD_DIAGNOSE_INFO

[返回目录](#)

5.20.7 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 lHandle)
参数: [in] lHandle 句柄, NET_DVR_StartRemoteConfig 的返回值
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说明:

[返回目录](#)

5.21 硬盘管理

5.21.1 远程格式化设备硬盘 NET_DVR_FormatDisk

函数: LONG NET_DVR_FormatDisk(LONG lUserID, LONG lDiskNumber)
参数: [in] lUserID NET_DVR_Login_V40 的返回值
[in] lDiskNumber 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](#)

5.21.2 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](#)

5.21.3 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](#)

5.22 零通道预览和配置

参数配置

5.22.1 获取设备的配置信息 NET_DVR_GetDVRConfig

函 数: BOOL NET_DVR_GetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpOutBuffer, DWORD dwOutBufferSize, LPDWORD lpBytesReturned)
 参 数: [in]IUserID NET_DVR_Login_V40 的返回值
 [in]dwCommand 设备配置命令, 详见表 5.62
 [in]IChannel 设备只有一个零通道, 其中通道号设为 1
 [out]lpOutBuffer 接收数据的缓冲指针, 详见表 5.62
 [in]dwOutBufferSize 接收数据的缓冲长度(单位: 字节), 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 5.62 所示。

表 5.62 零通道相关参数获取

dwCommand 宏定义	dwCommand 含义	IChannel 取值	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_ZEROCHANCFG	获取零通道压缩参数	通道号	NET_DVR_ZEROCHANCFG	1102
NET_DVR_GET_ZERO_PREVIEWCFG_V30	获取零通道本地预览参数	通道号	NET_DVR_PREVIEWCFG_V30	1104
NET_DVR_GET_ZERO_ZOOM	获取零通道缩放参数	通道号	NET_DVR_ZERO_ZOOMCFG	1107

[返回目录](#)

5.22.2 设置设备的配置信息 **NET_DVR_SetDVRConfig**

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

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]dwCommand 设备配置命令, 详见表 5.63
[in]IChannel 设备只有一个零通道, 其中通道号设为 1
[in]lpInBuffer 输入数据的缓冲指针, 详见表 5.63
[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 5.63 所示。

表 5.63 零通道相关参数设置

dwCommand 宏定义	dwCommand 含义	通道号	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_ZEROCHANCFG	设置零通道压缩参数	通道号	NET_DVR_ZEROCHANCFG	1103
NET_DVR_SET_ZERO_PREVIEWCFG_V30	设置零通道本地预览参数	通道号	NET_DVR_PREVIEWCFG_V30	1105
NET_DVR_SET_ZERO_ZOOM	设置零通道缩放参数	通道号	NET_DVR_ZERO_ZOOMCFG	1106

[返回目录](#)

实时预览

5.22.3 开启零通道预览 **NET_DVR_ZeroStartPlay**

函 数: LONG NET_DVR_ZeroStartPlay(LONG IUserID, LPNET_DVR_CLIENTINFO lpClientInfo, fRealDataCallBack_V30 cbRealDataCallBack, void* pUser, BOOL bBlocked)

参 数: [in]IRealHandle NET_DVR_RealPlay_V40 的返回值
[in]lpClientInfo 预览参数 (设备只有一个零通道, 其中通道号设为 1)
[in]cbRealDataCallBack 码流数据回调函数
[in]pUser 用户数据
[in]bBlocked 请求码流过程是否阻塞: 0- 否, 1- 是

```
typedef void(CALLBACK *fRealDataCallBack_V30)(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 5.64
[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 5.64 码流数据类型

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

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

说明：该接口中可以设置当前预览操作是否阻塞（通过 bBlocked 参数设置）。若设为不阻塞，表示发起与设备的连接就认为连接成功，如果发生码流接收失败、播放失败等情况以预览异常的方式通知上层。若设为阻塞，表示直到播放操作完成才返回成功与否。
该接口中的回调函数可以置为空，此时该函数将不回调码流数据给用户，但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。

[返回目录](#)

5.22.4 停止预览 **NET_DVR_ZeroStopPlay**

函数：BOOL NET_DVR_ZeroStopPlay(LONG IPlayHandle)

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](#)

其他功能

5.22.5 零通道产生一个关键帧 **NET_DVR_ZeroMakeKeyFrame**

函数：BOOL NET_DVR_ZeroMakeKeyFrame(LONG IUserID, LONG IZeroChan)

参数：[in]IUserID NET_DVR_Login_V40 的返回值

[in]IZeroChan 零通道号，加上起始通道号（即从 1 开始）

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

说明：

[返回目录](#)

5.22.6 零通道预览画面翻页 **NET_DVR_ZeroTurnOver**

函 数: `BOOL NET_DVR_ZeroTurnOver(LONG IUserID, LONG IChannel, BOOL bNextPreview)`

参 数: `[in]IUserID` `NET_DVR_Login_V40` 的返回值
 `[in]IChannel` 零通道号，加上起始通道号（即从 1 开始）
 `[in]bNextPreview` 向下或向上翻页：TRUE-下一页；FALSE-上一页

返回值: TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [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 透明通道

5.23.1 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](#)

5.23.2 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](#)

5.23.3 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](#)

5.24 向串口发送数据

5.24.1 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 IUserID, DWORD dwSerialPort, DWORD dwSerialIndex, char *pSendBuf, DWORD dwBufSize)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值

[in]dwSerialPort 串口类型: 1-232, 2-485

[in]dwSerialIndex 表示第几个 232 或者 485, 从 1 开始

[in]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](#)

5.24.2 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 IUserID, char *pSendBuf, DWORD dwBufSize)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值

[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](#)

5.25 设备手动录像

5.25.1 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, 不支持默认为手动录像。](#)
当某通道已经开启定时录像的前提下首次开启手动录像, 此次操作未生效, 仍保持定时录像状态, 且查询设备状态 (见 NET_DVR_GetDVRWorkState_V30 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](#)

5.25.2 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](#)

5.26 远程面板控制

5.26.1 远程控制面板上的按键 **NET_DVR_ClickKey**

函 数: BOOL NET_DVR_ClickKey(LONG IUserID, LONG IKeyIndex)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IKeyIndex 面板上的按键, 详见表 5.65

表 5.65 远程控制面板按键

lKeyIndex 宏定义	宏定义值	含义
KEY_CODE_1	1	按钮 1
KEY_CODE_2	2	按钮 2
KEY_CODE_3	3	按钮 3
KEY_CODE_4	4	按钮 4
KEY_CODE_5	5	按钮 5
KEY_CODE_6	6	按钮 6
KEY_CODE_7	7	按钮 7
KEY_CODE_8	8	按钮 8
KEY_CODE_9	9	按钮 9
KEY_CODE_0	10	按钮 0
KEY_CODE_POWER	11	POWER
KEY_CODE_MENU	12	MENU
KEY_CODE_ENTER	13	ENTER
KEY_CODE_CANCEL	14	ESCS
KEY_CODE_UP/KEY_PTZ_UP_START	15	"上"或者"云台上开始"或者云台上开始
KEY_CODE_DOWN/KEY_PTZ_DOWN_START	16	"下"或者"云台下开始"或者云台下开始
KEY_CODE_LEFT/KEY_PTZ_LEFT_START	17	"左"或者"云台左开始"或者云台左开始
KEY_CODE_RIGHT/KEY_PTZ_RIGHT_START	18	"右"或者"云台右开始"或者云台右开始
KEY_CODE_EDIT/KEY_PTZ_AP1_START	19	"EDIT"或者"光圈+开始"
KEY_CODE_ADD	20	增加
KEY_CODE_MINUS	21	减小
KEY_CODE_PLAY	22	"PLAY"
KEY_CODE_REC	23	"REC"
KEY_CODE_PAN/KEY_PTZ_AP2_START	24	"PAN"或者"光圈-开始"
KEY_CODE_M/KEY_PTZ_FOCUS2_START	25	"多画面"或者"聚焦-开始"
KEY_CODE_A/KEY_PTZ_FOCUS1_START	26	"输入法"或者"聚焦+开始"
KEY_CODE_F1	27	"对讲"
KEY_CODE_F2	28	"系统信息"
KEY_PTZ_UP_STOP	32	"云台上结束"
KEY_PTZ_DOWN_STOP	33	"云台下结束"
KEY_PTZ_LEFT_STOP	34	"云台左结束"
KEY_PTZ_RIGHT_STOP	35	"云台右结束"
KEY_PTZ_AP1_STOP	36	"光圈+结束"

KEY_PTZ_AP2_STOP	37	"光圈-结束"
KEY_PTZ_FOCUS1_STOP	38	"聚焦+结束"
KEY_PTZ_FOCUS2_STOP	39	"聚焦-结束"
KEY_PTZ_B1_START	40	"变倍+开始"
KEY_PTZ_B1_STOP	41	"变倍+结束"
KEY_PTZ_B2_START	42	"变倍-开始"
KEY_PTZ_B2_STOP	43	"变倍-结束"
KEY_CODE_11	44	按钮 11
KEY_CODE_12	45	按钮 12
KEY_CODE_13	46	按钮 13
KEY_CODE_14	47	按钮 14
KEY_CODE_15	48	按钮 15
KEY_CODE_16	49	按钮 16

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

5.26.2 禁用设备本地面板控制 **NET_DVR_LockPanel**

函 数: BOOL NET_DVR_LockPanel(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](#)

5.26.3 恢复设备本地面板控制 **NET_DVR_UnLockPanel**

函 数: BOOL NET_DVR_UnLockPanel(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](#)

5.27 码流加密

5.27.1 设置设备码流加密密钥 **NET_DVR_InquestStreamEncrypt**

函 数: BOOL NET_DVR_InquestStreamEncrypt(LONG IUserID, LONG IChannel, BOOL bEncrypt)
参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in] IChannel 通道号
[in] bEncrypt 加密标记: TRUE-加密; FALSE-不加密
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

5.27.2 获取设备码流加密状态 **NET_DVR_InquestGetEncryptState**

函 数: BOOL NET_DVR_InquestGetEncryptState(LONG IUserID, LONG IChannel, BOOL *bEncrypt)
参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in] IChannel 通道号
[out] bEncrypt 加密标记: TRUE-加密; FALSE-不加密
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

5.28 获取 IPC 协议列表

5.28.1 获取设备支持的 IPC 协议表 **NET_DVR_GetIPCProtoList**

函 数: BOOL NET_DVR_GetIPCProtoList(LONG IUserID, LPNET_DVR_IPC_PROTO_LIST lpProtoList)
参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[out]lpProtoList IPC 协议列表结构
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明: 该接口用于获取当前设备所支持的外接 IPC 的协议。

[返回目录](#)

5.29 远程扫描和修改 IPC 信息

5.29.1 远程扫描获取 IPC 信息列表 **NET_DVR_GetSadpInfoList**

函 数: BOOL NET_DVR_GetSadpInfoList(LONG lUserID, LPNET_DVR_SADPINFO_LIST lpSadpInfoList)
参 数: [in]lUserID NET_DVR_Login_V40 的返回值
[out]lpSadpInfoList IPC 信息列表结构
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

5.29.2 远程修改 IPC 信息 **NET_DVR_UpdateSadpInfo**

函 数: BOOL NET_DVR_UpdateSadpInfo(LONG lUserID, LPNET_DVR_SADP_VERIFY lpSadpVerify, LPNET_DVR_SADPINFO lpSadpInfo)
参 数: [in]lUserID NET_DVR_Login_V40 的返回值
[in]lpSadpVerify 校验信息
[in]lpSadpInfo 修改的 IPC 信息列表结构
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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](#)

5.30 邮件测试

5.30.1 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 lUserID)
参 数: [in]lUserID 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](#)

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

函 数: BOOL NET_DVR_GetEmailTestProgress(LONG lEmailTestHandle, DWORD* pState)
参 数: [in]lEmailTestHandle 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 5.66 所示。

表 5.66 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。

说 明：

[返回目录](#)

5.30.3 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。

说 明：

[返回目录](#)

5.31 文件上传下载

5.31.1 上传文件 **NET_DVR_UploadFile_V40**

函 数：LONG NET_DVR_UploadFile_V40(LONG IUserID, DWORD dwUploadType, LPVOID lpInBuffer, DWORD dwInBufferSize, char *sFileName, LPVOID lpOutBuffer, DWORD dwOutBufferSize)

参 数：[in] IUserID NET_DVR_Login_V40 的返回值

[in] dwUploadType 上传文件类型，详见表 5.67

[in] lpInBuffer 不同的 dwUploadType，输入参数不同，详见表 5.67

[in] dwInBufferSize 输入缓冲区大小

[in] *sFileName 上传文件的绝对路径（包括文件名），最多 128 字节，文件名最多 32 字节（含最后的'\0'）

[out] lpOutBuffer 输出参数。不同的 dwUploadType，输出参数不同，具体参见下文列表

[in] dwOutBufferSize 输出缓冲区大小

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

说 明：

表 5.67 文件上传类型

dwUploadType 宏定义	宏定义值	dwUploadType 含义	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
UPLOAD_CERTIFICATE	1	上传证书	NET_DVR_CERT_PARAM	
UPLOAD_RECORD_FILE	4	上传录像文件(云存储模式)	NET_DVR_UPLOAD_RECORD_INFO	NET_DVR_UPLOAD_FILE_RET

[返回目录](#)

5.31.2 获取文件上传的进度和状态 **NET_DVR_GetUploadState**

函 数: LONG NET_DVR_GetUploadState(LONG IUploadHandle, LPDWORD pProgress)

参 数: [in] IUploadHandle 文件上传的句柄, NET_DVR_UploadFile_V40 的返回值
[out] pProgress 返回的进度值, 取值范围: 0~100

返回值: -1 表示函数调用失败, 其他为上传的状态值: 1- 上传成功; 2- 正在上传; 3- 上传失败; 4- 网络断开, 状态未知。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

5.31.3 停止文件上传 **NET_DVR_UploadClose**

函 数: BOOL NET_DVR_UploadClose(LONG IUploadHandle)

参 数: [in] IUploadHandle 文件上传的句柄, NET_DVR_UploadFile_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](#)

5.31.4 开始下载文件 **NET_DVR_StartDownload**

函 数: LONG NET_DVR_StartDownload(LONG IUserID, DWORD dwDownloadType, LPVOID lpInBuffer, DWORD dwInBufferSize, char const *sFileName)

参 数: [in] IUserID 用户 ID, NET_DVR_Login_V40 的返回值
[in] dwDownloadType 下载文件类型, 详见表 5.68
[in] lpInBuffer 输入参数。不同的 dwUploadType, 输入参数不同, 详见表 5.68
[in] dwInBufferSize 输入缓冲区大小
[in] sFileName 下载文件的保存路径(绝对路径, 包括文件名)

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

说 明:

表 5.68 文件下载类型

dwDownloadType	取值	含义	lpInBuffer 对应结构体
NET_SDK_DOWNLOAD_CERT	0	下载证书	NET_DVR_CERT_PARAM
NET_SDK_DOWNLOAD_IPC_CFG_FILE	1	下载 IPC 配置文件	NULL

[返回目录](#)

5.31.5 获取文件下载的进度和状态 **NET_DVR_GetDownloadState**

函 数: LONG NET_DVR_GetDownloadState(LONG IDownloadHandle, LPDWORD pProgress)

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

返回值: -1 表示函数调用失败, 其他为下载的状态值: 1- 下载成功; 2- 正在下载; 3- 下载失败; 4- 网络断开, 状态未知。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

5.31.6 停止文件下载 **NET_DVR_StopDownload**

函 数: BOOL NET_DVR_StopDownload(LONG IHandle);

参 数: [in] IHandle 文件下载的句柄, NET_DVR_StartDownload 的返回值

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

说 明:

[返回目录](#)

5.32 IP SAN 文件目录查找

5.32.1 查找 IPSAN 文件目录 **NET_DVR_FindIpSanDirectory**

函 数: LONG NET_DVR_FindIpSanDirectory(LONG IUserID, LPNET_DVR_IPSAN_SERACH_PARAM IpIpsanSearchParam)

参 数: [in] IUserID NET_DVR_Login_V40 的返回值
[in] IpIpsanSearchParam IPSAN 文件目录查找参数, 包括 IPSAN 的 IP 地址和端口

返回值: -1 表示失败, 其他值作为 NET_DVR_FindNextDirectory 等接口的句柄。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: 该接口指定了要查找的 IPSAN, 调用成功后, 就可以调用 NET_DVR_FindNextDirectory 接口来逐个获取该 IPSAN 的目录信息。

[返回目录](#)

5.32.2 逐个获取查找到的目录信息 **NET_DVR_FindNextDirectory**

函 数: LONG NET_DVR_FindNextDirectory(LONG IFindHandle, LPNET_DVR_IPSAN_SERACH_RET IpFindData)

参 数: [in] IFindHandle 查找句柄, NET_DVR_FindIpSanDirectory 的返回值
[out] IpFindData 保存文件目录信息的指针

返回值: -1 表示失败, 其他值表示当前的获取状态等信息, 详见表 5.69。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

表 5.69 查找状态信息

宏定义	宏定义值	含义
NET_DVR_FILE_SUCCESS	1000	获取目录信息成功
NET_DVR_FILE_NOFOUND	1001	未查找到目录
NET_DVR_ISFINDING	1002	正在查找请等待
NET_DVR_NOMOREFILE	1003	没有更多的目录，查找结束
NET_DVR_FILE_EXCEPTION	1004	查找目录时异常

说明： 在调用该接口获取查找目录之前，必须先调用 `NET_DVR_FindIpSanDirectory` 得到当前的查找句柄。此接口用于获取一条已查找到的目录信息，若要获取全部的已查找到的目录信息，需要循环调用此接口。

[返回目录](#)

5.32.3 停止 IPSAN 文件目录搜索，释放资源 `NET_DVR_FindDirectoryClose`

函数： `BOOL NET_DVR_FindDirectoryClose(LONG IFindHandle)`

参数： [in] IFindHandle 查找句柄，`NET_DVR_FindIpSanDirectory` 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](#)

5.33 设备维护管理

获取设备工作状态

5.33.1 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： 此处获取的是 IPC 的硬盘状态，其他设备的硬盘状态通过 [NET_DVR_GetDVRConfig](#) 的 `NET_DVR_GET_HDCFG` 命令获取。

[返回目录](#)

5.33.2 获取设备运行状态 `NET_DVR_GetDeviceStatus`

函数： `BOOL NET_DVR_GetDeviceStatus(LONG IUserID, DWORD dwCommand, DWORD dwCount, LPVOID lpInBuffer, DWORD dwInBufferSize, LPVOID lpStatusList, LPVOID lpOutBuffer, DWORD dwOutBufferSize)`

参 数:	[in] IUserID	用户 ID 号, NET_DVR_Login_V30 返回值
	[in] dwCommand	设备配置命令, 详见表 5.70
	[in] dwCount	要设置的个数, 设为 1
	[in] lpInBuffer	配置条件缓冲区, 详见表 5.71
	[in] dwInBufferSize	缓冲区长度
	[out] lpStatusList	错误信息列表, 和要查询的监控点一一对应, 例如 lpStatusList[2]就对应 lpInBuffer[2], 由用户分配内存, 每个错误信息为 4 个字节(1 个 32 位无符号整数值), 参数值: 0- 成功, 大于 0- 失败
	[out] lpOutBuffer	设备返回的参数内容(详见表 5.71), 和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 0 值, 对应 lpOutBuffer 的内容就是无效的
	[in] dwOutBufferSize	输出缓冲区大小

表 5.70 状态获取命令

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_GET_ALARMIN_STATUS	获取报警输入状态	9115
NET_DVR_GET_ALARMOUT_STATUS	获取报警输出状态	9116
NET_DVR_GET_AUDIO_CHAN_STATUS	获取语音对讲状态	9117

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

说 明: 该接口是带有发送数据的批量获取设备状态信息的通用接口。全部获取时 dwCount 置为 0xffffffff, lpInBuffer 置为 NULL, dwInBufferSize 置为 0, lpStatusList 置为 NULL。lpOutBuffer 前面 4 个字节为个数(N), 后面为设备返回的 N 个信息内容(按通道号 1~N 排列), 如果设置的 lpOutBuffer 缓冲区不足, 仅返回部分信息, 可以根据返回的个数(前 4 字节的值)重新获取。不同的获取功能对应不同的结构体和命令号, 如表 5.71 所示。

表 5.71 获取设备状态

dwCommand 宏定义	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_GET_ALARMIN_STATUS	dwCount 个 4 字节报警输入通道号	dwCount 个 4 字节状态值(0-没有报警, 1-有报警)
NET_DVR_GET_ALARMOUT_STATUS	dwCount 个 4 字节报警输出通道号	dwCount 个 4 字节状态值(0-没有报警, 1-有报警)
NET_DVR_GET_AUDIO_CHAN_STATUS	dwCount 为 1, 4 字节语音对讲通道号	1 个 4 字节状态(0-未开启, 1-开启)

[返回目录](#)

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

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

参 数:	[in]IUserID	用户 ID 号, NET_DVR_Login_V40 的返回值
	[in]dwCommand	控制命令, 详见表 5.72
	[in]lpInBuffer	输入参数, 具体内容跟控制命令相关, 详见表 5.72
	[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 5.72 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](#)

5.33.4 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](#)

5.33.5 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

5.33.6 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](#)

5.33.7 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](#)

5.33.8 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](#)

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

函数： int NET_DVR_GetUpgradeState(LONG IUpgradeHandle)

参数： [in]IUpgradeHandle NET_DVR_Upgrade 的返回值

返回值： -1 表示失败，其他值定义：1- 升级成功；2- 正在升级；3- 升级失败；4- 网络断开，状态未知；5- 升级文件语言版本不匹配。

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

说明：

[返回目录](#)

5.33.10 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 5.73 所示。

表 5.73 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](#)

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

函 数： BOOL NET_DVR_CloseUpgradeHandle(LONG lUpgradeHandle)

参 数： [in]lUpgradeHandle 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](#)

在线升级

5.33.12 获取在线升级相关信息 **NET_DVR_GetSTDConfig**

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

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 5.74

[in&out]lpConfigParam 配置输入输出参数，不同的配置功能对应不同的输入输出参数，详见表 5.75

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

说 明： 获取配置参数时，lpConfigParam 结构体中的 lpInBuffer 无效，设为 NULL。对于不同的配置功能（dwCommand），lpConfigParam 中 lpCondBuffer、lpOutBuffer 分别对应不同的内容，具体如表 5.75 示。

表 5.74 参数获取命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_ONLINEUPGRADE_PROGRESS	获取在线升级状态和进度	9302
NET_DVR_GET_FIRMWARECODE	获取识别码	9303
NET_DVR_GET_ONLINEUPGRADE_SERVER	获取升级服务器状态	9304
NET_DVR_GET_ONLINEUPGRADE_VERSION	获取新版本信息	9305
NET_DVR_GET_RECOMMEN_VERSION	检测是否推荐升级到此版本	9306

表 5.75 参数获取结构

dwCommand 宏定义	lpCondBuffer	lpOutBuffer
NET_DVR_GET_ONLINEUPGRADE_PROGRESS	NULL	NET_DVR_ONLINEUPGRADE_STATUS
NET_DVR_GET_FIRMWARECODE	NET_DVR_FIRMWARECODE_COND	NET_DVR_FIRMWARECODE_LIST
NET_DVR_GET_ONLINEUPGRADE_SERVER	NULL	NET_DVR_ONLINEUPGRADE_SERVER
NET_DVR_GET_ONLINEUPGRADE_VERSION	NET_DVR_ONLINEUPGRADE_VERSION_COND	NET_DVR_ONLINEUPGRADE_VERSION_RET
NET_DVR_GET_RECOMMEN_VERSION	NET_DVR_RECOMMEN_VERSION_COND	NET_DVR_RECOMMEN_VERSION_RET

[返回目录](#)

5.33.13 远程控制在线升级 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 5.49
[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 5.49

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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 5.49 所示。

表 5.76 远程控制命令

dwCommand 宏定义	宏定义值	含义	lpCondBuffer 对应结构体
NET_DVR_SET_ONLINE_UPGRADE	9301	允许在线升级	NULL

- 设备是否支持在线升级功能或者支持的参数能力, 可以通过设备能力集进行判断, 对应在线升级能力集(OnlineUpgradeCap), 相关接口: [NET_DVR_GetSTDAbility](#), 能力集类型: NET_DVR_GET_ONLINEUPGRADE_ABILITY。
- 在线升级过程:
 - 1) 设备注册到服务器, 从服务器上获取新版本信息。
 - 2) 客户端登录设备后在通过接口 NET_DVR_GetSTDConfig 获取可以升级的服务器、版本等信息, 然后调用该接口下发命令允许在线升级, 升级过程中可以定时获取升级进度。
 - 3) 设备收到可在线升级命令后从服务器下载升级包自行升级。

[返回目录](#)

日志查找

5.33.14 查找设备的日志信息（可搜索带 S.M.A.R.T 信息的日志）

NET_DVR_FindDVRLog_V30

函 数: LONG NET_DVR_FindDVRLog_V30(LONG lUserID, LONG lSelectMode, DWORD dwMajorType, DWORD dwMinorType, LPNET_DVR_TIME lpStartTime, LPNET_DVR_TIME lpStopTime, BOOL bOnlySmart = FALSE)

参 数: [in]lUserID NET_DVR_Login_V40 的返回值
[in]lSelectMode 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 5.77
[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见表 5.78~表 5.81
[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, 一般设备 (81xx, 80xx 等) 支持 2000 条, 9000v2.0 支持 4000 条, 81xxST 支持 1000 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 5.77 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 5.78 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

表 5.79 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
MINOR_RECORD_OVERFLOW	0x41	缓冲区溢出
MINOR_DSP_ABNORMAL	0x42	DSP 异常
MINOR_ANR_RECORD_FAIED	0x43	ANR 录像失败
MINOR_SPARE_WORK_DEVICE_EXCEPT	0x44	热备设备工作机异常
MINOR_START_IPC_MAS_FAILED	0x45	开启 IPC MAS 失败
MINOR_IPCM_CRASH	0x46	IPCM 异常重启
MINOR_POE_POWER_EXCEPTION	0x47	POE 供电异常

MINOR_UPLOAD_DATA_CS_EXCEPTION	0x48	云存储数据上传失败
MINOR_SYNC_IPC_PASSWD	0x53	同步 IPC 密码异常
MINOR_EZVIZ_OFFLINE	0x54	萤石下线异常

表 5.80 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_LOCAL_ADD_FILE	0x99	本地导入文件
MINOR_REMOTE_DELETE_HDISK	0x9a	远程删除异常不存在的硬盘
MINOR_REMOTE_LOAD_HDISK	0x9b	远程加载硬盘
MINOR_REMOTE_UNLOAD_HDISK	0x9c	远程卸载硬盘
MINOR_LOCAL_OPERATE_LOCK	0x9d	本地操作锁定
MINOR_LOCAL_OPERATE_UNLOCK	0x9e	本地操作解除锁定
MINOR_REMOTE_BYPASS	0xd0	远程旁路
MINOR_REMOTE_UNBYPASS	0xd1	远程旁路恢复
MINOR_REMOTE_SET_ALARMIN_CFG	0xd2	远程设置报警输入参数
MINOR_REMOTE_GET_ALARMIN_CFG	0xd3	远程获取报警输入参数
MINOR_REMOTE_SET_ALARMOUT_CFG	0xd4	远程设置报警输出参数
MINOR_REMOTE_GET_ALARMOUT_CFG	0xd5	远程获取报警输出参数
MINOR_REMOTE_ALARMOUT_OPEN_MAN	0xd6	远程手动开启报警输出
MINOR_REMOTE_ALARMOUT_CLOSE_MAN	0xd7	远程手动关闭报警输出
MINOR_REMOTE_ALARM_ENABLE_CFG	0xd8	远程设置报警主机的 RS485 串口使能状态
MINOR_DBDATA_OUTPUT	0xd9	导出数据库记录
MINOR_DBDATA_INPUT	0xda	导入数据库记录
MINOR_MU_SWITCH	0xdb	级联切换
MINOR_MU_PTZ	0xdc	级联 PTZ 控制
MINOR_LOCAL_CONF_REB_RAID	0x101	本地配置自动重建
MINOR_LOCAL_CONF_SPARE	0x102	本地配置热备
MINOR_LOCAL_ADD_RAID	0x103	本地创建阵列
MINOR_LOCAL_DEL_RAID	0x104	本地删除阵列

MINOR_LOCAL_MIG_RAID	0x105	本地迁移阵列
MINOR_LOCAL_REB_RAID	0x106	本地手动重建阵列
MINOR_LOCAL_QUICK_CONF_RAID	0x107	本地一键配置
MINOR_LOCAL_ADD_VD	0x108	本地创建虚拟磁盘
MINOR_LOCAL_DEL_VD	0x109	本地删除虚拟磁盘
MINOR_LOCAL_RP_VD	0x10a	本地修复虚拟磁盘
MINOR_LOCAL_FORMAT_EXPANDVD	0x10b	本地扩展虚拟磁盘扩容
MINOR_LOCAL_RAID_UPGRADE	0x10c	本地 raid 卡升级
MINOR_LOCAL_STOP_RAID	0x10d	本地暂停 RAID 操作(即安全拔盘)
MINOR_REMOTE_CONF_REB_RAID	0x111	远程配置自动重建
MINOR_REMOTE_CONF_SPARE	0x112	远程配置热备
MINOR_REMOTE_ADD_RAID	0x113	远程创建阵列
MINOR_REMOTE_DEL_RAID	0x114	远程删除阵列
MINOR_REMOTE_MIG_RAID	0x115	远程迁移阵列
MINOR_REMOTE_REB_RAID	0x116	远程手动重建阵列
MINOR_REMOTE_QUICK_CONF_RAID	0x117	远程一键配置
MINOR_REMOTE_ADD_VD	0x118	远程创建虚拟磁盘
MINOR_REMOTE_DEL_VD	0x119	远程删除虚拟磁盘
MINOR_REMOTE_RP_VD	0x11a	远程修复虚拟磁盘
MINOR_REMOTE_FORMAT_EXPANDVD	0x11b	远程虚拟磁盘扩容
MINOR_REMOTE_RAID_UPGRADE	0x11c	远程 raid 卡升级
MINOR_REMOTE_STOP_RAID	0x11d	远程暂停 RAID 操作(即安全拔盘)
MINOR_LOCAL_START_PIC_REC	0x121	本地开始抓图
MINOR_LOCAL_STOP_PIC_REC	0x122	本地停止抓图
MINOR_LOCAL_SET_SNMP	0x125	本地配置 SNMP
MINOR_LOCAL_TAG_OPT	0x126	本地标签操作
MINOR_REMOTE_START_PIC_REC	0x131	远程开始抓图
MINOR_REMOTE_STOP_PIC_REC	0x132	远程停止抓图
MINOR_REMOTE_SET_SNMP	0x135	远程配置 SNMP
MINOR_REMOTE_TAG_OPT	0x136	远程标签操作
MINOR_REMOTE_TAG_OPT	0x136	远程标签操作
MINOR_LOCAL_VOUT_SWITCH	0x140	本地输出口切换操作
MINOR_STREAM_CABAC	0x141	码流压缩性能选项配置操作
MINOR_LOCAL_LOAD_HDISK	0x300	本地加载硬盘
MINOR_LOCAL_DELETE_HDISK	0x301	本地删除异常不存在的硬盘

MINOR_REMOTE_CREATE_STORAGE_POOL	0x211c	远程添加存储池
MINOR_REMOTE_DEL_STORAGE_POOL	0x211d	远程删除存储池
MINOR_REMOTE_DEL_PIC	0x2120	远程删除图片数据
MINOR_REMOTE_DEL_RECORD	0x2121	远程删除录像数据
MINOR_REMOTE_CLOUD_ENABLE	0x2123	远程设置云存储启用
MINOR_REMOTE_CLOUD_DISABLE	0x2124	远程设置云存储禁用
MINOR_REMOTE_CLOUD_MODIFY_PARAM	0x2125	远程修改云存储池参数
MINOR_REMOTE_CLOUD_MODIFY_VOLUME	0x2126	远程修改云存储池容量

表 5.81 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_RUN_STATUS_INFO	0xaa	系统运行状态信息
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](#)

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

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

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

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](#)

5.33.16 释放查找日志的资源 NET_DVR_FindLogClose_V30

函 数: BOOL NET_DVR_FindLogClose_V30(LONG lLogHandle)

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

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

说 明:

[返回目录](#)

远程备份

5.33.17 获取设备磁盘列表 NET_DVR_GetDiskList

函 数: BOOL NET_DVR_GetDiskList(LONG lUserID, LPNET_DVR_DISKABILITY_LIST lpDiskList)

参 数: [in]lUserID NET_DVR_Login_V40 的返回值

[out]lpDiskList 设备可用备份磁盘信息结构

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

说 明: 该接口用于获取设备可用备份文件的磁盘资源信息, 在备份文件功能接口的输入参数中需要用到。

[返回目录](#)

5.33.18 备份统一接口 NET_DVR_Backup

函 数: DWORD NET_DVR_Backup(long lUserID, DWORD dwBackupType, void* lpBackupBuff, DWORD dwBackupBuffSize)

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

[in]dwBackupType 备份类型: 1- 按文件名备份录像文件, 2- 按时间段备份录像文件, 3- 备份图片, 4- 恢复审讯事件, 5- 备份日志

[in]lpBackupBuff 指向备份参数指针, 详见表 5.82

[in]dwBackupBuffSize 备份参数大小

返回值: -1 失败, 0-510 的返回句柄值作为 NET_DVR_GetBackupProgress, NET_DVR_StopBackup 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: 不同的备份类型 (dwBackupType) 对应不同的备份参数结构体 (lpBackupBuff), 详见表 5.82。

表 5.82 备份操作类型

dwBackupType	含义	lpBackupBuff
1	按文件名备份录像文件	NET_DVR_BACKUP_NAME_PARAM
2	按时间段备份录像文件	NET_DVR_BACKUP_TIME_PARAM
3	备份图片	NET_DVR_BACKUP_PICTURE_PARAM
5	备份日志	NET_DVR_BACKUP_LOG_PARAM

[返回目录](#)

5.33.19 获取备份的进度 **NET_DVR_GetBackupProgress**

函 数: BOOL NET_DVR_GetBackupProgress(LONG lHandle, DWORD* pState)

参 数: [in]lHandle NET_DVR_Backup 的返回值

[out] pState 当前备份的进度，进度值的取值范围为[0,100)，其他值的定义见表 5.83

表 5.83 磁盘备份进度

宏定义	宏定义值	含义
BACKUP_SUCCESS	100	备份完成
BACKUP_CHANGE_DEVICE	101	备份设备已满，更换设备继续备份
BACKUP_SEARCH_DEVICE	300	正在搜索备份设备
BACKUP_SEARCH_FILE	301	正在搜索录像文件或者图片
BACKUP_EXCEPTION	400	备份异常
BACKUP_FAIL	500	备份失败
BACKUP_TIME_SEG_NO_FILE	501	时间段内无录像文件或者图片
BACKUP_NO_RESOURCE	502	申请不到资源
BACKUP_DEVICE_LOW_SPACE	503	备份设备容量不足
BACKUP_DISK_FINALIZED	504	刻录光盘封盘
BACKUP_DISK_EXCEPTION	505	刻录光盘异常
BACKUP_DEVICE_NOT_EXIST	506	备份设备不存在
BACKUP_OTNER_BACKUP_WORK	507	有其他备份操作在进行
BACKUP_USER_NO_RIGHT	508	用户没有操作权限
BACKUP_OPERATE_FAIL	509	操作失败

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。

说 明: 在进度为 100 或者备份出错时，需调用 NET_DVR_StopBackup()停止备份。

[返回目录](#)

5.33.20 停止备份 **NET_DVR_StopBackup**

函 数: BOOL NET_DVR_StopBackup(LONG lHandle)

参 数: [in]lHandle NET_DVR_Backup 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](#)

恢复设备默认参数

5.33.21 恢复设备默认参数 **NET_DVR_RestoreConfig**

函 数: `BOOL NET_DVR_RestoreConfig(LONG lUserID)`

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

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

说 明:

[返回目录](#)

导入/导出配置文件

5.33.22 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](#)

5.33.23 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](#)

5.33.24 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](#)

5.33.25 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](#)

网流量检测

5.33.26 开始网络流量检测 **NET_DVR_StartNetworkFlowTest**

函 数: LONG NET_DVR_StartNetworkFlowTest(LONG IUserID, NET_DVR_FLOW_TEST_PARAM* pFlowTest, FLOWTESTCALLBACK fFlowTestCallback, void *pUser)

参 数: [in]IUserID NET_DVR_Login_V30 的返回

[in]pFlowTest 网络流量参数

[in]fFlowTestCallback 网络流量检测回调函数

[in]pUser 用户数据

```
typedef void(CALLBACK *fFlowTestCallback)(LONG IFlowHandle, LPNET_DVR_FLOW_INFO
pFlowInfo, void *pUser)
```

[out]IFlowHandle 当前的检测句柄

[out]pFlowInfo 网络流量检测结果

[out]pUser 用户数据

返回值: -1 表示失败, 其他值作为 NET_DVR_StopNetworkFlowTest 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:

[返回目录](#)

5.33.27 停止网络流量检测 **NET_DVR_StopNetworkFlowTest**

函 数: BOOL NET_DVR_StopNetworkFlowTest(LONG IHandle)

参 数: [in]IHandle NET_DVR_StartNetworkFlowTest 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](#)

获取 UPNP 端口映射状态

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

函 数: BOOL NET_DVR_GetUpnpNatState(LONG IUserID, LPNET_DVR_UPNP_NAT_STATE lpState)

参 数: [in]IUserID NET_DVR_Login_V30 的返回
[out]lpState UPNP 端口映射状态

返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [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

5.33.29 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](#)

5.33.30 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_USER_NOT_SUCC_LOGIN	102	用户还没登录成功
NET_DVR_USE_LOG_SWITCH_FILE	103	正在使用日志开关文件
NET_DVR_POOL_PORT_EXHAUST	104	端口池中用于绑定的端口已耗尽
NET_DVR_PACKET_TYPE_NOT_SUPPORT	105	码流封装格式错误
NET_DVR_IPPARA_IPID_ERROR	106	IP 接入配置时 IPID 有误
NET_DVR_LOAD_HCPREVIEW_SDK_ERROR	107	预览组件加载失败
NET_DVR_LOAD_HCVOICETALK_SDK_ERROR	108	语音组件加载失败
NET_DVR_LOAD_HCALARM_SDK_ERROR	109	报警组件加载失败

NET_DVR_LOAD_HCPLAYBACK_SDK_ERROR	110	回放组件加载失败
NET_DVR_LOAD_HCDISPLAY_SDK_ERROR	111	显示组件加载失败
NET_DVR_LOAD_HCINDUSTRY_SDK_ERROR	112	行业应用组件加载失败
NET_DVR_LOAD_HCGENERALCFGMGGR_SDK_ERROR	113	通用配置管理组件加载失败
NET_DVR_CORE_VER_MISMATCH	121	单独加载组件时，组件与 core 版本不匹配
NET_DVR_CORE_VER_MISMATCH_HCPREVIEW	122	预览组件与 core 版本不匹配
NET_DVR_CORE_VER_MISMATCH_HCVOICETALK	123	语音组件与 core 版本不匹配
NET_DVR_CORE_VER_MISMATCH_HCALARM	124	报警组件与 core 版本不匹配
NET_DVR_CORE_VER_MISMATCH_HCPLAYBACK	125	回放组件与 core 版本不匹配
NET_DVR_CORE_VER_MISMATCH_HCDISPLAY	126	显示组件与 core 版本不匹配
NET_DVR_CORE_VER_MISMATCH_HCINDUSTRY	127	行业应用组件与 core 版本不匹配
NET_DVR_CORE_VER_MISMATCH_HCGENERALCFGMGGR	128	通用配置管理组件与 core 版本不匹配
NET_DVR_COM_VER_MISMATCH_HCPREVIEW	136	预览组件与 HCNetSDK 版本不匹配
NET_DVR_COM_VER_MISMATCH_HCVOICETALK	137	语音组件与 HCNetSDK 版本不匹配
NET_DVR_COM_VER_MISMATCH_HCALARM	138	报警组件与 HCNetSDK 版本不匹配
NET_DVR_COM_VER_MISMATCH_HCPLAYBACK	139	回放组件与 HCNetSDK 版本不匹
NET_DVR_COM_VER_MISMATCH_HCDISPLAY	140	显示组件与 HCNetSDK 版本不匹配
NET_DVR_COM_VER_MISMATCH_HCINDUSTRY	141	行业应用组件与 HCNetSDK 版本不匹配
NET_DVR_COM_VER_MISMATCH_HCGENERALCFGMGGR	142	通用配置管理组件与 HCNetSDK 版本不匹配
NET_DVR_ALIAS_DUPLICATE	150	别名重复（EasyDDNS 的配置）
NET_DVR_USERNAME_NOT_EXIST	152	用户名不存在（V5.1.7~V5.3.1 版本的 IPC、IPD 的错误码）
NET_ERR_USERNAME_LOCKED	153	用户名被锁定
NET_DVR_INVALID_USERID	154	无效用户 ID
NET_DVR_LOW_LOGIN_VERSION	155	登录版本低
NET_DVR_LOAD_LIBEAY32_DLL_ERROR	156	加载 libeay32.dll 库失败
NET_DVR_LOAD_SSLEAY32_DLL_ERROR	157	加载 ssleay32.dll 库失败
NET_DVR_TEST_SERVER_FAIL_CONNECT	165	连接测试服务器失败
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_SDK_ERR_CHAN_AUDIO_BIND	821	通道未绑定或绑定语音对讲失败
NET_DVR_N_PLUS_ONE_MODE	822	设备当前处于 N+1 模式，不支持设置云存储
NET_DVR_CLOUD_STORAGE_OPENED	823	云存储模式已开启
阵列错误码		
NET_DVR_NAME_NOT_ONLY	200	名称已存在
NET_DVR_OVER_MAX_ARRAY	201	阵列达到上限
NET_DVR_OVER_MAX_VD	202	虚拟磁盘达到上限
NET_DVR_VD_SLOT_EXCEED	203	虚拟磁盘槽位已满
NET_DVR_PD_STATUS_INVALID	204	重建阵列所需物理磁盘状态错误
NET_DVR_PD_BE_DEDICATE_SPARE	205	重建阵列所需物理磁盘为指定热备
NET_DVR_PD_NOT_FREE	206	重建阵列所需物理磁盘非空闲
NET_DVR_CANNOT_MIG2NEWMODE	207	不能从当前的阵列类型迁移到新的阵列类型
NET_DVR_MIG_PAUSE	208	迁移操作已暂停

NET_DVR_MIG_CANCEL	209	正在执行的迁移操作已取消
NET_DVR_EXIST_VD	210	阵列上存在虚拟磁盘，无法删除阵列
NET_DVR_TARGET_IN_LD_FUNCTIONAL	211	对象物理磁盘为虚拟磁盘组成部分且工作正常
NET_DVR_HD_IS_ASSIGNED_ALREADY	212	指定的物理磁盘被分配为虚拟磁盘
NET_DVR_INVALID_HD_COUNT	213	物理磁盘数量与指定的 RAID 等级不匹配
NET_DVR_LD_IS_FUNCTIONAL	214	阵列正常，无法重建
NET_DVR_BGA_RUNNING	215	存在正在执行的后台任务
NET_DVR_LD_NO_ATAPI	216	无法用 ATAPI 盘创建虚拟磁盘
NET_DVR_MIGRATION_NOT_NEED	217	阵列无需迁移
NET_DVR_HD_TYPE_MISMATCH	218	物理磁盘不属于同意类型
NET_DVR_NO_LD_IN_DG	219	无虚拟磁盘，无法进行此项操作
NET_DVR_NO_ROOM_FOR_SPARE	220	磁盘空间过小，无法被指定为热备盘
NET_DVR_SPARE_IS_IN_MULTI_DG	221	磁盘已被分配为某阵列热备盘
NET_DVR_DG_HAS_MISSING_PD	222	阵列缺少盘
NET_DVR_NAME_EMPTY	223	名称为空
NET_DVR_INPUT_PARAM	224	输入参数有误
NET_DVR_PD_NOT_AVAILABLE	225	物理磁盘不可用
NET_DVR_ARRAY_NOT_AVAILABLE	226	阵列不可用
NET_DVR_PD_COUNT	227	物理磁盘数不正确
NET_DVR_VD_SMALL	228	虚拟磁盘太小
NET_DVR_NO_EXIST	229	不存在
NET_DVR_NOT_SUPPORT	230	不支持该操作
NET_DVR_NOT_FUNCTIONAL	231	阵列状态不是正常状态
NET_DVR_DEV_NODE_NOT_FOUND	232	虚拟磁盘设备节点不存在
NET_DVR_SLOT_EXCEED	233	槽位达到上限
NET_DVR_NO_VD_IN_ARRAY	234	阵列上不存在虚拟磁盘
NET_DVR_VD_SLOT_INVALID	235	虚拟磁盘槽位无效
NET_DVR_PD_NO_ENOUGH_SPACE	236	所需物理磁盘空间不足
NET_DVR_ARRAY_NONFUNCTION	237	只有处于正常状态的阵列才能进行迁移
NET_DVR_ARRAY_NO_ENOUGH_SPACE	238	阵列空间不足
NET_DVR_STOPPING_SCANNING_ARRAY	239	正在执行安全拔盘或重新扫描
NET_DVR_NOT_SUPPORT_16T	240	不支持创建大于 16T 的阵列
安全激活相关错误码		
NET_DVR_ERROR_DEVICE_NOT_ACTIVATED	250	设备未激活（设备未激活时，登录失败，返回错误码）
NET_DVR_ERROR_RISK_PASSWORD	251	有风险的密码（设置用户密码或者激活的时候为风险密码）
NET_DVR_ERROR_DEVICE_HAS_ACTIVATED	252	设备已激活（已激活的设备，再次激活时返回错误）
N+1 功能错误码		
NET_SDK_ERR_REMOTE_DISCONNECT	803	远端无法连接
NET_SDK_ERR_RD_ADD_RD	804	备机不能添加备机
NET_SDK_ERR_BACKUP_DISK_EXCEPT	805	备份盘异常
NET_SDK_ERR_RD_LIMIT	806	备机数已达上限
NET_SDK_ERR_ADDED_RD_IS_WD	807	添加的备机是工作机
NET_SDK_ERR_ADD_ORDER_WRONG	808	添加顺序出错，比如没有被工作机添加为备机，就添

		加工作机
NET_SDK_ERR_WD_ADD_WD	809	工作机不能添加工作机
NET_SDK_ERR_WD_SERVICE_EXCETP	810	工作机 CVR 服务异常
NET_SDK_ERR_RD_SERVICE_EXCETP	811	备机 CVR 服务异常
NET_SDK_ERR_ADDED_WD_IS_RD	812	添加的工作机是备机
NET_SDK_ERR_PERFORMANCE_LIMIT	813	性能达到上限
NET_SDK_ERR_ADDED_DEVICE_EXIST	814	添加的设备已经存在
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
VQD 错误码		
NET_ERR_VQD_TIME_CONFLICT	1500	VQD 诊断时间段冲突
NET_ERR_VQD_PLAN_NO_EXIST	1501	VQD 诊断计划不存在
NET_ERR_VQD_CHAN_NO_EXIST	1502	VQD 监控点不存在
NET_ERR_VQD_CHAN_MAX	1503	VQD 计划数已达上限
NET_ERR_VQD_TASK_MAX	1504	VQD 任务数已达上限

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_DESCRIBERROR	410	RTSP DESCRIBE 交互错误
NET_DVR_RTSP_DESCRIBESENDTIMEOUT	411	RTSP DESCRIBE 发送超时
NET_DVR_RTSP_DESCRIBESENDERROR	412	RTSP DESCRIBE 发送失败
NET_DVR_RTSP_DESCRIBERECDTIMEOUT	413	RTSP DESCRIBE 接收超时
NET_DVR_RTSP_DESCRIBERECDATALOST	414	RTSP DESCRIBE 接收数据错误
NET_DVR_RTSP_DESCRIBERECDERROR	415	RTSP DESCRIBE 接收失败

NET_DVR_RTSP_DESCRIBESERVERERR	416	RTSP DESCRIBE 服务器返回错误状态。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_SETUPSENDDTIMEOUT	421	RTSP SETUP 发送超时
NET_DVR_RTSP_SETUPSENDERERROR	422	RTSP SETUP 发送错误
NET_DVR_RTSP_SETUPRECVTIMEOUT	423	RTSP SETUP 接收超时
NET_DVR_RTSP_SETUPRECVDATALOST	424	RTSP SETUP 接收数据错误
NET_DVR_RTSP_SETUPRECVERROR	425	RTSP SETUP 接收失败
NET_DVR_RTSP_OVER_MAX_CHAN	426	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_PLAYSENDDTIMEOUT	431	RTSP PLAY 发送超时
NET_DVR_RTSP_PLAYSENDERERROR	432	RTSP PLAY 发送错误
NET_DVR_RTSP_PLAYRECVTIMEOUT	433	RTSP PLAT 接收超时
NET_DVR_RTSP_PLAYRECVDATALOST	434	RTSP PLAY 接收数据错误
NET_DVR_RTSP_PLAYRECVERROR	435	RTSP PLAY 接收失败
NET_DVR_RTSP_PLAYSERVERERR	436	RTSP PLAY 服务器返回错误状态
NET_DVR_RTSP_TEARDOWNERROR	440	RTSP TEARDOWN 交互错误
NET_DVR_RTSP_TEARDOWNSENDDTIMEOUT	441	RTSP TEARDOWN 发送超时
NET_DVR_RTSP_TEARDOWNSENDERERROR	442	RTSP TEARDOWN 发送错误
NET_DVR_RTSP_TEARDOWNRECVTIMEOUT	443	RTSP TEARDOWN 接收超时
NET_DVR_RTSP_TEARDOWNRECVDATALOST	444	RTSP TEARDOWN 接收数据错误
NET_DVR_RTSP_TEARDOWNRECVERROR	445	RTSP TEARDOWN 接收失败
NET_DVR_RTSP_TEARDOWNSERVERERR	446	RTSP TEARDOWN 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_MEMERY	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