

设备(CVR)

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.....	5
3 函数调用顺序	10
3.1 SDK 基本调用的主要流程	10
3.2 通道管理模块流程.....	12
3.3 实时预览模块流程.....	14
3.4 云台控制模块流程.....	15
3.5 回放和下载模块流程.....	16
3.6 参数配置模块流程.....	17
3.7 远程设备维护模块流程.....	19
3.8 报警模块流程.....	20
3.8.1 报警（布防）流程	20
3.8.2 报警（监听）流程	21
4 函数调用实例	22
4.1 通道配置示例代码.....	22
4.2 预览模块示例代码.....	25
4.3 云台控制模块示例代码	31
4.4 回放和下载模块示例代码	34
4.5 参数配置模块示例代码	40
4.6 远程设备维护模块示例代码	42
4.7 报警模块示例代码.....	45
5 函数说明	50
5.1 SDK 初始化	50
5.1.1 初始化 SDK NET_DVR_Init	50
5.1.2 释放 SDK 资源 NET_DVR_Cleanup	50
5.2 SDK 本地功能	50
SDK 本地参数配置.....	50
5.2.1 获取 SDK 本地参数 NET_DVR_GetSDKLocalCfg	50
5.2.2 设置 SDK 本地参数 NET_DVR_SetSDKLocalCfg.....	51
连接和接收超时时间及重连设置	52
5.2.3 设置网络连接超时时间和连接尝试次数 NET_DVR_SetConnectTime	52
5.2.4 设置重连功能 NET_DVR_SetReconnect.....	52
5.2.5 设置接收超时时间 NET_DVR_SetRecvTimeOut	52
多网卡绑定	52
5.2.6 获取所有 IP，用于支持多网卡接口 NET_DVR_GetLocalIP	52

5.2.7	设置 IP 绑定 NET_DVR_SetValidIP.....	53
	SDK 版本、状态、能力.....	53
5.2.8	获取 SDK 的版本号和 build 信息 NET_DVR_GetSDKBuildVersion	53
5.2.9	获取当前 SDK 的状态信息 NET_DVR_GetSDKState	53
5.2.10	获取当前 SDK 的功能信息 NET_DVR_GetSDKAbility	54
	SDK 启用写日志.....	54
5.2.11	启用写日志文件 NET_DVR_SetLogToFile.....	54
	异常消息回调.....	54
5.2.12	注册接收异常、重连等消息的窗口句柄或回调函数 NET_DVR_SetExceptionCallBack_V30.....	54
	获取错误信息.....	56
5.2.13	返回最后操作的错误码 NET_DVR_GetLastError.....	56
5.2.14	返回最后操作的错误码信息 NET_DVR_GetErrorMsg	56
5.3	用户注册.....	57
5.3.1	通过解析服务器，获取设备的动态 IP 地址和端口号 NET_DVR_GetDVRIPByResolveSvr_EX	57
5.3.2	用户注册设备 NET_DVR_Login_V40.....	57
5.3.3	用户注销 NET_DVR_Logout.....	58
5.4	获取设备能力集.....	58
5.4.1	获取设备能力集 NET_DVR_GetDeviceAbility	58
5.4.2	获取设备能力集 NET_DVR_GetSTDAbility.....	59
5.5	实时预览.....	60
	强制 I 帧	60
5.5.1	强制 I 帧控制 NET_DVR_RemoteControl	60
5.5.2	主码流动态产生一个关键帧 NET_DVR_MakeKeyFrame	60
5.5.3	子码流动态产生一个关键帧 NET_DVR_MakeKeyFrameSub	60
	实时预览	61
5.5.4	实时预览 NET_DVR_RealPlay_V40.....	61
5.5.5	停止预览 NET_DVR_StopRealPlay.....	61
5.5.6	获取预览时用来解码和显示的播放库句柄 NET_DVR_GetRealPlayerIndex.....	62
	预览画面叠加字符和图像	62
5.5.7	预览画面叠加字符和图像，Linux 下无此接口 NET_DVR_RigisterDrawFun	62
	预览时播放声音控制.....	62
5.5.8	设置声音播放模式 NET_DVR_SetAudioMode.....	62
5.5.9	独占声卡模式下开启声音 NET_DVR_OpenSound	63
5.5.10	独占声卡模式下开启声音 NET_DVR_CloseSound	63
5.5.11	共享声卡模式下开启声音 NET_DVR_OpenSoundShare	63
5.5.12	共享声卡模式下关闭声音 NET_DVR_CloseSoundShare	63
5.5.13	调节播放音量 NET_DVR_Volume	64
	实时数据回调和录像.....	64
5.5.14	设置实时码流数据回调函数 NET_DVR_SetRealDataCallBack	64
5.5.15	捕获数据并保存到指定的文件中 NET_DVR_SaveRealData	65
5.5.16	停止数据捕获 NET_DVR_StopSaveRealData.....	65
	预览抓图	65
5.5.17	设置抓图模式 NET_DVR_SetCapturePictureMode	65
5.5.18	预览时，单帧数据捕获并保存成图片 NET_DVR_CapturePicture.....	65

5.6	云台控制.....	66
	云台控制操作.....	66
5.6.1	云台控制操作（需先启动图像预览）NET_DVR_PTZControl.....	66
5.6.2	云台控制操作（不用启动图像预览）NET_DVR_PTZControl_Other.....	67
5.6.3	带速度的云台控制操作（需先启动图像预览）NET_DVR_PTZControlWithSpeed.....	68
5.6.4	带速度的云台控制操作（不用启动图像预览）NET_DVR_PTZControlWithSpeed_Other.....	68
	云台预置点操作.....	69
5.6.5	云台预置点操作，需先启动预览 NET_DVR_PTZPreset.....	69
5.6.6	云台预置点操作 NET_DVR_PTZPreset_Other.....	69
	云台巡航操作.....	70
5.6.7	云台巡航操作，需先启动预览 NET_DVR_PTZPCruise.....	70
5.6.8	云台巡航操作 NET_DVR_PTZPCruise_Other.....	70
	云台花样扫描路径操作.....	71
5.6.9	云台花样扫描路径操作，需先启动预览 NET_DVR_PTZTrack.....	71
5.6.10	云台花样扫描路径操作 NET_DVR_PTZTrack_Other.....	71
5.7	录像回放、下载、锁定及备份.....	72
	月历录像查询.....	72
5.7.1	获取月历录像分布 NET_DVR_GetDeviceConfig.....	72
	通道模式录像查找.....	73
5.7.2	根据文件类型、时间查找设备录像文件 NET_DVR_FindFile_V40.....	73
5.7.3	逐个获取查找到的文件信息 NET_DVR_FindNextFile_V40.....	73
5.7.4	关闭文件查找，释放资源 NET_DVR_FindClose_V30.....	73
	流 ID 模式录像查找.....	74
5.7.5	根据事件查找录像文件 NET_DVR_FindFileByEvent_V40.....	74
5.7.6	逐个获取查找到的文件信息 NET_DVR_FindNextEvent_V40.....	74
5.7.7	关闭文件查找，释放资源 NET_DVR_FindClose_V30.....	74
	智能搜索.....	75
5.7.8	开始智能搜索 NET_DVR_SmartSearch_V40.....	75
5.7.9	逐个获取查找到的智能录像信息 NET_DVR_SearchNextInfo.....	75
5.7.10	停止智能搜索 NET_DVR_StopSearch.....	76
	录像远程回放.....	76
5.7.11	按时间回放录像 NET_DVR_PlayBackByTime_V40.....	76
5.7.12	按时间倒放录像 NET_DVR_PlayBackReverseByTime_V40.....	77
5.7.13	控制录像回放的状态 NET_DVR_PlayBackControl_V40.....	77
5.7.14	停止回放录像 NET_DVR_StopPlayBack.....	79
	回放数据回调和录像.....	79
5.7.15	捕获回放的录像数据，并保存成文件 NET_DVR_PlayBackSaveData.....	79
5.7.16	停止保存录像数据 NET_DVR_StopPlayBackSave.....	79
5.7.17	注册回调函数，捕获录像数据 NET_DVR_SetPlayDataCallBack_V40.....	80
	回放抓图等其他操作.....	80
5.7.18	获取录像回放时显示的 OSD 时间 NET_DVR_GetPlayBackOsdTime.....	80
5.7.19	录像回放时抓图，并保存在文件中 NET_DVR_PlayBackCaptureFile.....	81
5.7.20	刷新显示回放窗口 NET_DVR_RefreshPlay.....	81
5.7.21	获取回放时用来解码显示的播放库句柄 NET_DVR_GetPlayBackPlayerIndex.....	81

下载录像文件	82
5.7.22 按时间下载录像文件 NET_DVR_GetFileByTime_V40	82
5.7.23 控制录像下载的状态 NET_DVR_PlayBackControl_V40	82
5.7.24 停止下载录像文件 NET_DVR_StopGetFile	83
5.7.25 获取当前下载录像文件的进度 NET_DVR_GetDownloadPos	83
录像文件锁定和解锁	83
5.7.26 通道模式按时间段锁定录像 NET_DVR_LockFileByTime	83
5.7.27 通道模式按时间段解锁录像 NET_DVR_UnLockFileByTime	84
5.7.28 流模式按时间段锁定录像 NET_DVR_LockStreamFileByTime	84
5.7.29 流模式按时间段解锁录像 NET_DVR_UnlockStreamFileByTime	84
录像时间段检索	85
5.7.30 启动长连接远程配置 NET_DVR_StartRemoteConfig	85
5.7.31 逐个获取查找到的信息 NET_DVR_GetNextRemoteConfig	85
5.7.32 关闭长连接 NET_DVR_StopRemoteConfig	86
录像缩略图	86
5.7.33 启动长连接远程配置 NET_DVR_StartRemoteConfig	86
5.7.34 关闭长连接 NET_DVR_StopRemoteConfig	87
5.8 图片的查找、回放及备份	88
查找图片	88
5.8.1 根据类型和时间查找图片 NET_DVR_FindPicture	88
5.8.2 逐个获取查找到的图片 NET_DVR_FindNextPicture_V40	88
5.8.3 关闭图片查找，释放资源 NET_DVR_CloseFindPicture	89
回放（下载）图片	89
5.8.4 图片回放 NET_DVR_GetPicture_V30	89
5.9 远程参数配置	89
通用参数配置	89
5.9.1 获取设备的配置信息 NET_DVR_GetDVRConfig	89
5.9.2 设置设备的配置信息 NET_DVR_SetDVRConfig	90
批量参数配置	91
5.9.3 批量获取配置信息 NET_DVR_GetDeviceConfig	91
5.9.4 批量设置配置信息 NET_DVR_SetDeviceConfig	92
长连接参数配置	93
5.9.5 启动长连接远程配置 NET_DVR_StartRemoteConfig	93
5.9.6 获取长连接配置的状态 NET_DVR_GetRemoteConfigState	94
5.9.7 关闭长连接 NET_DVR_StopRemoteConfig	95
5.10 录像回传功能	95
5.10.1 批量获取参数信息 NET_DVR_GetDeviceConfig	95
5.10.2 批量设置参数信息 NET_DVR_SetDeviceConfig	96
5.10.3 获取设备参数 NET_DVR_GetSTDConfig	97
5.10.4 设置设备参数 NET_DVR_SetSTDConfig	98
5.10.5 手动添加回传任务 NET_DVR_SetDeviceConfigEx	98
5.10.6 远程控制 NET_DVR_RemoteControl	99
5.10.7 启动长连接远程配置 NET_DVR_StartRemoteConfig	99
5.10.8 逐个获取查找到的信息 NET_DVR_GetNextRemoteConfig	100

5.10.9	关闭长连接 NET_DVR_StopRemoteConfig.....	101
5.11	设备热备管理.....	101
N+1 模式	101
5.11.1	启动长连接远程配置 NET_DVR_StartRemoteConfig	101
5.11.2	逐个获取查找到的信息 NET_DVR_GetNextRemoteConfig.....	102
5.11.3	关闭长连接 NET_DVR_StopRemoteConfig.....	103
5.11.4	获取设备的配置信息 NET_DVR_GetDVRConfig	103
5.11.5	设置设备的配置信息 NET_DVR_SetDVRConfig.....	103
5.11.6	远程控制 NET_DVR_RemoteControl	104
N+0 模式	104
5.11.7	批量获取配置信息 NET_DVR_GetDeviceConfig	104
5.12	存储设备管理.....	105
存储参数配置	105
5.12.1	获取设备的配置信息 NET_DVR_GetDVRConfig	105
5.12.2	设置设备的配置信息 NET_DVR_SetDVRConfig.....	106
5.12.3	数据文件配置及异地备份配置 NET_DVR_STDXMLConfig.....	106
5.12.4	上传文件 NET_DVR_UploadFile_V40	108
5.12.5	获取文件上传的进度和状态 NET_DVR_GetUploadState.....	108
5.12.6	获取当前数据上传的结果信息 NET_DVR_GetUploadResult	108
5.12.7	停止文件上传 NET_DVR_UploadClose.....	109
5.12.8	开始下载文件 NET_DVR_StartDownload.....	109
5.12.9	获取文件下载的进度和状态 NET_DVR_GetDownloadState.....	109
5.12.10	停止文件下载 NET_DVR_StopDownload	110
硬盘格式化	110
5.12.11	远程格式化设备硬盘 NET_DVR_FormatDisk.....	110
5.12.12	获取格式化硬盘的进度 NET_DVR_GetFormatProgress.....	110
5.12.13	关闭格式化硬盘句柄, 释放资源 NET_DVR_CloseFormatHandle	111
虚拟磁盘(逻辑卷)操作	111
5.12.14	获取虚拟磁盘列表 NET_DVR_GetVDList	111
5.12.15	创建虚拟磁盘 NET_DVR_CreateVD.....	111
5.12.16	删除虚拟磁盘 NET_DVR_DeIVD.....	111
5.13	布防、撤防.....	112
设置报警等信息上传的回调函数	112
5.13.1	注册回调函数, 接收报警消息 NET_DVR_SetDVRMessageCallBack_V30	112
布防撤防	112
5.13.2	建立报警上传通道, 获取报警等信息 NET_DVR_SetupAlarmChan_V41.....	112
5.13.3	撤销报警上传通道 NET_DVR_CloseAlarmChan_V30	113
5.14	监听报警.....	113
5.14.1	启动监听, 接收设备主动上传的报警等信息 NET_DVR_StartListen_V30	113
5.14.2	停止监听(支持多线程) NET_DVR_StopListen_V30	114
5.15	手动录像.....	114
5.15.1	远程手动启动设备录像 NET_DVR_StartManualRecord.....	114
5.15.2	远程手动停止设备录像 NET_DVR_StopManualRecord	114
5.16	设备维护管理.....	115

在线用户信息	115
5.16.1 获取设备的配置信息 NET_DVR_GetDVRConfig	115
设备在线状态	115
5.16.2 设备在线状态检测 NET_DVR_RemoteControl	115
远程升级	116
5.16.3 设置远程升级时网络环境 NET_DVR_SetNetworkEnvironment	116
5.16.4 远程升级 NET_DVR_Upgrade	116
5.16.5 获取远程升级的进度 NET_DVR_GetUpgradeProgress	116
5.16.6 获取远程升级的状态 NET_DVR_GetUpgradeState	117
5.16.7 获取远程升级的阶段信息 NET_DVR_GetUpgradeStep	117
5.16.8 关闭远程升级句柄，释放资源 NET_DVR_CloseUpgradeHandle	117
日志查找	118
5.16.9 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	118
5.16.10 逐条获取查找到的日志信息 NET_DVR_FindNextLog_V30	124
5.16.11 释放查找日志的资源 NET_DVR_FindLogClose_V30	124
恢复设备默认参数	125
5.16.12 恢复设备默认参数 NET_DVR_RestoreConfig	125
远程重启	125
5.16.13 重启设备 NET_DVR_RebootDVR	125
5.16.14 重启 CVR 服务 NET_DVR_RemoteControl	125
导入/导出配置文件	126
5.16.15 导出配置文件 NET_DVR_GetConfigFile_V30	126
5.16.16 导出配置文件 NET_DVR_GetConfigFile	126
5.16.17 导入配置文件 NET_DVR_SetConfigFile_EX	126
5.16.18 导入配置文件 NET_DVR_SetConfigFile	127
5.17 获取 IPC 协议列表	127
5.17.1 获取设备支持的 IPC 协议表 NET_DVR_GetIPCProtoList	127
6 错误代码及说明	128
6.1 网络通讯库错误码	128
6.2 RTSP 通讯库错误码	132
6.3 软解码库错误码	133
6.4 语音对讲库错误码	134

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、网络存储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。

本文档仅介绍网络存储(CVR)设备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：

DS-A9 系列：DS-A908RE-B 等

DS-A71 系列：DS-A71024R、DS-A71036R、DS-A71048R、DS-A71072R 等

DS-A72 系列：S-A72024R、DS-A72036R、DS-A72048R、DS-A72072R 等

DS-A80 系列：DS-A80216S、DS-A80624S、DS-A80636S、DS-A80648S 等

DS-A81 系列：DS-A81016S、DS-A81116S 等

DS-A82 系列：DS-A82012D、DS-A82024D、DS-A82060D 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、报警、日志管理、设备维护管理、存储管理、参数配置（系统配置、通道配置、录像配置、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、播放库等功能组件，我们提供 Windows 和 Linux 两个版本的 SDK，各自所包含的组件如下：

表 1.1 Windows SDK 组件

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

		HCGeneralCfgMgr.dll	DLL 库文件	HCNetSDKCom 文件夹
RTSP 通讯库		StreamTransClient.dll	DLL 库文件	HCNetSDKCom 文件夹
转封装库		SystemTransform.dll	DLL 库文件	HCNetSDKCom 文件夹
字符转码库		libiconv2.dll	DLL 库文件	HCNetSDKCom 文件夹
模拟能力集		LocalXml.zip	XML 文件包	
帧分析库		AnalyzeData.dll	DLL 库文件	HCNetSDKCom 文件夹
语音对讲库		AudioIntercom.dll	DLL 库文件	HCNetSDKCom 文件夹
		OpenAL32.dll	DLL 库文件	HCNetSDKCom 文件夹
播放库	核心库文件	PlayM4.h、WindowsPlayM4.h	头文件	
		PlayCtrl.lib	LIB 库文件	
		PlayCtrl.dll	DLL 库文件	
	视频渲染库	SuperRender.dll	DLL 库文件	
	音频渲染库	AudioRender.dll	DLL 库文件	
	小鹰眼库	EagleEyeRender.dll	DLL 库文件	
	GPU 硬解码库	HWDecode.dll	DLL 库文件	
	鱼眼库	MP_Render.dll	DLL 库文件	
	视频后处理库	MP_VIE.dll	DLL 库文件	
	测温信息抓图库	YUVProcess.dll	DLL 库文件	
	DirectX 组件库	D3DCompiler_43.dll	DLL 库文件	

表 1.2 Linux SDK 组件

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

字符转码库		libiconv2.so	SO 库文件	HCNetSDKCom 文件夹
帧分析库		libanalyzedata.so	SO 库文件	HCNetSDKCom 文件夹
播放库	核心库文件	PlayM4.h、LinuxPlayM4.h	头文件	
		libPlayCtrl.so	SO 库文件	
	视频渲染库	libSuperRender.so	SO 库文件	
	音频渲染库	libAudioRender.so	SO 库文件	

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

- **网络通讯库：**设备网络 SDK 的主体，主要用于网络客户端与各类产品之间的通讯交互，负责远程功能调控，远程参数配置及码流数据的获取和处理等。设备网络 SDK V5.0 针对产品应用业务进行细化，对之前版本的 SDK 的功能模块进行组件化，其中外部接口（HCNetSDK.dll）仍然保持和设备网络 SDK V4.x 版本保存一致（向下兼容），其他单独的业务功能（预览、回放等）可以加载单独的模块组件，多个业务功能也可以组合使用。**更新 SDK 时，HCNetSDK.dll、HCCore.dll 以及 HCNetSDKCom 文件夹下的功能组件库文件都需要更新加载，且 HCNetSDKCom 文件夹名不能修改。**
- **hpr 库：**网络通讯库的依赖库，Linux SDK 使用时和网络通讯库同时加载。
- **RTSP 通讯库：**支持 RTSP 传输协议的网络库。当需要对支持 RTSP 协议的产品进行取流等操作时必须加载该项组件。
- **转封装库：**库的功能可以分为两种：一种是将标准码流转换成采用我们公司封装格式的码流。当用户需要对支持 RTSP 协议的产品捕获采用本公司封装格式的码流数据时（即当设置 NET_DVR_RealPlay_V40 接口中的回调函数捕获数据或者调用 NET_DVR_SetRealDataCallBack 接口捕获数据时）必须加载该组件。另一种功能是能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。CVR 暂不支持语音对讲功能。**
- **字符转换库：**电脑字符集和设备字符集不一致时，SDK 内部需要进行字符编码转换，SDK 默认使用 libiconv 库进行类型转换。如果用户不想使用 libiconv 编码库，可以调用 NET_DVR_SetSDKLocalCfg (类型: NET_SDK_LOCAL_CFG_TYPE_BYTE_ENCODE) 设置字符转码回调函数，将用户自己的字符编码接口告知 SDK，然后 SDK 将使用用户提供的字符编码接口进行字符串处理。
- **模拟能力集：**如果需要获取设备能力集（NET_DVR_GetDeviceAbility），建议调用 NET_DVR_SetSDKLocalCfg 启用模拟能力集，此时需要加载 LocalXml.zip（要求和网络通讯库放在同一个目录下）。
- **帧分析库：**用于分析视音频帧数据，调用 NET_DVR_SetESRealPlayCallBack、NET_DVR_SetPlayBackESCallBack 设置裸码流回调函数等接口时，必须加载该库文件。

- **播放库：**主要用于对实时码流数据进行解码显示（实现预览功能）和对录像文件进行回放解码等。用户如果需要在 SDK 内部进行对实时流和录像码流播放显示时（即 NET_DVR_RealPlay_V40 接口的第二个结构体参数的播放句柄设置成有效句柄时）必须加载该组件，而如果用户仅需要用网络通讯库捕获到数据后再外部自行处理就不需要加载该组件，这种情况下用户在外部分自行解码将更灵活，可参见软解码库函数说明《播放器 SDK 编程指南》。

2 SDK 版本更新

Version V5.2.7.30 (build20170428)

- CVR V2.3.2-7
- 新增硬盘卷配置功能（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
命令：NET_DVR_GET_HDVOLUME_CFG、NET_DVR_SET_HDVOLUME_CFG
- 存储总能力集 RacmCap 更新，对应接口：[NET_DVR_GetSTDAbility](#)，能力集类型：
NET_DVR_GET_RACM_CAP。新增能力节点<isSupportFileData>（文件相关配置能力）、
<isSupportRemoteBackup>（异地备份相关配置能力）。
- 新增数据文件相关配置及异地备份相关配置功能（对应接口：[NET_DVR_STDXMLConfig](#)）
- 新增上传和下载文件卷数据文件相关功能。对应接口：[NET_DVR_UploadFile_V40](#)、
[NET_DVR_GetUploadState](#)、[NET_DVR_GetUploadResult](#)、[NET_DVR_UploadClose](#)、[NET_DVR_StartDownload](#)、
[NET_DVR_GetDownloadState](#)、[NET_DVR_StopDownload](#)

Version V5.2.5.30 (build20161009)

- CVR V2.3.2-5
- 新增一键配置 CVR V50 扩展命令（对应接口：[NET_DVR_StartRemoteConfig](#)）：
报警信息类型：NET_DVR_ONE_KEY_CONFIG_V50。
- 一键配置 CVR 对应的 [NET_DVR_GetRemoteConfigState](#) 接口返回的状态值新增：1101(热备盘个数过多)、
1102(创建图片卷失败)。
- 新增获取录像缩略图的功能（对应接口：[NET_DVR_StartRemoteConfig](#)）：
命令：NET_DVR_GET_THUMBNAIIS、NET_DVR_GET_FIGURE。
- CVR 设备能力集(CVRAbility，对应接口：[NET_DVR_GetDeviceAbility](#)，能力集类型：DEVICE_ABILITY_INFO)
扩展，新增节点：<OneKeyConfigCVRV50>(一键配置 CVR V50 扩展)。
- 设备软硬件能力集(BasicCapability，对应接口：[NET_DVR_GetDeviceAbility](#)，能力集类型：
DEVICE_SOFTHARDWARE_ABILITY)扩展，新增节点：<isSupportGetFigure>(是否支持按时间点获取缩略图)、
<isSupportGetThumbnails>(是否支持按时间段获取缩略图)。

Version V5.2.3.20 (build20160722)

- CVR V2.3.2-4
- 新增 RACM 存储总能力集（对应接口：[NET_DVR_GetSTDAbility](#)）：
能力集类型：NET_DVR_GET_RACM_CAP。
- 新增回传功能基础配置能力集（对应接口：[NET_DVR_GetSTDAbility](#)）：
能力集类型：NET_DVR_GET_RECORD_PASSBACK_BASIC_CFG_CAP。
- 新增回传功能基础配置功能（对应接口：[NET_DVR_GetSTDConfig](#) 和 [NET_DVR_SetSTDConfig](#)）：
命令：NET_DVR_GET_RECORD_PASSBACK_BASIC_CFG、NET_DVR_SET_RECORD_PASSBACK_BASIC_CFG。
- 新增返回任务 ID 的手动添加录像回传任务功能（对应接口：[NET_DVR_SetDeviceConfigEx](#)）：
命令：NET_DVR_ADD_RECORD_PASSBACK_TASK_MANUAL_V50。
- 新增回传历史录像计划能力集（对应接口：[NET_DVR_GetSTDAbility](#)）：
能力集类型：NET_DVR_GET_RECORD_PASSBACK_HISTORY_PLAN_CFG_CAP。
- 新增回传历史录像计划配置功能（对应接口：[NET_DVR_GetSTDConfig](#) 和 [NET_DVR_SetSTDConfig](#)）：
命令：NET_DVR_GET_RECORD_PASSBACK_HISTORY_PLAN_CFG、

NET_DVR_SET_RECORD_PASSBACK_HISTORY_PLAN_CFG。

- NET_DVR_RECORD_PASSBACK_MANUAL_COND(手动添加录像回传任务查询条件参数)扩展, 使用 21 个保留字节新增参数: byTimeSegmentQuety(是否启用时间范围查询)、struBeginTime(查询开始时间)、struEndTime(查询结束时间)、dwTaskID(任务 ID), 参数 byType(查询方式)新增取值: 4-根据任务 ID 获取任务。
- CVR 设备能力集(CVRAbility, 对应接口: [NET DVR GetDeviceAbility](#), 能力集类型: DEVICE_ABILITY_INFO)扩展: <GetAllRecordPassBackManulTask>(获取所有手动录像回传任务能力)中新增子节点: <timeSegmentQuety>(是否启用时间范围查询), 子节点<type>(查询方式)新增取值: 4-根据任务 ID 获取任务。

Version V5.1.6.5 (build20160113)

- CVR V2.3.2-3
- 新增录像完整性检测功能 (对应接口: [NET DVR StartRemoteConfig](#) 和 [NET DVR GetNextRemoteConfig](#)):
命令: NET_DVR_RECORD_CHECK。
- 新增获取录像段总大小功能 (对应接口: [NET DVR StartRemoteConfig](#) 和 [NET DVR GetNextRemoteConfig](#)):
命令: NET_DVR_GET_RECORD_SEGMENT_CFG。
- 新增手动添加录像回传任务功能 (对应接口: [NET DVR SetDeviceConfig](#)):
命令: NET_DVR_ADD_RECORD_PASSBACK_TASK_MANUAL。
- 新增删除手动录像回传任务功能 (对应接口: [NET DVR SetDeviceConfig](#)):
命令: NET_DVR_DEL_RECORD_PASSBACK_TASK_MANUAL。
- 新增获取所有录像回传手动任务功能 (对应接口: [NET DVR StartRemoteConfig](#) 和 [NET DVR GetNextRemoteConfig](#)):
命令: NET_DVR_GET_ALL_RECORD_PASSBACK_TASK_MANUAL。
- 新增控制录像回传手动任务功能 (对应接口: [NET DVR RemoteControl](#)):
命令: NET_DVR_RECORD_PASSBACK_TASK_MANUAL_CTRL。
- 新增录像回传计划配置功能 (对应接口: [NET DVR GetDeviceConfig](#) 和 [NET DVR SetDeviceConfig](#)):
命令: NET_DVR_GET_RECORD_PASSBACK_PLAN_CFG、NET_DVR_SET_RECORD_PASSBACK_PLAN_CFG。
- 新增获取设备存储信息功能 (对应接口: [NET DVR GetDVRConfig](#)):
命令: NET_DVR_GET_DEV_STORAGE_CFG。
- 新增获取在线用户信息功能 (对应接口: [NET DVR GetDVRConfig](#)):
命令: NET_DVR_GET_ONLINE_USER_CFG。
- NET_DVR_DEVICE_RUN_STATUS(设备运行状态信息)使用 4 个保留字节新增参数: fNetworkFlow(网络流量)。
- 设备通用能力集(DEVICE_ABILITY_INFO), 其中 CVR 设备能力集(CVRAbility)扩展, 新增节点:
<RecordCheck>(是否支持录像检测)、<AddRecordPassBackManul>(是否支持手动添加录像回传任务)、<GetAllRecordPassBackManulTask>(是否支持获取所有录像回传手动任务)、<RecordPassBackManulTaskCtrl>(是否支持控制录像回传手动任务)、<RecordPassBackManulTaskDel>(是否支持删除录像回传手动任务)、<RecordPassBackPlanConfig>(是否支持录像回传计划配置)、<GetSegmentRecordtotalSize>(是否支持获取录像段总大小)、<DeviceStorageStatus>(是否支持获取设备存储信息)、<OnlineUser>(是否支持获取在线用户)。

Version V5.1.5.5 (build20151111)

- CVR V2.3.2

- 新增获取 N+0 模式下重定向设备信息功能（对应接口：[NET_DVR_GetDeviceConfig](#)）：
命令：NET_DVR_GET_RELOCATE_INFO。
- 新增能力集（对应接口：[NET_DVR_GetDeviceAbility](#)，对应能力集类型：DEVICE_ABILITY_INFO）：
CVR 设备能力集 CVRAbility。
- 新增支持图片查找接口：[NET_DVR_FindNextPicture_V40](#)。
- NET_DVR_SEARCH_EVENT_PARAM_V40(事件录像查询条件)和 NET_DVR_SEARCH_EVENT_RET_V40(事件录像查询结果)中 wMinorType 新增主类型为 EVENT_STREAM_INFO(流 ID 信息)对应的次类型：9~32。
- 新增错误码：824、825。

Version V5.1.1.16 (build20150528)

- CVR Mars V2.3.1
- 新增智能搜索接口扩展，支持流 ID 方式查询：
[NET_DVR_SmartSearch_V40](#)。
- 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：
<SmartSearch>(智能搜索能力)。

Version V4.3.0.6 (build20140722)

- CVR V2.3.1
- 流的来源信息配置扩展（对应接口：[NET_DVR_GetDeviceConfig](#) 和：[NET_DVR_SetDeviceConfig](#)）：
命令：NET_DVR_GET_STREAM_SRC_INFO_V40、NET_DVR_SET_STREAM_SRC_INFO_V40。
- 新增回放抽帧功能（对应接口：[NET_DVR_PlayBackControl_V40](#)）：
命令：NET_DVR_START_DRAWFRAME、NET_DVR_STOP_DRAWFRAME。
- 新增 CVR 设备报警信息扩展类型（对应接口：[NET_DVR_SetDVRMessageCallBack_V30](#) 和
[NET_DVR_StartListen_V30](#)）：COMM_ALARM_DEVICE_V40。
- 录像类型扩展：NET_DVR_FILECOND_V40(文件查找条件)的 dwFileType、NET_DVR_FINDDATA_V40(文件查找结果信息)的 byFileType 均增加类型取值：21-信号量、22-回传、23-回迁录像、24-遮挡。
- NET_DVR_RECORD_V30(通道录像参数)使用 2 个保留字节新增参数：bySVCLLevel(SVC 抽帧类型)、byRecordManage(录像调度)。
- NET_DVR_SETUPALARM_PARAM(报警布防参数)使用 1 个保留字节新增参数：byRetDevInfoVersion(CVR 上传报警信息回调版本)。
- 设备流能力(STREAM_ABILITY)扩展：
 - 1) 新增节点：<RTSPProMode>(RTSP 取流模式)、<URLCascadeMode>(URL 级联取流模式)。
 - 2) <DirectStreamMode>(直接取流模式)新增子节点：<LocalBackUp>(是否支持本地备份)、<supportChanNo>(是否支持 VAG 取流)。
 - 3) <StreamMediaMode>(流媒体取流模式)新增子节点：<localBackUp>(是否支持本地备份)。
 - 4) <getStreamMode>(支持的取流模式)取值新增：7-通过标准 RTSP 接入设备，8-支持 URL 级联方式到前端取流。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中录像相关能力(RecordAbility)，新增节点：
<drawFrameInPlayBack>(是否支持回放抽帧)。
- 设备通用能力集(DEVICE_ABILITY_INFO)，其中通道输入能力(ChannellInputAbility)，<recordPlanAbility>(录像计划能力)新增取值：recordManage(录像调度)。

Version V4.2.2.6 (build20130618)

- CVR V2.2.0

- 新增 N+1 模式备机个数配置功能（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_GET_MAX_MACHINE_NUM_CFG、NET_DVR_SET_MAX_MACHINE_NUM_CFG。
- 新增 N+1 设备信息查询功能（对应接口：[NET_DVR_StartRemoteConfig](#)）：
配置命令：NET_DVR_GET_N_PLUS_ONE_DEVICE_INFO。
- 新增 N+1 设备添加删除功能（对应接口：[NET_DVR_RemoteControl](#)）：
控制命令：NET_DVR_ADD_DEVICE、NET_DVR_DEL_DEVICE。
- 新增数据回迁状态配置功能（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_GET_DATA_CALLBACK_CFG、NET_DVR_SET_DATA_CALLBACK_CFG。
- 新增用户权限配置功能（对应接口：[NET_DVR_GetDeviceConfig](#) 和 [NET_DVR_SetDeviceConfig](#)）：
NET_DVR_GET_USER_RIGHT_CFG、NET_DVR_SET_USER_RIGHT_CFG。
- 新增 CVR 一键配置功能（对应接口：[NET_DVR_StartRemoteConfig](#)）：
配置命令：NET_DVR_ONE_KEY_CONFIG。
- 新增重启 CVR 服务功能（对应接口：[NET_DVR_RemoteControl](#)）：
控制命令：NET_DVR_RESTART_SERVICE。
- 新增上传报警信息（对应接口：[NET_DVR_SetDVRMessageCallBack_V30](#) 和 [NET_DVR_StartListen_V30](#)）：
NET_DVR_CVR_ALARM。
- NET_DVR_RECORD_V30 使用 5 个保留字节新增参数：byStreamType(码流类型)、byPassbackRecord(是否回传录像)、wLockDuration(录像锁定时长)、byRecordBackup(录像是否存档)。
- NET_DVR_TIME_LOCK、NET_DVR_STREAM_TIME_LOCK 使用 8 个保留字节新增参数：strUnlockTimePoint(解锁时间)。
- NET_DVR_SEARCH_EVENT_PARAM、NET_DVR_SEARCH_EVENT_RET 中增加主类型为 EVENT_STREAM_INFO(流 ID 信息)时的次类型取流。
- NET_DVR_SEARCH_EVENT_PARAM、NET_DVR_SEARCH_EVENT_RET 中增加主类型为 EVENT_STREAM_INFO(流 ID 信息)时的次类型(wMinorType)取流。
- NET_DVR_SEARCH_EVENT_PARAM 中结构 struStreamIDParam 使用 1 个保留字节新增参数：
byBackupVolumeNum(存档卷号)。
- NET_DVR_SEARCH_EVENT_RET 中结构 struStreamIDParam 使用 4 个保留字节新增参数：dwFileIndex(存档卷上的文件索引)。
- NET_DVR_STREAM_RECORD_STATUS、NET_DVR_CHANS_RECORD_STATUS 使用 1 个保留字节新增参数：
byOffLineRecord(断网录像使能)，其参数 byRecord 增加类型取值：2-空闲，3-无连接，4-无输入视频，5-未加载，6-存档中，7-回传中，8-用户名或密码错，9-未验证，10-存档中和录像中，11-录像回传中和录像中。

Version V4.1.4

- 新增对录像段加锁接口：
[NET_DVR_LockFileByTime](#)、[NET_DVR_UnLockFileByTime](#)。
- 新增按流 ID 对录像段加锁接口：
[NET_DVR_LockStreamFileByTime](#)、[NET_DVR_UnlockStreamFileByTime](#)。
- 新增通道录像状态信息配置功能（对应接口：[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）：
NET_DVR_GET_RECORD_CHANNEL_INFO。
- 新增按流 ID 批量配置功能（对应接口：[NET_DVR_GetDeviceConfig](#) 和 [NET_DVR_SetDeviceConfig](#)）：
NET_DVR_GET_STREAM_RECORD_STATUS、NET_DVR_SET_STREAM_RECORD_STATUS、
NET_DVR_GET_STREAM_RECORD_INFO、NET_DVR_SET_STREAM_RECORD_INFO、
NET_DVR_GET_STREAM_SRC_INFO、NET_DVR_GET_STREAM_SRC_INFO。

- 录像回放接口扩展，新增按流 ID 回放：
[NET_DVR_PlayBackByTime_V40](#)。
- 新增按流 ID 开启设备手动录像的接口：
[NET_DVR_StartManualRecord](#)、[NET_DVR_StopManualRecord](#)。
- 新增按流 ID 查找录像功能：
NET_DVR_SEARCH_EVENT_PARAM、NET_DVR_SEARCH_EVENT_RET。
- NET_DVR_IPCHANINFO 使用 1 个保留字节增加参数 byIPIDHigh。
- NET_DVR_SINGLE_HD 使用 2 个保留字节增加参数 byDiskDriver 和 byRecycling。
- NET_DVR_SEARCH_EVENT_RET 联合体中增加结构体 struStreamIDRet。

3 函数调用顺序

3.1 SDK 基本调用的主要流程

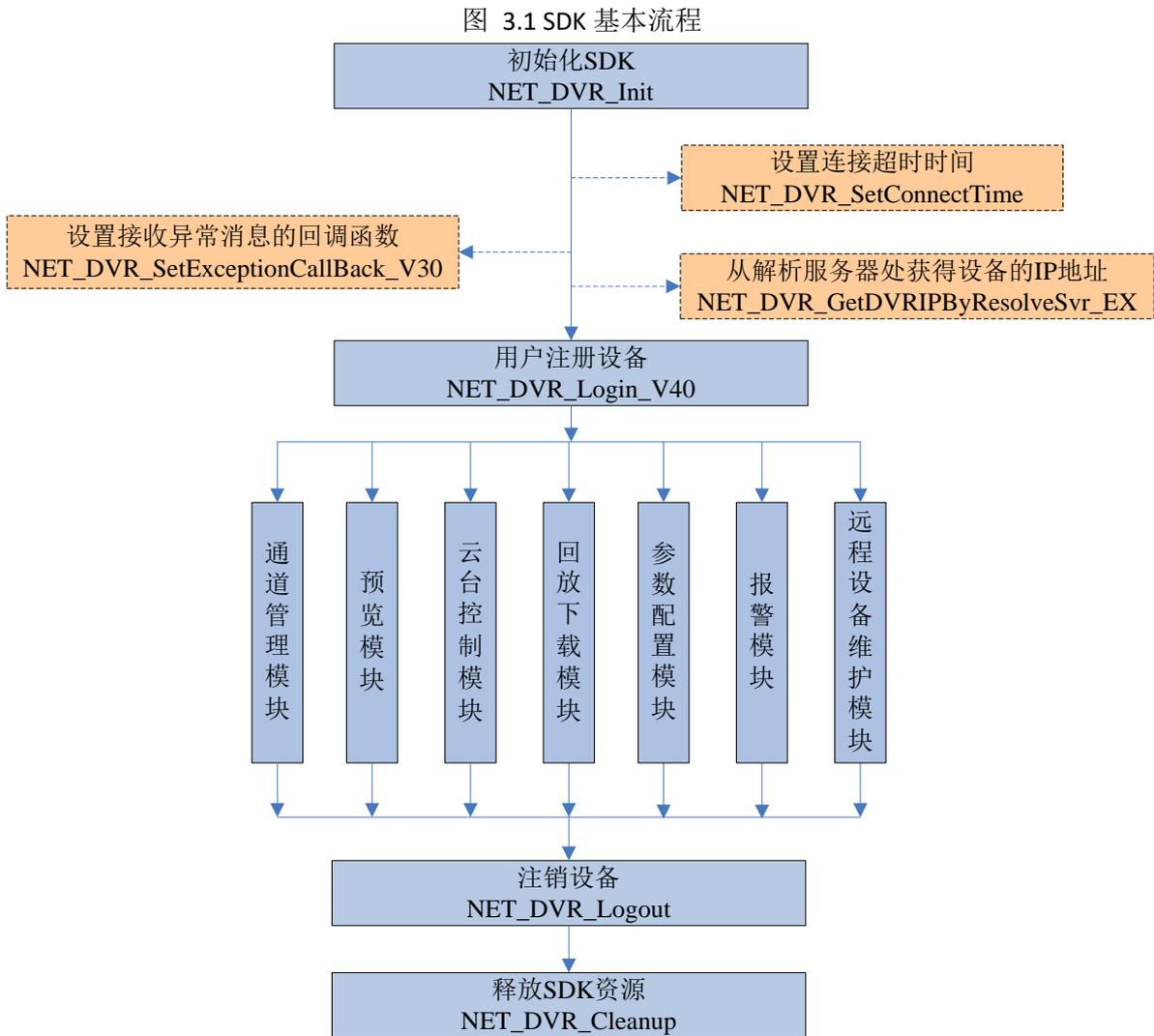


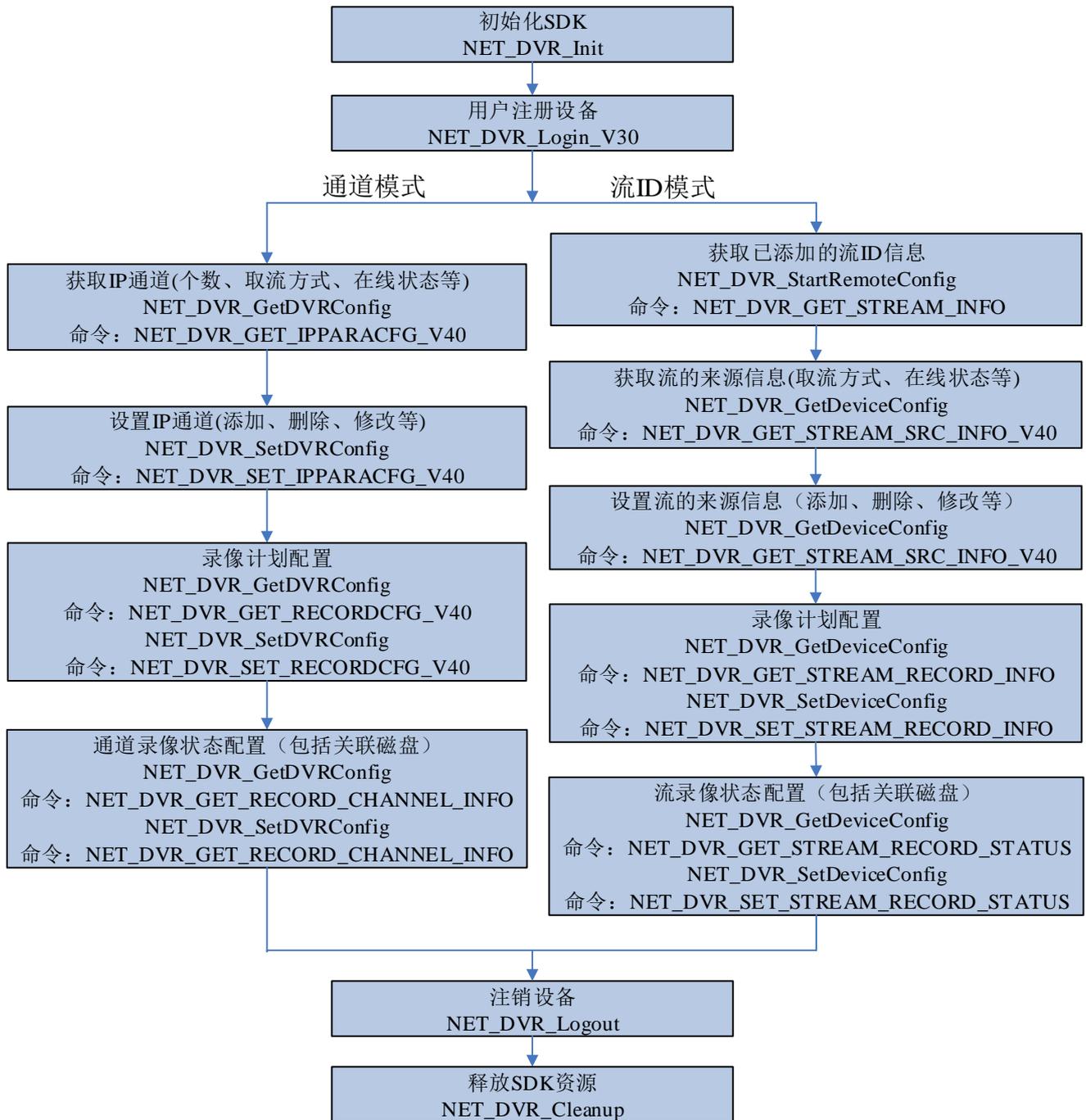
图 3.1 中虚线框的流程是可选部分，不会影响其他流程和模块的功能使用。按实现功能的不同可以分成十个模块，实现每个模块的功能时初始化 SDK、用户注册设备、注销设备和释放 SDK 资源这 4 个流程是必不可少的。

- 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。
- 设置接收异常消息的回调函数 ([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 V40](#)）：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。CVR 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。
- 通道管理模块：包括通道添加、删除、修改，录像计划配置，流状态等。CVR 的通道管理分通道模式和流 ID 模式，具体流程详见[通道管理模块流程](#)。
- 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。具体流程详见[预览模块流程](#)。
- 云台控制模块：实现对云台的基本操作、预置点、巡航、花样扫描的控制。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。具体流程详见[云台控制模块流程](#)。
- 回放和下载模块：可以通过按时间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。具体流程详见[回放和下载模块流程](#)。
- 通用参数配置模块：设置和获取设备的参数，主要包括设备参数、网络参数、用户参数、硬盘管理参数等参数信息。具体流程详见[参数配置模块流程](#)。
- 远程设备维护模块：实现重启设备、恢复默认值、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](#)。
- 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](#)。

3.2 通道管理模块流程



CVR 添加编码器，其通道管理分通道模式和流 ID 模式。

- 通道模式：类似网络硬盘录像机的 IP 通道，使用通道号来标识设备添加的编码器。
 - 调用接口 [NET_DVR_GetDVRConfig](#)（命令：NET_DVR_GET_IPPARACFG_V40）可以获取 IP 通道资源信息（NET_DVR_IPPARACFG_V40），包括 IP 通道个数（dwDChanNum）、IP 通道起始通道号（dwStartDChan）、IP 通道取流模式、IP 通道前端接的编码设备信息、IP 通道在线状态等。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。通道相关参数获取和设置，需要按组循环调用接口，每组 64

个通道。

- 通道模式下，分别通过 [NET_DVR_GetDVRConfig](#)（命令：NET_DVR_GET_RECORDCFG_V40）和 [NET_DVR_SetDVRConfig](#)（命令：NET_DVR_SET_RECORDCFG_V40）获取和设置录像计划。
- 通道录像状态（包括不在录像、在录像、空闲、无连接等）和关联磁盘号（即录像卷）通过接口 [NET_DVR_GetDVRConfig](#)（命令：NET_DVR_GET_RECORD_CHANNEL_INFO）获取。添加通道后需要配置关联录像卷，通过接口 [NET_DVR_SetDVRConfig](#)（命令：NET_DVR_SET_RECORD_CHANNEL_INFO）进行设置。
- 通过 IE 直接登录 CVR 设备，配置添加编码器，其编码器名称设置 1~1024 的纯数字，该方法配置也属于通道模式，此处的数字值就是接口里面的通道号。



- 流 ID 模式：通过流 ID 标识 CVR 添加的编码器，流 ID 可以为字母、数字和"_"的组合。
 - 通过 [NET_DVR_StartRemoteConfig](#)（命令：NET_DVR_GET_STREAM_INFO）可以查找设备上已添加流 ID 信息，查找到的流 ID 结果在该接口设置的回调函数返回。查找到流 ID 之后，可以调用 [NET_DVR_GetDeviceConfig](#)（命令：NET_DVR_GET_STREAM_SRC_INFO_V40）获取对应的流来源信息，包括取流方式和前端编码设备信息。调用 [NET_DVR_SetDeviceConfig](#)（命令：NET_DVR_SET_STREAM_SRC_INFO_V40）可以添加、修改、删除流 ID 信息。通过 IE 直接添加编码器不是流 ID 模式，通过相关接口获取不到流 ID 和来源信息，但是通过流 ID 方式添加的通道，IE 登录设备可以查看到相关编码器信息。
 - 流 ID 模式下，分别通过 [NET_DVR_GetDeviceConfig](#)（命令：NET_DVR_GET_STREAM_RECORD_INFO）和 [NET_DVR_SetDeviceConfig](#)（命令：NET_DVR_SET_STREAM_RECORD_INFO）获取和设置录像计划。
 - 编码器状态信息（包括不在录像、在录像、空闲、无连接等）和关联磁盘号（即录像卷）通过接口 [NET_DVR_GetDeviceConfig](#)（命令：NET_DVR_GET_STREAM_RECORD_STATUS）获取。添加编码器后需要配置关联录像卷，通过接口 [NET_DVR_SetDeviceConfig](#)（命令：NET_DVR_SET_STREAM_RECORD_STATUS）进行设置。
 - 目前流 ID 模式下不支持预览和云台控制功能。
- 流 ID 模式的相关接口兼容通道模式，即通道模式的通道信息也可以通过流 ID 模式的相关接口进行获取和设置。

[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

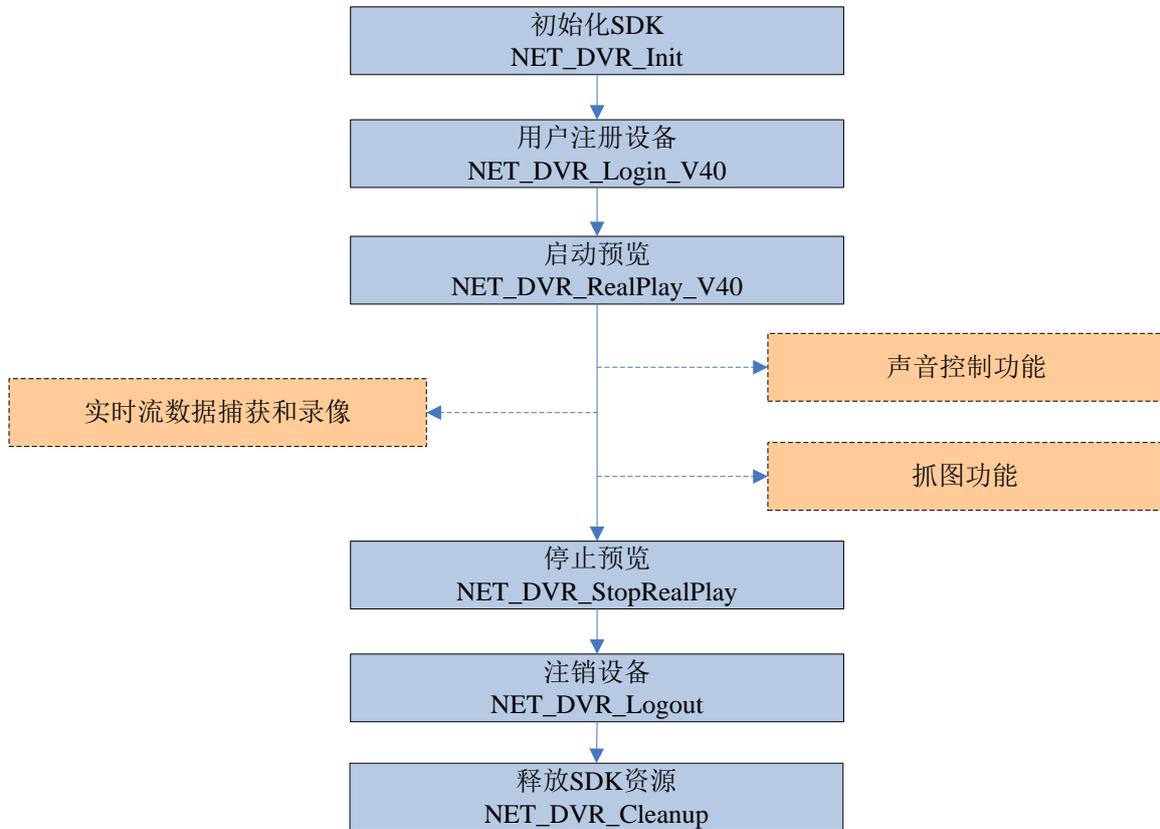


图 3.2 中虚线框部分的模块是与预览模块相关，必须在启动预览后才能调用，这些模块之间是并列的关系，各自完成相应的功能。

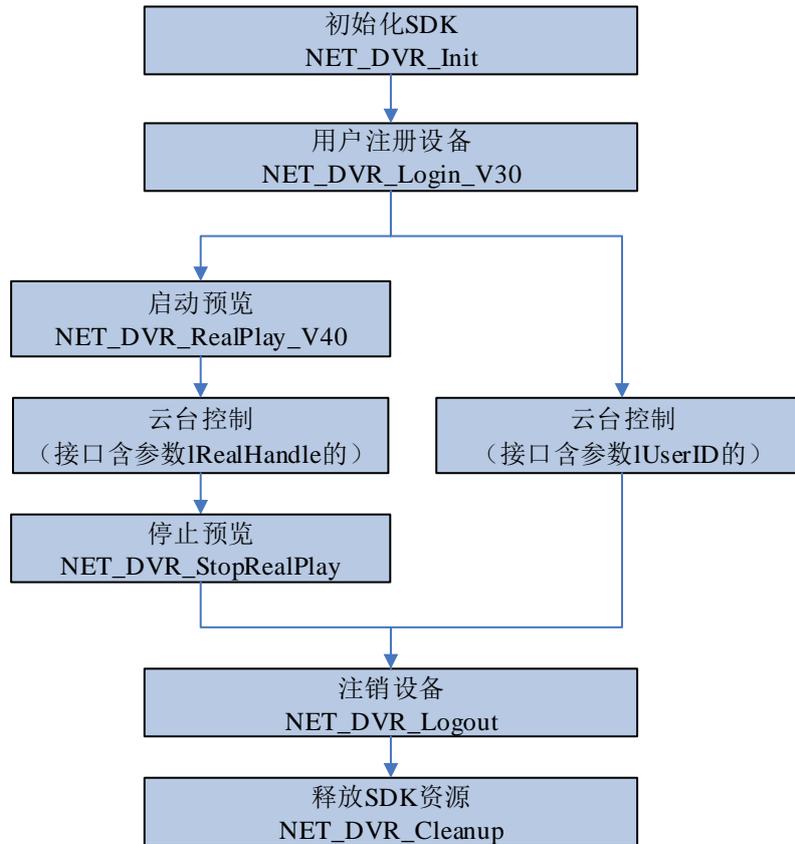
- 声音控制功能主要实现独占、共享声音的打开和关闭；音量的控制。相关接口有：[NET_DVR_OpenSound](#)、[NET_DVR_CloseSound](#)、[NET_DVR_OpenSoundShare](#)、[NET_DVR_CloseSoundShare](#)、[NET_DVR_Volume](#) 等。
- 实时流数据捕获和录像模块主要实现实时流数据回调和本地录像的功能，可以供用户后续处理。相关接口有：[NET_DVR_SetRealDataCallBack](#)、[NET_DVR_SetStandardDataCallBack](#)、[NET_DVR_SaveRealData](#) 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](#)。

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](#)），获取码流数据进行后续解码播放处理。

[调用实例代码](#)

3.4 云台控制模块流程

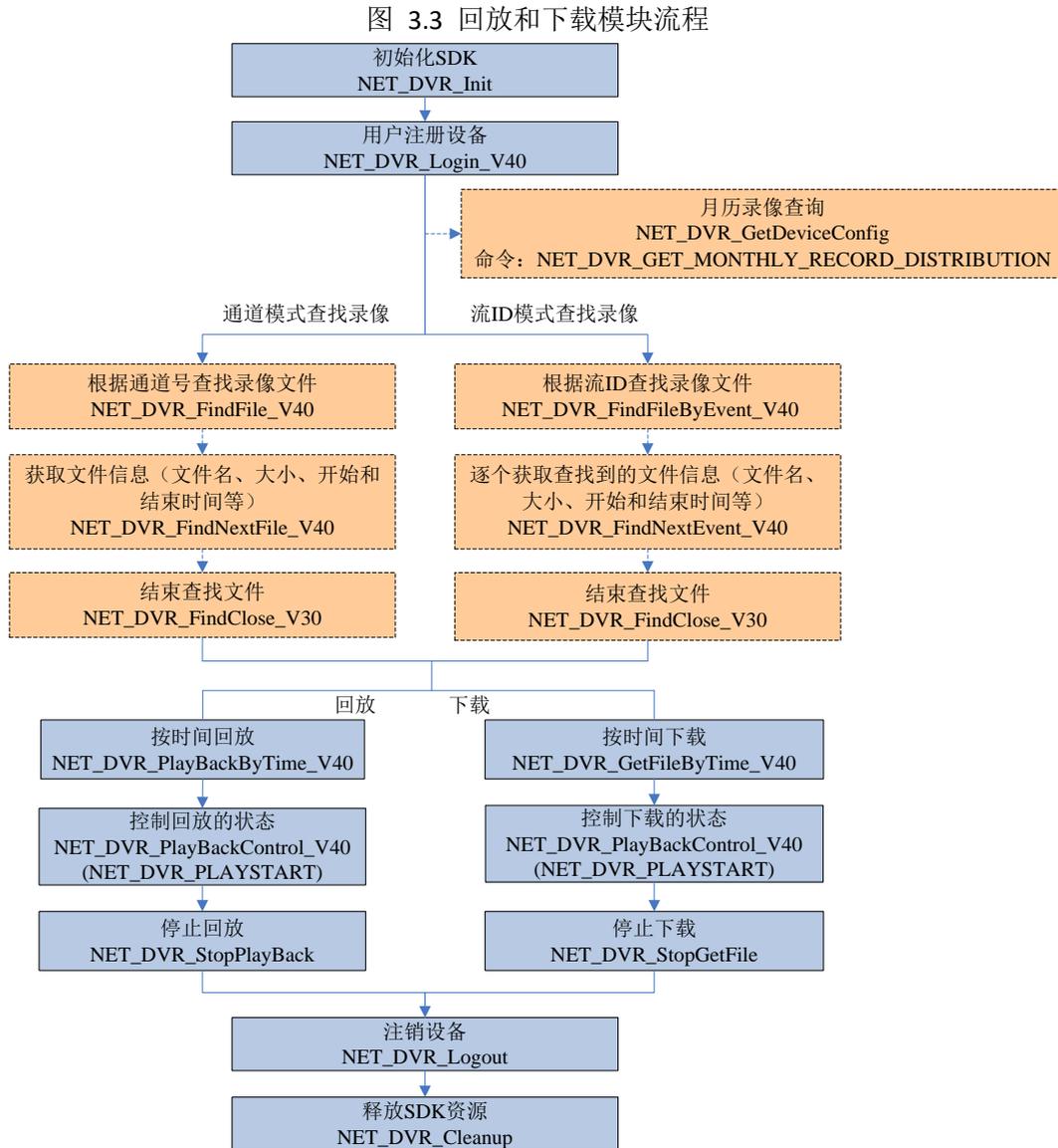


实现对云台的基本操作、预置点、巡航、花样扫描的控制。SDK 将云台控制分为两种模式：一种是通过图像预览返回的句柄进行控制；另一种是无预览限制，通过用户注册 ID 号进行云台控制。

- 在开启预览的前提下实现对云台控制的操作功能，预览接口返回的句柄作为云台控制接口的参数。相关接口：[NET_DVR_PTZControl](#)、[NET_DVR_PTZControlWithSpeed](#)、[NET_DVR_PTZPreset](#)、[NET_DVR_PTZCruise](#)、[NET_DVR_PTZTrack](#)。
- 不需要预览，注册登录设备之后即可以进行云台控制，登录接口返回的句柄作为云台控制接口的参数。相关接口：[NET_DVR_PTZControl Other](#)、[NET_DVR_PTZControlWithSpeed Other](#)、[NET_DVR_PTZPreset Other](#)、[NET_DVR_PTZCruise Other](#)、[NET_DVR_PTZTrack Other](#)。

[调用实例代码](#)

3.5 回放和下载模块流程



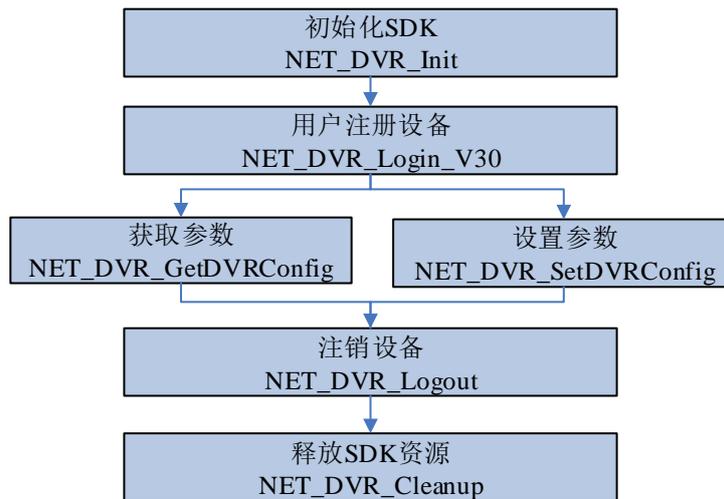
- 通过 [NET_DVR_GetDeviceConfig](#)（命令：NET_DVR_GET_MONTHLY_RECORD_DISTRIBUTION）可以查询流 ID 或者通道的指定年月的录像分布情况，比如 2015 年 4 月份的 1~30 日哪些天有录像文件。
- 详细的录像文件信息，通道模式和流 ID 模式下，分别使用不同的接口进行查找：
 - 通道模式下，根据通道号查找录像，相关接口：[NET_DVR_FindFile_V40](#)、[NET_DVR_FindNextFile_V40](#)、[NET_DVR_FindClose_V30](#)。查找结果包括录像开始和结束时间、录像大小、录像类型等。
 - 流 ID 模式下，根据流 ID 查找录像，相关接口：[NET_DVR_FindFileByEvent_V40](#)、[NET_DVR_FindNextEvent_V40](#)、[NET_DVR_FindClose_V30](#)。查找结果包括录像开始和结束时间、录像大小、录像类型等。
- CVR 设备只支持按时间回放或下载，调用回放或下载接口（[NET_DVR_PlayBackByTime_V40](#)、[NET_DVR_GetFileByTime_V40](#)）指定回放条件包括开始和结束时间、流 ID 或者通道号等，然后调用控制接口（[NET_DVR_PlayBackControl_V40](#)）的开始播放命令（NET_DVR_PLAYSTART）开始回放或下载。

- 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](#)

3.6 参数配置模块流程

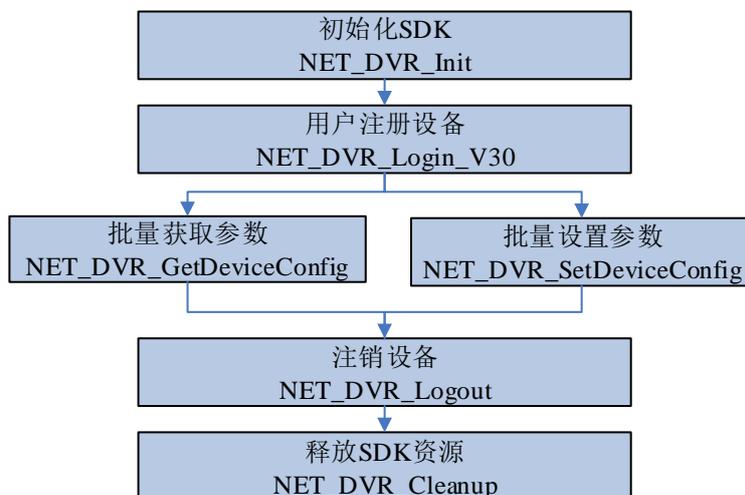
图 3.4 参数配置模块流程



通用参数配置，包括设备参数、网络参数、用户参数、时间参数等，不同的配置命令（ICommand）对应不同的参数配置功能。

- 需要先初始化 SDK 和用户注册登录设备，将用户注册接口返回的 ID 号作为配置接口的参数。
- 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。

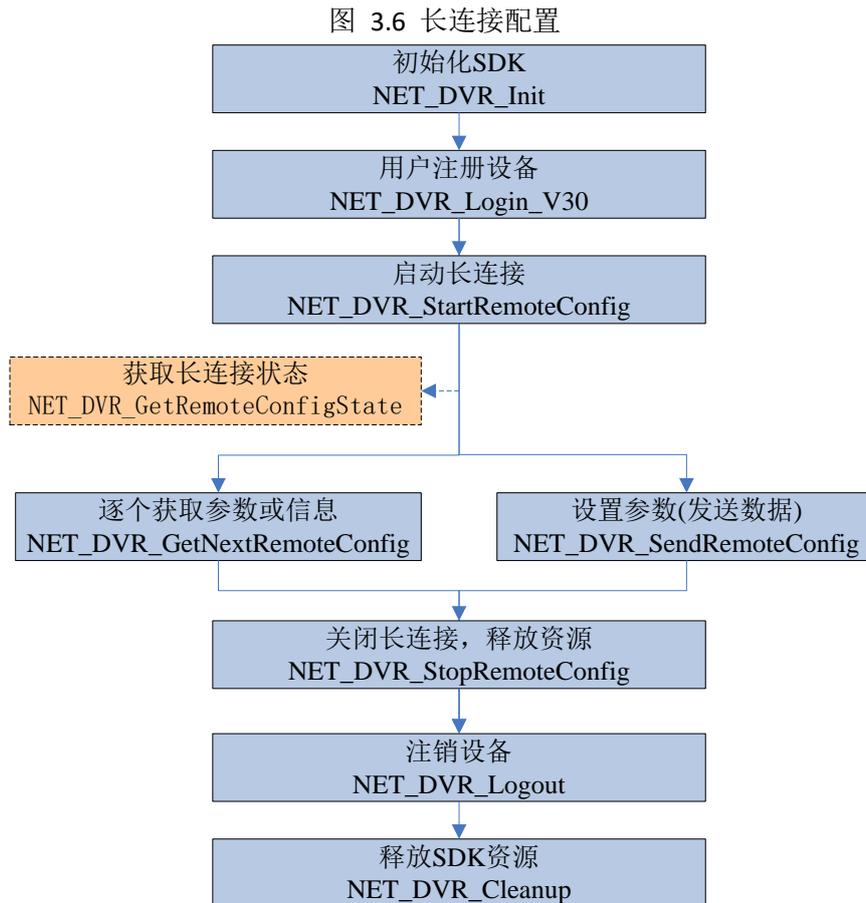
图 3.5 批量参数配置



批量参数配置，包括流来源、录像计划和状态、用户权限等参数配置，该接口支持批量功能。不同的

配置命令（ICommand）对应不同的参数配置功能。

- 实现参数配置需要先初始化 SDK 和用户注册登录设备，将用户注册接口返回的 ID 号作为配置接口的参数。
- 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_GetDeviceConfig](#)）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_SetDeviceConfig](#)），返回成功即设置成功。



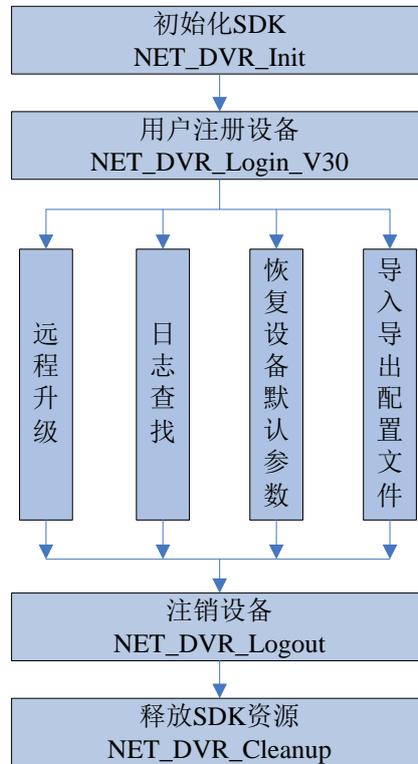
长连接配置，包括一键配置 CVR、流 ID 信息查询、N+1 设备查询等功能，相关配置耗时长，需要长连接来实现。不同的配置命令（ICommand）对应不同的参数配置功能。

- 先调用 [NET_DVR_StartRemoteConfig](#) 建立长连接，查找的结果通过回调函数返回或者调用 [NET_DVR_GetNextRemoteConfig](#) 逐个获取结果（不同的命令方法不一样，具体参考接口说明）。查找结束后调用 [NET_DVR_StopRemoteConfig](#) 断开长连接，释放资源。

[调用实例代码](#)

3.7 远程设备维护模块流程

图 3.7 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。

- 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。

[调用实例代码](#)

3.8 报警模块流程

3.8.1 报警（布防）流程

图 3.8 报警布防流程

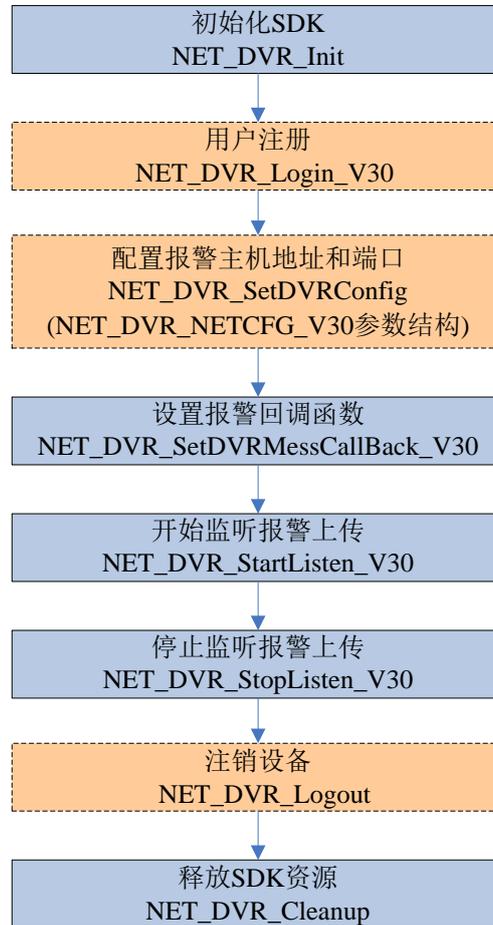


- “布防”报警方式：是指 SDK 主动连接设备，并发起报警上传命令，设备发生报警立即发送给 SDK。
- “布防”方式需要先进行用户注册（[NET DVR Login V30](#)），登录设备。然后设置报警回调函数（[NET DVR SetDVRMessageCallBack V30](#) 等），调用成功后进行布防（[NET DVR SetupAlarmChan V41](#)）。布防即连接设备建立报警上传通道。程序退出前需要调用撤防接口等操作断开连接、释放资源。

[调用实例代码](#)

3.8.2 报警（监听）流程

图 3.9 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。参数都设置完后，调用 [NET_DVR_StartListen_V30](#) 函数，开启 SDK 的监听端口，准备接收设备上传的报警信息。
- 该方式适用于多个设备向一台客户端上传报警，而且不需要设备登录即可完成，设备重启后不影响报警上传；缺点是设备只支持一个报警主机地址和端口号的配置。

[调用实例代码](#)

4 函数调用实例

4.1 通道配置示例代码

[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_USER_LOGIN_INFO struLoginInfo = {0};
    NET_DVR_DEVICEINFO_V40 struDeviceInfo = {0};

    strcpy((char *)struLoginInfo.sDeviceAddress,"192.0.0.64");//设备 IP 地址
    strcpy((char *)struLoginInfo.sUserName,"admin"); //设备登录用户名
    strcpy((char *)struLoginInfo.sPassword,"12345"); //设备登录密码
    struLoginInfo.wPort = 8000;
    struLoginInfo.bUseAsynLogin = 0; //同步登录，登录接口返回成功即登录成功

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

```

printf("The max number of analog channels: %d\n", struDeviceInfo.struDeviceV30.byChanNum); //模拟通道个数
printf("The max number of IP channels: %d\n", struDeviceInfo.struDeviceV30.byIPChanNum); //IP 通道个数
printf("The start number of IP channel: %d\n", struDeviceInfo.struDeviceV30.byStartDChan); //IP 通道起始通道号

//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.slplV4);
                }
                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.slplV4, \
IPAccessCfgV40.struStreamMode[i].uGetStream.struPUStream.struDevChanInfo.struIP.slplV4);
                }
                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);

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
IPAccessCfgV40.struIPDevInfo[iIPCh].byProType=0; //厂家私有协议
strcpy((char *)IPAccessCfgV40.struIPDevInfo[iDevID-1].struIP.sIpV4, "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.struIPDevInfo[iDevID-1].struIP.slIPV4);
}

//注销用户
NET_DVR_Logout(IUserID);

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

```

4.2 预览模块示例代码

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

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

    //-----
    // 注册设备
    LONG IUserID;
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    NET_DVR_DEVICEINFO_V40 struDeviceInfo = {0};

    strcpy((char *)struLoginInfo.sDeviceAddress, "192.0.0.64"); //设备 IP 地址
    strcpy((char *)struLoginInfo.sUserName, "admin"); //设备登录用户名
    strcpy((char *)struLoginInfo.sPassword, "12345"); //设备登录密码
    struLoginInfo.wPort = 8000;
    struLoginInfo.bUseAsynLogin = 0; //同步登录，登录接口返回成功即登录成功

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

```

```

NET_DVR_Cleanup();
return;
}

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

IRealPlayHandle = NET_DVR_RealPlay_V40(IUserID, &struPlayInfo, NULL, NULL);
if (IRealPlayHandle < 0)
{
printf("NET_DVR_RealPlay_V40 failed, error code: %d\n", NET_DVR_GetLastError());
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 lPort; //全局的播放库 port 号

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

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

            if (!PlayM4_GetPort(&lPort)) //获取播放库未使用的通道号
            {
                break;
            }
            //m_iPort = lPort;
            //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(lPort, STREAME_REALTIME)) //设置实时流播放模式
                {
                    break;
                }

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

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

```

```
                break;
            }
        }
        break;
    default: //其他数据
        if (dwBufSize > 0 && lPort != -1)
        {
            if (!PlayM4_InputData(lPort, pBuffer, dwBufSize))
            {
                break;
            }
        }
        break;
    }
}

void CALLBACK g_ExceptionCallBack(DWORD dwType, LONG lUserID, LONG lHandle, 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

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

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

return;
```

```
}
```

4.3 云台控制模块示例代码

[相关模块流程](#)

```
#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);

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

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

```
//-----  
// 注册设备  
LONG IUserID;  
  
NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};  
NET_DVR_DEVICEINFO_V40 struDeviceInfo = {0};  
  
strcpy((char *)struLoginInfo.sDeviceAddress,"192.0.0.64"); //设备 IP 地址  
strcpy((char *)struLoginInfo.sUserName,"admin"); //设备登录用户名  
strcpy((char *)struLoginInfo.sPassword,"12345"); //设备登录密码  
struLoginInfo.wPort = 8000;  
struLoginInfo.bUseAsynLogin = 0; //同步登录，登录接口返回成功即登录成功  
  
IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfo);  
if (IUserID < 0)  
{  
    printf("NET_DVR_Login_V40 failed, error code: %d\n", NET_DVR_GetLastError());  
    NET_DVR_Cleanup();  
    return;  
}  
  
//云台控制:云台左转  
if(!NET_DVR_PTZControl_Other(IUserID,1,PAN_LEFT,0)) //开始左转  
{  
    printf("NET_DVR_PTZControl_Other failed, error code: %d\n", NET_DVR_GetLastError());  
    NET_DVR_Cleanup();  
    return;  
}  
  
printf("云台开始左转!\n");  
Sleep(5000); //转动 5s  
  
if(!NET_DVR_PTZControl_Other(IUserID,1,PAN_LEFT,1)) //停止左转  
{  
    printf("NET_DVR_PTZControl_Other failed, error code: %d\n", NET_DVR_GetLastError());  
    NET_DVR_Cleanup();  
    return;  
}  
printf("云台左转停止!\n");  
  
//-----  
//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.lChannel = 1;        //预览通道号
struPlayInfo.dwStreamType = 0;    //0-主码流，1-子码流，2-码流 3，3-码流 4，以此类推
struPlayInfo.dwLinkMode = 0;     //0- TCP 方式，1- UDP 方式，2- 多播方式，3- RTP 方式，4-RTP/RTSP，5-RSTP/HTTP
struPlayInfo.bBlocked = 1;       //0- 非阻塞取流，1- 阻塞取流

IRealPlayHandle = NET_DVR_RealPlay_V40(IUserID, &struPlayInfo, NULL, NULL);
if (IRealPlayHandle < 0)
{
    printf("NET_DVR_RealPlay_V40 failed, error code: %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//云台控制:云台右转
if(!NET_DVR_PTZControl(IRealPlayHandle,PAN_RIGHT,0)) //开始右转
{
    printf("NET_DVR_PTZControl failed, error code: %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

Sleep(5000); //转动 5s

if(!NET_DVR_PTZControl(IRealPlayHandle,PAN_RIGHT,1)) //停止右转
{
    printf("NET_DVR_PTZControl failed, error code: %d\n", NET_DVR_GetLastError());
    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;  
}
```

4.4 回放和下载模块示例代码

[相关模块流程图](#)

示例一：查找录像文件

```
#include <stdio.h>  
#include <iostream>  
#include "Windows.h"  
#include "HCNetSDK.h"  
using namespace std;  
  
void main() {  
  
    //-----  
    // 初始化  
    NET_DVR_Init();  
    //设置连接时间与重连时间  
    NET_DVR_SetConnectTime(2000, 1);  
    NET_DVR_SetReconnect(10000, true);  
  
    //-----  
    // 注册设备  
    LONG IUserID;  
    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};  
    NET_DVR_DEVICEINFO_V40 struDeviceInfo = {0};  
  
    strcpy((char *)struLoginInfo.sDeviceAddress, "192.0.0.64"); //设备 IP 地址  
    strcpy((char *)struLoginInfo.sUserName, "admin"); //设备登录用户名  
    strcpy((char *)struLoginInfo.sPassword, "12345"); //设备登录密码  
    struLoginInfo.wPort = 8000;  
    struLoginInfo.bUseAsynLogin = 0; //同步登录，登录接口返回成功即登录成功  
  
    IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfo);  
    if (IUserID < 0)  
    {  
        printf("NET_DVR_Login_V40 failed, error code: %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 lFindHandle = NET_DVR_FindFile_V40(lUserID, &struFileCond);
if(lFindHandle < 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(lFindHandle, &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};
        printf ("Starting time: %d-%d-%d %d:%d:%d\n", struFileData.struStartTime.dwYear,
struFileData.struStartTime.dwMonth, struFileData.struStartTime.dwDay, struFileData.struStartTime.dwHour,
struFileData.struStartTime.dwMinute, struFileData.struStartTime.dwSecond);
        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：按时间回放录像

```
#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 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_PLAYCOND struDownloadCond={0};  
struDownloadCond.dwChannel=1; //通道号  
struDownloadCond.struStartTime.dwYear   = 2013; //开始时间  
struDownloadCond.struStartTime.dwMonth  = 6;  
struDownloadCond.struStartTime.dwDay    = 14;  
struDownloadCond.struStartTime.dwHour   = 9;  
struDownloadCond.struStartTime.dwMinute = 50;  
struDownloadCond.struStartTime.dwSecond =0;  
struDownloadCond.struStopTime.dwYear    = 2013; //结束时间  
struDownloadCond.struStopTime.dwMonth   = 6;  
struDownloadCond.struStopTime.dwDay     = 14;  
struDownloadCond.struStopTime.dwHour    = 10;  
struDownloadCond.struStopTime.dwMinute  = 7;  
struDownloadCond.struStopTime.dwSecond = 0;  
  
//-----  
//按时间下载  
int hPlayback;  
hPlayback = NET_DVR_GetFileByTime_V40(IUserID, "./test.mp4",&struDownloadCond);  
if(hPlayback < 0)  
{  
    printf("NET_DVR_GetFileByTime_V40 fail,last error %d\n",NET_DVR_GetLastError());  
    NET_DVR_Logout(IUserID);  
    NET_DVR_Cleanup();  
    return;  
}  
  
//-----  
//开始下载  
if(!NET_DVR_PlayBackControl_V40(hPlayback, NET_DVR_PLAYSTART, NULL, 0, NULL,NULL))  
{
```

```
printf("Play back control failed [%d]\n",NET_DVR_GetLastError());
NET_DVR_Logout(IUserID);
NET_DVR_Cleanup();
return;
}

int nPos = 0;
for(nPos = 0; nPos < 100&& nPos >= 0; nPos = NET_DVR_GetDownloadPos(hPlayback))
{
printf("Be downloading... %d %%\n",nPos); //下载进度
Sleep(5000); //millisecond
}
if(!NET_DVR_StopGetFile(hPlayback))
{
printf("failed to stop get file [%d]\n",NET_DVR_GetLastError());
NET_DVR_Logout(IUserID);
NET_DVR_Cleanup();
return;
}
if(nPos < 0 || nPos > 100)
{
printf("download err [%d]\n",NET_DVR_GetLastError());
NET_DVR_Logout(IUserID);
NET_DVR_Cleanup();
return;
}
printf("Be downloading... %d %%\n",nPos);

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

4.5 参数配置模块示例代码

[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_USER_LOGIN_INFO struLoginInfo = {0};
    NET_DVR_DEVICEINFO_V40 struDeviceInfo = {0};

    strcpy((char *)struLoginInfo.sDeviceAddress,"192.0.0.64"); //设备 IP 地址
    strcpy((char *)struLoginInfo.sUserName,"admin"); //设备登录用户名
    strcpy((char *)struLoginInfo.sPassword,"12345"); //设备登录密码
    struLoginInfo.wPort = 8000;
    struLoginInfo.bUseAsyncLogin = 0; //同步登录，登录接口返回成功即登录成功

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

    int iRet;
    //获取通道 1 的压缩参数
    DWORD dwReturnLen;
    NET_DVR_COMPRESSIONCFG_V30 struParams = {0};
    iRet = NET_DVR_GetDVRConfig(IUserID, NET_DVR_GET_COMPRESSCFG_V30, 1, &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;
    }
}
```

```
}

//设置通道 1 的压缩参数
struParams.struNormHighRecordPara.dwVideoBitrate = 22;
iRet = NET_DVR_SetDVRConfig(IUserID, NET_DVR_SET_COMPRESSCFG_V30, 1, \
    &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;
}

//获取通道 1 的压缩参数
iRet = NET_DVR_GetDVRConfig(IUserID, NET_DVR_GET_COMPRESSCFG_V30, 1, \
    &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;
}
```

4.6 远程设备维护模块示例代码

[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 IUserID;

    NET_DVR_USER_LOGIN_INFO struLoginInfo = {0};
    NET_DVR_DEVICEINFO_V40 struDeviceInfo = {0};

    strcpy((char *)struLoginInfo.sDeviceAddress,"192.0.0.64"); //设备 IP 地址
    strcpy((char *)struLoginInfo.sUserName,"admin"); //设备登录用户名
    strcpy((char *)struLoginInfo.sPassword,"12345"); //设备登录密码
    struLoginInfo.wPort = 8000;
    struLoginInfo.bUseAsynLogin = 0; //同步登录，登录接口返回成功即登录成功

    IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfo);
    if (IUserID < 0)
    {
        printf("NET_DVR_Login_V40 failed, error code: %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;
}
```

4.7 报警模块示例代码

[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 ICommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD dwBufLen, void*
pUser)
{
    switch(ICommand)
    {
        case COMM_ALARM_DEVICE_V40:
            {
                NET_DVR_ALARMINFO_DEV_V40 struDeviceAlarm;
                memcpy(&struDeviceAlarm, pAlarmInfo, sizeof(NET_DVR_ALARMINFO_DEV_V40));

                char szLan[1024] = {0};
                sprintf(szLan, "Device Alarm(CVR): type[%d], num[%d]", struDeviceAlarm.dwAlarmType,
struDeviceAlarm.dwNumber);

                int iNumber = struDeviceAlarm.dwNumber;
                WORD *pContent = struDeviceAlarm.pNO;
                for (int i = 0; i < iNumber && i < 10; i++)
                {
                    sprintf(szLan, "%s %d", szLan, *(pContent));
                    pContent = pContent + 1;
                }
                printf(szLan);
            }
            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_USER_LOGIN_INFO struLoginInfo = {0};  
    NET_DVR_DEVICEINFO_V40 struDeviceInfo = {0};  
  
    strcpy((char *)struLoginInfo.sDeviceAddress,"192.0.0.64"); //设备 IP 地址  
    strcpy((char *)struLoginInfo.sUserName,"admin"); //设备登录用户名  
    strcpy((char *)struLoginInfo.sPassword,"12345"); //设备登录密码  
    struLoginInfo.wPort = 8000;  
    struLoginInfo.bUseAsyncLogin = 0; //同步登录，登录接口返回成功即登录成功  
  
    IUserID = NET_DVR_Login_V40(&struLoginInfo, &struDeviceInfo);  
    if (IUserID < 0)  
    {  
        printf("NET_DVR_Login_V40 failed, error code: %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.byRetDevInfoVersion = 1; /*CVR 上传报警信息回调结构体版本：  
        0- COMM_ALARM_DEVICE,  
        1- COMM_ALARM_DEVICE_V40*/  
  
    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)
{
    switch(ICommand)
    {
    {
    case COMM_ALARM_DEVICE_V40:
        {
            NET_DVR_ALARMINFO_DEV_V40 struDeviceAlarm;
            memcpy(&struDeviceAlarm, pAlarmInfo, sizeof(NET_DVR_ALARMINFO_DEV_V40));

            char szLan[1024] = {0};
            sprintf(szLan, "Device Alarm(CVR): type[%d], num[%d]", struDeviceAlarm.dwAlarmType,

```

```
struDeviceAlarm.dwNumber);
    int iNumber = struDeviceAlarm.dwNumber;
    WORD *pContent = struDeviceAlarm.pNO;
    for (int i = 0; i < iNumber && i < 10; i++)
    {
        sprintf(szLan, "%s %d", szLan, *(pContent));
        pContent = pContent + 1;
    }
    printf(szLan);
}
break;
default:
    break;
}
}

void main() {

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

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

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

```
NET_DVR_Cleanup();
return;
}

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

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

5 函数说明

5.1 SDK 初始化

5.1.1 初始化 SDK `NET_DVR_Init`

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

[返回目录](#)

5.1.2 释放 SDK 资源 `NET_DVR_Cleanup`

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

[返回目录](#)

5.2 SDK 本地功能

SDK 本地参数配置

5.2.1 获取 SDK 本地参数 `NET_DVR_GetSDKLocalCfg`

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

表 5.1 本地参数类型

enumType 宏定义	类型值	含义	IpOutBuff 对应结构体
NET_SDK_LOCAL_CFG_TYPE_TCP_PORT_BIND	0	本地 TCP 端口绑定配置	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、能力

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

函 数: DWORD NET_DVR_GetSDKBuildVersion()

参 数:

返回值: 获取 SDK 的版本号和 build 信息。

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

[返回目录](#)

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

函 数: BOOL NET_DVR_GetSDKState(LPNET_DVR_SDKSTATE pSDKState);

参 数: [out] pSDKState 状态信息结构

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

说 明:

[返回目录](#)

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

函 数: BOOL NET_DVR_GetSDKAbility(LPNET_DVR_SDKABL pSDKAbl)

参 数: [out] pSDKAbl 功能信息结构

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:

[返回目录](#)

SDK 启用写日志

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

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

参 数: [in]bLogEnable 日志的等级 (默认为 0):

0- 表示关闭日志;

1- 表示只输出 ERROR 错误日志;

2- 输出 ERROR 错误信息和 DEBUG 调试信息;

3- 输出 ERROR 错误信息、DEBUG 调试信息和 INFO 普通信息等所有信息

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

[in]bAutoDel 是否删除超出的文件数, 默认值为 TRUE

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

说 明:

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

[返回目录](#)

异常消息回调

5.2.12 注册接收异常、重连等消息的窗口句柄或回调函数

NET_DVR_SetExceptionCallBack_V30

函 数: **Windows 系统下:**

BOOL NET_DVR_SetExceptionCallBack_V30 (UINT nMessage,HWND hWnd,fExceptionCallBack

cbExceptionCallBack,void* pUser)

Linux 系统下:

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

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

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

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

表 5.3 异常消息类型

dwType 宏定义	宏定义值	含义
EXCEPTION_EXCHANGE	0x8000	用户交互时异常 (注册心跳超时, 心跳间隔为 2 分钟)
EXCEPTION_ALARM	0x8002	报警异常
EXCEPTION_PREVIEW	0x8003	网络预览异常
EXCEPTION_RECONNECT	0x8005	预览时重连
EXCEPTION_ALARMRECONNECT	0x8006	报警时重连
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_RECONNECTSUCESS	0x8015	预览时重连成功
ALARM_RECONNECTSUCESS	0x8016	报警时重连成功
RESUME_EXCHANGE	0x8017	用户交互恢复
EXCEPTION_RELOGIN	0x8040	用户重登陆
RELOGIN_SUCESS	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_ALARM: //报警上传时网络异常
        sprintf(tempbuf,"报警上传时网络异常!!!");
        TRACE("%s",tempbuf);
        //TODO: 关闭报警上传
        break;
    case EXCEPTION_PREVIEW: //网络预览时异常
        sprintf(tempbuf,"网络预览时网络异常!!!");
        TRACE("%s",tempbuf);
        //TODO: 关闭网络预览
        break;
    case EXCEPTION_RECONNECT: //预览时重连
        break;
    default:
        break;
    }
};
```

[返回目录](#)

获取错误信息

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

函数: `DWORD NET_DVR_GetLastError()`

参数:

返回值: 返回最后操作的错误码。详见[错误码宏定义](#)

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

[返回目录](#)

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

函数: `char* NET_DVR_GetErrorMsg(LONG *pErrorNo)`

参数: [out] pErrorNo 错误码数值的指针

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

说明:

[返回目录](#)

5.3 用户注册

5.3.1 通过解析服务器，获取设备的动态 IP 地址和端口号

NET_DVR_GetDVRIPByResolveSvr_EX

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

参 数：

[in]sServerIP	解析服务器(IPServer 或者 hiDDNS)的 IP 地址或者域名
[in]wServerPort	解析服务器的端口号，IP Server 端口号为 7071，hiDDNS 服务器的端口号为 80
[in]sDVRName	设备名称（或者设备域名）
[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](#)

5.3.2 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。
- 设备最多支持 32 个注册用户名，且同时最多允许 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](#)

5.3.3 用户注销 **NET_DVR_Logout**

函数: `BOOL NET_DVR_Logout(LONG IUserID)`

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

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

说明: 建议使用此接口实现注销功能。

[返回目录](#)

5.4 获取设备能力集

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

函数: `BOOL NET_DVR_GetDeviceAbility(LONG IUserID, DWORD dwAbilityType, char* pInBuf, DWORD dwInLength, char* pOutBuf, DWORD dwOutLength)`

参数: `[in]IUserID` `NET_DVR_Login_V40` 的返回值

`[in]dwAbilityType` 能力集类型, 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
EVICE_DYNCHAN_ABILITY	0x00b	设备数字通道能力
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_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
STREAM_ABILITY	0x012	设备流能力集

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(BasicCapability)
DEVICE_NETWORK_ABILITY	无	设备网络能力 XML 描述(NetworkSetting)
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(AudioVideoCompressInfo)
DEVICE_RAID_ABILITY	无	设备 RAID 能力 XML 描述(RAID)
DEVICE_DYNCHAN_ABILITY	获取设备数字通道能力	设备数字通道能力 XML 描述(DynChannelAbility)
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(UserAbility)
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(NetAppAbility)
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(VideoPicAbility)
DEVICE_ABILITY_INFO	获取录像相关能力集	录像相关能力 XML 描述(RecordAbility)
	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)
	获取设备通道输入能力集	设备通道输入能力 XML 描述(ChannelInputAbility)
	获取 CVR 设备能力集	日志 CVR 设备能力 XML 描述(CVRAbility)
STREAM_ABILITY	获取流能力集	流能力 XML 描述(StreamAbility)

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 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），不同的能力集对应不同的输入输出参数

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

说明：

表 5.6 获取设备能力集

宏定义	宏定义值	含义	lpCondBuffer	lpOutBuffer
NET_DVR_GET_RACM_CAP	6252	获取存储总能力集（RACM 能力）	NULL	RacmCap
NET_DVR_GET_RECORD_PASSBACK_BASIC_CFG_CAP	6248	获取回传功能基础配置能力集	NULL	RecordPassBackBasicCfg
NET_DVR_GET_RECORD_PASSBACK_HISTORY_PLAN_CFG_CAP	6255	获取回传历史录像计划能力集	NULL	RecordPassBackHistoryPlanCfg

[返回目录](#)

5.5 实时预览

强制 I 帧

5.5.1 强制 I 帧控制 **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 5.7
 [in]lpInBuffer 输入参数, 具体内容跟控制命令相关, 详见表 5.7
 [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 5.7。

表 5.7 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](#)

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

函 数: BOOL NET_DVR_MakeKeyFrame(LONG IUserID, DWORD IChannel)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
 [in]IChannel 通道号

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

说 明: 此接口用于重置 I 帧, 根据设置的预览参数(NET_DVR_CLIENTINFO)为主码流或者子码流分别调用 NET_DVR_MakeKeyFrame 或者 [NET_DVR_MakeKeyFrameSub](#) 实现重置 I 帧。

[返回目录](#)

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

函 数: BOOL NET_DVR_MakeKeyFrameSub(LONG IUserID, DWORD IChannel)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
 [in]IChannel 通道号

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

说 明: 此接口用于重置 I 帧, 根据设置的预览参数(NET_DVR_CLIENTINFO)为主码流或者子码流分别调用 [NET_DVR_MakeKeyFrame](#) 或者 NET_DVR_MakeKeyFrameSub 实现重置 I 帧。

[返回目录](#)

实时预览

5.5.4 实时预览 `NET_DVR_RealPlay_V40`

函数: `LONG NET_DVR_RealPlay_V40(LONG IUserID, LPNET_DVR_PREVIEWINFO lpPreviewInfo, REALDATACALLBACK fRealDataCallBack_V30, void *pUser)`

参数: [in] IUserID NET_DVR_Login_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 5.8
 [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 5.8 码流数据类型

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

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

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

[返回目录](#)

5.5.5 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](#)

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

函 数: int NET_DVR_GetRealPlayerIndex(LONG IRealHandle)

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

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

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

[返回目录](#)

预览画面叠加字符和图像

5.5.7 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](#)

预览时播放声音控制

5.5.8 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](#)

5.5.9 独占声卡模式下开启声音 NET_DVR_OpenSound

函数: BOOL NET_DVR_OpenSound(LONG IRealHandle)

参数: [in]IRealHandle NET_DVR_RealPlay_V40 的返回值

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

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

[返回目录](#)

5.5.10 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](#)

5.5.11 共享声卡模式下开启声音 NET_DVR_OpenSoundShare

函数: BOOL NET_DVR_OpenSoundShare(LONG IRealHandle)

参数: [in]IRealHandle NET_DVR_RealPlay_V40 的返回值

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

说明:

[返回目录](#)

5.5.12 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](#)

5.5.13 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](#)

实时数据回调和录像

5.5.14 设置实时码流数据回调函数 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](#)

5.5.15 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](#)

5.5.16 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](#)

预览抓图

5.5.17 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](#)

5.5.18 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](#)

5.6 云台控制

云台控制操作

5.6.1 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 5.10
- [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: CVR 设备通道模式下支持, 流 ID 模式时不支持。
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 5.10 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](#)

5.6.2 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 5.10
[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: CVR 设备通道模式下支持，流 ID 模式时不支持。

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](#)

5.6.3 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 5.10
[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: CVR 设备通道模式下支持, 流 ID 模式时不支持。

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](#)

5.6.4 带速度的云台控制操作（不用启动图像预览）

NET_DVR_PTZControlWithSpeed_Other

函数: BOOL NET_DVR_PTZControlWithSpeed(LONG IUserID, LONG IChannel, DWORD dwPTZCommand, DWORD dwStop, DWORD dwSpeed)

参数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IChannel 通道号
[in]dwPTZCommand 云台控制命令, 详见表 5.10
[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: CVR 设备通道模式下支持, 流 ID 模式时不支持。

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

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

函数： BOOL NET_DVR_PTZPreset(LONG IRealHandle,DWORD dwPTZPresetCmd,DWORD dwPresetIndex)

参数： [in]IRealHandle NET_DVR_RealPlay_V40 的返回值
[in]dwPTZPresetCmd 操作云台预置点命令，详见表 5.11
[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： CVR 设备通道模式下支持，流 ID 模式时不支持。

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 5.11 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](#)

5.6.6 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 5.11
[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： CVR 设备通道模式下支持，流 ID 模式时不支持。

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

5.6.7 云台巡航操作，需先启动预览 **NET_DVR_PTZCruise**

- 函数：** BOOL NET_DVR_PTZCruise(LONG lRealHandle,DWORD dwPTZCruiseCmd, BYTE byCruiseRoute, BYTE byCruisePoint, WORD wInput)
- 参数：**
- [in]lRealHandle NET_DVR_RealPlay_V40 的返回值
 - [in]dwPTZCruiseCmd 操作云台巡航命令，详见表 5.12
 - [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：** CVR 设备通道模式下支持，流 ID 模式时不支持。
- 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 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](#)

5.6.8 云台巡航操作 **NET_DVR_PTZCruise_Other**

- 函数：** BOOL NET_DVR_PTZCruise_Other(LONG lUserID,LONG lChannel,DWORD dwPTZCruiseCmd, BYTE byCruiseRoute, BYTE byCruisePoint, WORD wInput)
- 参数：**
- [in]lUserID NET_DVR_Login_V40 的返回值
 - [in]lChannel 通道号
 - [in]dwPTZCruiseCmd 操作云台巡航命令，详见表 5.12
 - [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: CVR 设备通道模式下支持, 流 ID 模式时不支持。

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

[返回目录](#)

云台花样扫描路径

5.6.9 云台花样扫描路径操作, 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 5.13

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

说明: CVR 设备通道模式下支持, 流 ID 模式时不支持。

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 操作命令

dwPTZTrackCmd 宏定义	宏定义值	含义
STA_MEM_CRUISE	34	开始记录花样扫描路径
STO_MEM_CRUISE	35	停止记录花样扫描路径
RUN_CRUISE	36	开始执行花样扫描路径

[返回目录](#)

5.6.10 云台花样扫描路径操作 **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 5.13

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

说明: CVR 设备通道模式下支持, 流 ID 模式时不支持。

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](#)

5.7 录像回放、下载、锁定及备份

月历录像查询

5.7.1 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 5.14
- [in] dwCount 一次要获取配置参数的个数，0 和 1 都表示 1 个信息，2 表示 2 个信息，最大 64 个
- [in] lpInBuffer 配置条件缓冲区，详见表 5.15
- [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 5.15），和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 0 值，对应 lpOutBuffer 的内容就是无效的
- [in] dwOutBufferSize 输出缓冲区大小

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 5.15 所示。

表 5.14 参数批量获取命令

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 5.15 批量获取设备参数

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

[返回目录](#)

通道模式录像查找

5.7.2 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](#)

5.7.3 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 5.16。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

表 5.16 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

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](#)

5.7.4 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](#)

流 ID 模式录像查找

5.7.5 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，支持搜索类型 EVENT_STREAM_INFO，通过流 ID 等条件查找录像
- 返回值:** -1 表示失败，其他值作为 NET_DVR_FindNextEvent_V40 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，只支持按时间回放 NET_DVR_PlayBackByTime_V40。

[返回目录](#)

5.7.6 逐个获取查找到的文件信息 **NET_DVR_FindNextEvent_V40**

- 函数:** LONG NET_DVR_FindNextEvent_V40(LONG IFindHandle, LPNET_DVR_SEARCH_EVENT_RET_V40 lpSearchEventRet)
- 参数:** [in]ISearchHandle NET_DVR_Login_V40 的返回值
[out]lpSearchEventRet 保存文件信息的指针，包括查找到的录像类型、起止时间等
- 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 5.17。

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](#)

5.7.7 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_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](#)

智能搜索

5.7.8 开始智能搜索 [NET_DVR_SmartSearch_V40](#)

函 数: LONG NET_DVR_SmartSearch_V40(LONG IUserID, LPNET_DVR_SMART_SEARCH_PARAM_V40 lpSmartSearchParam)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]lpSmartSearchParam 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](#)

5.7.9 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 lpSmartSearchRet)

参 数: [in]ISearchHandle NET_DVR_SmartSearch_V40 的返回值
[out]lpSmartSearchRet 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 5.18。

表 5.18 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](#)

5.7.10 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](#)

录像远程回放

5.7.11 按时间回放录像 **NET_DVR_PlayBackByTime_V40**

函数: `LONG NET_DVR_PlayBackByTime_V40(LONG IUserID, LPNET_DVR_VOD_PARA pVodPara)`

参数: `[in]IUserID` `NET_DVR_Login_V40` 的返回值

`[in]pVodPara` 回放参数, 包括流 ID 或通道号、回放起止时间等

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](#)

5.7.12 按时间倒放录像 **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，包括流 ID 或通道号、回放起止时间等

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](#)

5.7.13 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_PlayBackByTime_V40` 或 `NET_DVR_PlayBackReverseByTime_V40` 的返回值
[in]dwControlCode 控制录像回放状态命令，详见表 5.19
[in] lpInBuffer 指向输入参数的指针，详见表 5.20
[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 5.20
[out]lpOutLen 输出参数的长度

表 5.19 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_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_PLAYGETPOS	13	获取回放的进度，按时间回放只支持获取 0、100、200
NET_DVR_SETSPEED	24	设置码流速度，速度单位：kbps，最小为 256kbps，最大为设备带宽
NET_DVR_PLAYSETTIME	26	按绝对时间定位
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
NET_DVR_START_DRAWFRAME	34	开始抽帧回放
NET_DVR_STOP_DRAWFRAME	35	停止抽帧回放

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： 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 5.20 所示。

- 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）。
- 命令值为 NET_DVR_PLAYGETPOS 时，按时间回放或下载，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 5.20 回放控制参数

dwControlCode 宏定义	状态命令说明	lpInBuf	lpOutBuf
NET_DVR_PLAYSTART	开始播放	一个 4 字节整型的偏移量	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_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_PLAY_FORWARD	倒放切换为正放	应用层解码时需在 lpInBuffer 输入 NET_DVR_TIME 表示当前的播放时间,SDK 解码可将 lpInBuffer 置为 NULL	无

NET_DVR_PLAY_REVERSE	正放切换为倒放	应用层解码时需在 IplayBuffer 输入 NET_DVR_TIME 表示当前的播放时间,SDK 解码可将 IplayBuffer 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_START_DRAWFRAME	开始抽帧回放	NET_DVR_TIME_EX	无
NET_DVR_STOP_DRAWFRAME	停止抽帧回放	NET_DVR_TIME_EX	无

[返回目录](#)

5.7.14 停止回放录像 NET_DVR_StopPlayBack

函数: BOOL NET_DVR_StopPlayBack(LONG IplayHandle)

参数: [in]IplayHandle 回放句柄, NET_DVR_PlayBackByTime_V40 或
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](#)

回放数据回调和录像

5.7.15 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_PlayBackByTime_V40 或
NET_DVR_PlayBackReverseByTime_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](#)

5.7.16 停止保存录像数据 NET_DVR_StopPlayBackSave

函数: BOOL NET_DVR_StopPlayBackSave(LONG IplayHandle)

参数: [in]IplayHandle 播放句柄, NET_DVR_PlayBackByTime_V40 或
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](#)

5.7.17 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 5.21

[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 5.21 回放回调数据类型

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

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

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

[返回目录](#)

回放抓图等其他操作

5.7.18 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 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](#)

5.7.19 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 的返回值
[in]sFileName 保存图片数据的文件路径，包括文件名的绝对路径，例如：“C:/test.bmp”

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

5.7.20 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 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](#)

5.7.21 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 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

5.7.22 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
 [in]pDownloadCond 下载条件, 包括通道号或流 ID、下载起止时间等

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](#)

5.7.23 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_GetFileByTime_V40 的返回值
 [in]dwControlCode 控制录像回放状态命令, 详见表 5.22
 [in]lpInBuffer 指向输入参数的指针, 详见表 5.23
 [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 5.23
 [out]lpOutLen 输出参数的长度

表 5.22 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_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 5.23 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 5.23 下载控制参数

状态命令宏定义	状态命令说明	IpInBuf	IpOutBuf
NET_DVR_PLAYSTART	开始下载	一个 4 字节整型的偏移量	无
NET_DVR_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](#)

5.7.24 停止下载录像文件 NET_DVR_StopGetFile

函数： BOOL NET_DVR_StopGetFile(LONG IFileHandle)

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

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

说明：

[返回目录](#)

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

函数： int NET_DVR_GetDownloadPos(LONG IFileHandle)

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

返回值： -1 表示失败，0~100 表示下载的进度，100 表示下载结束。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： 按时间下载进度只支持获取 0、100（完成）、200（异常）三个值。

[返回目录](#)

录像文件锁定和解锁

5.7.26 通道模式按时间段锁定录像 NET_DVR_LockFileByTime

函数： BOOL NET_DVR_LockFileByTime(LONG IUserID, LPNET_DVR_TIME_LOCK IpLockPara, LPNET_DVR_LOCK_RETURN IpLockReturn)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IpLockPara 录像加锁参数, 包括通道号、时间段、录像类型和锁定时间等
[out]IpLockPara 录像加锁结果, 包括被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: 当录像被锁定后, 将不会被覆盖。

[返回目录](#)

5.7.27 通道模式按时间段解锁录像 NET_DVR_UnLockFileByTime

函 数: BOOL NET_DVR_UnLockFileByTime(LONG IUserID, LPNET_DVR_TIME_LOCK IpLockPara, LPNET_DVR_LOCK_RETURN IpLockReturn)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IpLockPara 录像解锁参数, 包括通道号、时间段、录像类型和锁定时间等
[out]IpLockReturn 录像解锁结果, 包括被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](#)

5.7.28 流模式按时间段锁定录像 NET_DVR_LockStreamFileByTime

函 数: BOOL NET_DVR_LockStreamFileByTime(LONG IUserID, LPNET_DVR_STREAM_TIME_LOCK IpLockPara, LPNET_DVR_LOCK_RETURN IpLockReturn)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]IpLockPara 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]IpLockReturn 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: 该接口也支持通道模式下按时间段锁定录像。

[返回目录](#)

5.7.29 流模式按时间段解锁录像 NET_DVR_UnlockStreamFileByTime

函 数: BOOL NET_DVR_UnlockStreamFileByTime(LONG IUserID, LPNET_DVR_STREAM_TIME_LOCK IpLockPara, LPNET_DVR_LOCK_RETURN IpLockReturn)

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: 该接口也支持通道模式下按时间段解锁录像。

[返回目录](#)

录像时间段检索

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

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

参数: [in] IUserID 用户 ID 号, NET_DVR_Login_V30 返回值
 [in] dwCommand 配置命令, 详见表 5.24
 [in] lpInBuffer 输入参数, 具体内容跟配置命令相关, 详见表 5.24
 [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 dwCommand、dwType 相关
 [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: 调用该接口启动长连接远程配置后, 部分命令还需要调用其他接口获取信息或者设置相关参数, 如表 5.25 所示。

表 5.24 长连接参数配置

dwCommand 宏定义	宏定义值	含义	lpInBuffer 对应结构	cbStateCallback
NET_DVR_RECORD_CHECK	6233	录像完整性检测	NET_DVR_RECORD_CHECK_COND	NULL
NET_DVR_GET_RECORD_SEGMENT_CFG	6242	获取录像段总大小	NET_DVR_RECORD_SEGMENT_COND	NULL

表 5.25 后续接口调用

dwCommand 宏定义	含义	后续接口调用
NET_DVR_RECORD_CHECK	录像完整性检测	NET_DVR_GetNextRemoteConfig
NET_DVR_GET_RECORD_SEGMENT_CFG	获取录像段总大小	NET_DVR_GetNextRemoteConfig

[返回目录](#)

5.7.31 逐个获取查找到的信息 **NET_DVR_GetNextRemoteConfig**

函数: LONG NET_DVR_GetNextRemoteConfig(LONG IHandle, void *lpOutBuff, DWORD dwOutBuffSize)

参数: [in] IHandle 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 5.27
 [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 5.26。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

表 5.26 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_NEXT_STATUS_NEED_WAIT	1001	需等待设备发送数据, 继续调用 NET_DVR_GetNextRemoteConfig
NET_SDK_GET_NEXT_STATUS_FINISH	1002	数据全部取完, 可调用 NET_DVR_StopRemoteConfig 结束长连接
NET_SDK_GET_NEXT_STATUS_FAILED	1003	出现异常, 可调用 NET_DVR_StopRemoteConfig 结束长连接

说明: 调用 [NET_DVR_StartRemoteConfig](#) 时传入不同的命令号(dwCommand), lpOutBuff 对应不同的结构体, 如表 5.27 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 5.27 长连接参数获取

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer
NET_DVR_RECORD_CHECK	6233	录像完整性检测	NET_DVR_RECORD_CHECK_RET
NET_DVR_GET_RECORD_SEGMENT_CFG	6242	获取录像段总大小	NET_DVR_RECORD_SEGMENT_RET

[返回目录](#)

5.7.32 关闭长连接 [NET_DVR_StopRemoteConfig](#)

函数: BOOL NET_DVR_StopRemoteConfig(LONG IHandle)

参数: [in] IHandle 句柄, NET_DVR_StartRemoteConfig 的返回值

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

说明: 关闭长连接配置接口所创建的句柄, 释放资源。

[返回目录](#)

录像缩略图

5.7.33 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, 不同的功能对应不同的命令号, lpInBuffer 等参数也对应不同的内容, 详见表 5.28

[in] lpInBuffer 输入参数, 具体内容跟配置命令相关, 详见表 5.28

[in] dwInBufferLen 输入缓冲的大小
 [in] cbStateCallback 状态回调函数
 [in] pUserData 用户数据
typedef void(CALLBACK *fRemoteConfigCallback)(DWORD dwType, void *lpBuffer, DWORD dwBufLen, void *pUserData)

[out] dwType 配置状态，具体定义如下所示：

```
enum _NET_SDK_CALLBACK_TYPE_{
    NET_SDK_CALLBACK_TYPE_STATUS = 0, //回调状态值
    NET_SDK_CALLBACK_TYPE_PROGRESS, //回调进度值
    NET_SDK_CALLBACK_TYPE_DATA       //回调数据内容
}NET_SDK_CALLBACK_TYPE
```

[out] lpBuffer 存放数据的缓冲区指针，具体内容跟 dwCommand、dwType 相关，详见表 5.29

[out] dwBufLen 缓冲区大小

[out] pUserData 用户数据

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

说明： 调用该接口建立长连接，需要设置回调函数，查询结果直接在回调里面返回。

表 5.28 长连接参数配置

dwCommand 宏定义	宏定义值	含义	lpInBuffer 对应结构	cbStateCallback
NET_DVR_GET_THUMBNAILS	6253	按时间段获取缩略图（录像的缩略图，支持流 ID)	NET_DVR_THUMBNAILS_COND	返回状态、信息数据
NET_DVR_GET_FIGURE	6610	按时间点获取缩略图	NET_DVR_GET_FIGURE_COND	返回状态、信息数据

表 5.29 回调数据内容

dwType	含义	lpBuffer 对应内容
NET_SDK_CALLBACK_TYPE_STATUS	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	进度值	lpBuffer 的值表示进度(DWORD)
NET_SDK_CALLBACK_TYPE_DATA	信息数据	lpBuffer 的值表示信息数据。 NET_DVR_GET_THUMBNAILS 按时间段获取缩略图时，对应结构体： NET_DVR_THUMBNAILS_RESULT; NET_DVR_GET_FIGURE 按时间点获取缩略图时，对应结构体： NET_DVR_FIGURE_INFO。

[返回目录](#)

5.7.34 关闭长连接 **NET_DVR_StopRemoteConfig**

函数： BOOL NET_DVR_StopRemoteConfig(LONG lHandle)

参 数: [in] IHandle 句柄, NET_DVR_StartRemoteConfig 的返回值
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明: 关闭长连接配置接口所创建的句柄, 释放资源。

[返回目录](#)

5.8 图片的查找、回放及备份

查找图片

5.8.1 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。

[返回目录](#)

5.8.2 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 5.30 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 5.30 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](#)

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

函数: BOOL NET_DVR_CloseFindPicture(LONG IFindHandle)

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

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

说明:

[返回目录](#)

回放（下载）图片

5.8.4 图片回放 **NET_DVR_GetPicture_V30**

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

参数: [in]IUserID NET_DVR_Login_V40 的返回值

[in]sDVRFileName 图片名称

[in]sSavedFileName 保存图片的缓冲区

[in]dwBufLen 缓冲区大小

[out]lpdwRetLen 实际收到的数据长度指针，不能为 NULL

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

说明: 图片为 JPEG 格式，数据保存在缓冲区 sSavedFileName 中。

[返回目录](#)

5.9 远程参数配置

通用参数配置

5.9.1 获取设备的配置信息 **NET_DVR_GetDVRConfig**

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

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

[in]dwCommand 设备配置命令，详见表 5.31

[in]IChannel 通道号，不同的命令对应不同的取值，如果该参数无效则置为 0xFFFFFFFF 即可，详见表 5.31

[out]lpOutBuffer 接收数据的缓冲指针，详见表 5.31

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

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

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

说明: 不同的获取功能对应不同的结构体和命令号，如表 5.31 所示。

表 5.31 参数获取命令

dwCommand 宏定义	dwCommand 含义	IChannel 取值	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_DEVICECFG_V40	获取设备参数(扩展)	无效	NET_DVR_DEVICECFG_V40	1100
NET_DVR_GET_TIMECFG	获取时间参数	无效	NET_DVR_TIME	118
NET_DVR_GET_NETCFG_V30	获取网络参数	无效	NET_DVR_NETCFG_V30	1000
NET_DVR_GET_RECORDCFG_V40	获取录像计划参数	通道号	NET_DVR_RECORD_V40	1008
NET_DVR_GET_IPPARACFG_V40	获取 IP 接入配置参数	组号, 从 0 开始, 每组 64 个通道	NET_DVR_IPPARACFG_V40	1062
NET_DVR_GET_USERCFG_V40	获取用户参数	组号, 从 0 开始, 每组 32 个用户	NET_DVR_USER_V40	6187
NET_DVR_GET_RECORD_CHANNEL_INFO	获取通道录像状态信息	组号, 从 0 开始, 每组 64 个通道	NET_DVR_CHAN_GROUP_RECORD_STATUS	6013

[返回目录](#)

5.9.2 设置设备的配置信息 NET_DVR_SetDVRConfig

函数: BOOL NET_DVR_SetDVRConfig(LONG lUserID, DWORD dwCommand, LONG IChannel, 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 5.32
[in]IChannel 通道号, 不同的命令对应不同的取值, 如果该参数无效则置为 0xFFFFFFFF 即可, 详见表 5.32
[in]lpInBuffer 输入数据的缓冲指针, 详见表 5.32
[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 5.32 所示。

表 5.32 参数设置命令

dwCommand 宏定义	dwCommand 含义	IChannel 取值	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_DEVICECFG_V40	设置设备参数(扩展)	无效	NET_DVR_DEVICECFG_V40	1101
NET_DVR_SET_TIMECFG	设置时间参数	无效	NET_DVR_TIME	119
NET_DVR_SET_NETCFG_V30	设置网络参数	无效	NET_DVR_NETCFG_V30	1001
NET_DVR_SET_RECORDCFG_V40	设置录像计划参数	通道号	NET_DVR_RECORD_V40	1009
NET_DVR_SET_IPPARACFG_V40	设置 IP 接入配置参数	组号, 从 0 开始, 每组 64 个通道	NET_DVR_IPPARACFG_V40	1063
NET_DVR_SET_USERCFG_V40	设置用户参数	组号, 从 0 开始, 每组 32 个用户	NET_DVR_USER_V40	6188
NET_DVR_SET_RECORD_CHANNEL_INFO	设置通道录像状态信息	组号, 从 0 开始, 每组 64 个通道	NET_DVR_CHAN_GROUP_RECORD_STATUS	6014

[返回目录](#)

批量参数配置

5.9.3 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 5.33
 - [in] dwCount 一次要获取配置参数的个数，0 和 1 都表示 1 个信息，2 表示 2 个信息，最大 64 个
 - [in] lpInBuffer 配置条件缓冲区，详见表 5.34
 - [in] dwInBufferSize 缓冲区长度
 - [out] lpStatusList 错误信息列表，和要查询的配置一一对应，例如 lpStatusList[2]就对应 lpInBuffer[2]，由用户分配内存，每个错误信息为 4 个字节，参数值：0 或者 1 表示成功，其他值为失败对应的错误号
 - [out] lpOutBuffer 设备返回的参数内容（详见表 5.34），和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 0 值，对应 lpOutBuffer 的内容就是无效的
 - [in] dwOutBufferSize 输出缓冲区大小
- 返回值：** TRUE 表示成功，但不代表每一个配置都成功，哪一个成功，对应查看 lpStatusList[n]值；FALSE 表示全部失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。
- 说明：** 该接口是带有发送数据的批量获取配置信息的通用接口。lpInBuffer 指定需要获取的信息，lpOutBuffer 保存获取得到的 dwCount 个配置信息。不同的 dwCommand 对应不同的结构体和命令号，如表 5.34 所示。

表 5.33 参数批量获取命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_STREAM_SRC_INFO_V40	获取流的来源信息	6024
NET_DVR_GET_STREAM_RECORD_STATUS	获取流录像状态信息	6021
NET_DVR_GET_STREAM_RECORD_INFO	获取录像计划信息(流模式)	6019
NET_DVR_GET_USER_RIGHT_CFG	获取用户权限，dwCount 为 1	6210

表 5.34 批量获取设备参数

dwCommand	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_GET_STREAM_SRC_INFO_V40	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_STREAM_SRC_INFO_V40
NET_DVR_GET_STREAM_RECORD_STATUS	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_STREAM_RECORD_STATUS
NET_DVR_GET_STREAM_RECORD_INFO	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_STREAM_INFO
NET_DVR_GET_USER_RIGHT_CFG	NET_DVR_USER_COND	NET_DVR_USER_RIGHT_CFG

[返回目录](#)

5.9.4 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, 详见表 5.35
 - [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 5.36
 - [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 5.36), 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 5.36 所示。

表 5.35 参数批量设置命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_SET_STREAM_SRC_INFO_V40	设置流的来源信息	6025
NET_DVR_SET_STREAM_RECORD_STATUS	设置流录像状态信息	6022
NET_DVR_SET_STREAM_RECORD_INFO	设置录像计划信息(流模式)	6020
NET_DVR_SET_USER_RIGHT_CFG	设置用户权限, dwCount 为 1	6211

表 5.36 批量设置设备参数

dwCommand	lpInBuffer 对应结构体	lpInParamBuffer 对应结构体
NET_DVR_SET_STREAM_SRC_INFO_V40	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_STREAM_SRC_INFO_V40
NET_DVR_SET_STREAM_RECORD_STATUS	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_STREAM_RECORD_STATUS
NET_DVR_SET_STREAM_RECORD_INFO	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_STREAM_INFO
NET_DVR_SET_USER_RIGHT_CFG	NET_DVR_USER_COND	NET_DVR_USER_RIGHT_CFG

[返回目录](#)

长连接参数配置

5.9.5 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 pUserData)

参数: [in] IUserID 用户 ID 号, NET_DVR_Login_V30 返回值
[in] dwCommand 配置命令, 详见表 5.37
[in] lpInBuffer 输入参数, 具体内容跟配置命令相关, 详见表 5.37
[in] dwInBufferLen 输入缓冲的大小
[in] cbStateCallback 状态回调函数
[in] pUserData 用户数据

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

[out] dwType 配置状态, 具体定义详见表 5.38
[out] lpBuffer 存放数据的缓冲区指针, 具体内容跟 dwCommand、dwType 相关
[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: 调用该接口启动长连接远程配置后, 部分命令还需要调用其他接口获取信息或者设置相关参数, 如表 5.39 所示。

表 5.37 长连接参数配置

dwCommand 宏定义	宏定义值	含义	lpInBuffer 对应结构	cbStateCallback
NET_DVR_ONE_KEY_CONFIG	6212	一键配置 CVR	NULL	NULL
NET_DVR_ONE_KEY_CONFIG_V50	6251	一键配置 CVR(扩展, 支持指定 RAID 类型、热备盘/阵列比例、图片卷比例等条件)	NET_DVR_ONE_KEY_CFG_V50	NULL
NET_DVR_GET_STREAM_INFO	6023	获取已添加的流 ID 信息	NULL	返回状态、进度、查找到的流 ID 信息

表 5.38 回调数据内容

dwCommand 宏定义	dwType	lpInBuffer
NET_DVR_GET_STREAM_INFO	<pre>typedef enum{ NET_SDK_CALLBACK_TYPE_STATUS = 0, //回调状态值 NET_SDK_CALLBACK_TYPE_PROGRESS, //回调进度值 NET_SDK_CALLBACK_TYPE_DATA //回调数据内容 }NET_SDK_CALLBACK_TYPE;</pre>	<p>dwType 为 0 时对应状态值, 详见枚举类型 NET_SDK_CALLBACK_STATUS_NORMAL;</p> <p>dwType 为 2 时对应流 ID 信息: NET_DVR_STREAM_INFO</p> <pre>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;</pre>

表 5.39 后续接口调用

dwCommand 宏定义	含义	后续接口调用
NET_DVR_ONE_KEY_CONFIG	一键配置 CVR	NET_DVR_GetRemoteConfigState
NET_DVR_ONE_KEY_CONFIG_V50	一键配置 CVR 扩展	NET_DVR_GetRemoteConfigState
NET_DVR_GET_STREAM_INFO	获取已添加的流 ID 信息	无

[返回目录](#)

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

函数: BOOL NET_DVR_GetRemoteConfigState(LONG IHandle, void *pState)

参数: [in] IHandle 句柄, NET_DVR_StartRemoteConfig 的返回值
 [out] pState 返回的状态值, 不同的配置命令对应不同的状态取值, 详见表 5.40

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: 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 5.40 所示。

表 5.40 长连接配置状态

dwCommand 宏定义	宏定义值	含义
NET_DVR_ONE_KEY_CONFIG	6212	一键配置 CVR
NET_DVR_ONE_KEY_CONFIG_V50	6251	一键配置 CVR 扩展
pState 状态宏定义	宏定义值	含义
NET_SDK_OKC_STATUS_SUCCESS	1000	一键配置成功
NET_SDK_OKC_STATUS_FAILED	1002	一键配置失败
NET_SDK_OKC_STATUS_START_CONFIG	1003	开始配置
NET_SDK_OKC_CHECK_HD	1004	检测磁盘
NET_SDK_OKC_INIT_HD	1005	初始化磁盘
NET_SDK_OKC_CREATE_RAID_OR_SINGLE_DISK	1006	创建阵列或者单盘模式
NET_SDK_OKC_INIT_CVR_SERVICE	1007	初始化 CVR 服务
NET_SDK_OKC_CREATE_RECORD_VOLUME	1008	创建录像卷
NET_SDK_OKC_ERR_LOAD_CONF_FAILED	1009	加载配置文件失败
NET_SDK_OKC_ERR_NOT_SUPPORT_RAID_LEVLE	1010	不支持此种类型的 RAID
NET_SDK_OKC_ERR_CONFIGURATION_CONFLICT	1011	系统已经存在 RAID 或存储池
NET_SDK_OKC_ERR_GET_DISK_INFO_FAILED	1012	获取磁盘信息失败
NET_SDK_OKC_ERR_CHECK_DISK_FAILED	1013	检测磁盘失败
NET_SDK_OKC_ERR_INIT_DISK_FAILED	1014	初始化磁盘失败
NET_SDK_OKC_ERR_DISK_CAPACITY_SMALL	1015	磁盘总容量不足
NET_SDK_OKC_ERR_BOTH_SV_NS	1016	同时存在 SV 盘和 NS 盘

NET_SDK_OKC_ERR_CREATE_RAID_FAILED	1017	创建 RAID 失败
NET_SDK_OKC_ERR_GET_RAID_FAILED	1018	获取 RAID 失败
NET_SDK_OKC_ERR_CREATE_SPARE_FAILED	1019	创建热备盘失败
NET_SDK_OKC_ERR_CREATE_STORAGE_POOL_FAILED	1020	创建存储池失败
NET_SDK_OKC_ERR_GET_POOL_INFO_FAILED	1021	获取存储池信息失败
NET_SDK_OKC_ERR_CREATE_LUN_FAILED	1022	创建 lun 卷失败
NET_SDK_OKC_ERR_GET_LUN_INFO_FAILED	1023	获取 lun 信息失败
NET_SDK_OKC_ERR_CREATE_BACKUP_FAILED	1024	创建预留卷失败
NET_SDK_OKC_ERR_GET_BACKUP_FAILED	1025	获取预留卷失败
NET_SDK_OKC_ERR_CREATE_PRIVATE_LUN_FAILED	1026	创建私有卷失败
NET_SDK_OKC_ERR_CREATE_RV_FAILED	1027	创建录像卷失败
NET_SDK_OKC_ERR_CREATE_ARCH_RV_FAILED	1028	创建存档卷失败
NET_SDK_OKC_ERR_START_CVR_SERVICE_FAILED	1029	开启 CVR 服务失败
NET_SDK_OKC_ERR_SAPRE_NUM_EXCEED	1101	热备盘个数过多
NET_SDK_OKC_ERR_CREATE_PIC_VOLUME_FAILED	1102	创建图片卷失败

[返回目录](#)

5.9.7 关闭长连接 **NET_DVR_StopRemoteConfig**

函数: BOOL NET_DVR_StopRemoteConfig(LONG IHandle)

参数: [in] IHandle 句柄, NET_DVR_StartRemoteConfig 的返回值

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

说明: 关闭长连接配置接口所创建的句柄, 释放资源。

[返回目录](#)

5.10 录像回传功能

5.10.1 批量获取参数信息 **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 5.41
[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 5.42
[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 5.42），和要查询的监控点一一对应。如果某个监控点对应的 `lpStatusList` 信息为大于 0 值，对应 `lpOutBuffer` 的内容就是无效的

[in] dwOutBufferSize 输出缓冲区大小

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

说明： 该接口是带有发送数据的批量获取配置信息的通用接口。`lpInBuffer` 指定需要获取的信息，`lpOutBuffer` 保存获取到的 `dwCount` 个配置信息。不同的 `dwCommand` 对应不同的结构体和命令号，如表 5.42 所示。

表 5.41 参数批量获取命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_RECORD_PASSBACK_PLAN_CFG	获取录像回传计划配置，dwCount 设为 1	6238

表 5.42 批量获取设备参数

dwCommand	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_GET_RECORD_PASSBACK_PLAN_CFG	NET_DVR_STREAM_INFO	NET_DVR_RECORD_PASSBACK_SCH_CFG

[返回目录](#)

5.10.2 批量设置参数信息 `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 5.43
- [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 5.44
- [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 5.44），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 5.44 所示。

表 5.43 参数批量设置命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_ADD_RECORD_PASSBACK_TASK_MANUAL	手动添加录像回传任务	6234
NET_DVR_DEL_RECORD_PASSBACK_TASK_MANUAL	删除手动录像回传任务	6237
NET_DVR_SET_RECORD_PASSBACK_PLAN_CFG	设置录像回传计划配置，dwCount 设为 1	6239

表 5.44 批量设置设备参数

dwCommand	lpInBuffer 对应结构体	lpInParamBuffer 对应结构体
NET_DVR_ADD_RECORD_PASSBACK_TASK_MANUAL	dwCount 个 NET_DVR_ADD_RECORD_PASSBACK_MANUAL_COND	dwCount 个 NET_DVR_ADD_RECORD_PASSBACK_MANUAL_CFG
NET_DVR_DEL_RECORD_PASSBACK_TASK_MANUAL	dwCount 个 NET_DVR_DEL_RECORD_PASSBACK_MANUAL_COND	dwCount 个 NET_DVR_DEL_RECORD_PASSBACK_MANUAL_CFG
NET_DVR_SET_RECORD_PASSBACK_PLAN_CFG	NET_DVR_STREAM_INFO	NET_DVR_RECORD_PASSBACK_SCH_CFG

[返回目录](#)

5.10.3 获取设备参数 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 5.45
[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 5.46

返回值： TRUE 表示成功，FALSE 表示失败。接口返回失败请调用 [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 5.46 所示。

表 5.45 获取设备参数

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_GET_RECORD_PASSBACK_BASIC_CFG	获取回传功能基础配置	6249
NET_DVR_GET_RECORD_PASSBACK_HISTORY_PLAN_CFG	获取回传历史录像计划配置	6256

表 5.46 获取设备参数

dwCommand 宏定义	lpCondBuffer	lpOutBuffer
NET_DVR_GET_RECORD_PASSBACK_BASIC_CFG	NULL	NET_DVR_REC_PASSBACK_BASIC_CFG
NET_DVR_GET_RECORD_PASSBACK_HISTORY_PLAN_CFG	NET_DVR_STREAM_INFO	NET_DVR_RECORD_PASSBACK_HISTORY_PLAN_CFG

[返回目录](#)

5.10.4 设置设备参数 **NET_DVR_SetSTDConfig**

- 函数:** BOOL NET_DVR_SetSTDConfig(LONG IUserID, DWORD dwCommand, LPNET_DVR_STD_CONFIG lpConfigParam)
- 参数:** [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 设备配置命令, 详见表 5.47
[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 5.48
- 返回值:** TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
- 说明:** 设置配置参数时, lpConfigParam 结构体里面的 lpOutBuffer 无效, 设为 NULL。对于不同的配置功能 (dwCommand), lpConfigParam 中的 lpCondBuffer、lpInBuffer 分别对应不同的内容, 具体如表 5.48 所示。

表 5.47 设置设备参数

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_SET_RECORD_PASSBACK_BASIC_CFG	设置回传功能基础配置	6250
NET_DVR_SET_RECORD_PASSBACK_HISTORY_PLAN_CFG	设置回传历史录像计划配置	6257

表 5.48 设置设备参数

dwCommand 宏定义	lpCondBuffer	lpInBuffer
NET_DVR_SET_RECORD_PASSBACK_BASIC_CFG	NULL	NET_DVR_REC_PASSBACK_BASIC_CFG
NET_DVR_SET_RECORD_PASSBACK_HISTORY_PLAN_CFG	NET_DVR_STREAM_INFO	NET_DVR_RECORD_PASSBACK_HISTORY_PLAN_CFG

[返回目录](#)

5.10.5 手动添加回传任务 **NET_DVR_SetDeviceConfigEx**

- 函数:** BOOL NET_DVR_SetDeviceConfigEx(LONG iUserID, DWORD dwCommand, DWORD dwCount, NET_DVR_IN_PARAM *lpInParam, NET_DVR_OUT_PARAM *lpOutParam)
- 参数:** [in] IUserID 登录主控板, NET_DVR_Login_V40 的返回值
[in] dwCommand 设备配置命令, 详见表 5.49
[in] dwCount 批量配置个数, 0 和 1 都表示 1 个, 依次递增, 最大 64 个
[in] lpInParam 输入参数缓冲区, 详见结构体: [NET_DVR_IN_PARAM](#), 其中参数取值根据不同的 dwCommand 而不同, 详见表 5.50
[in] lpOutParam 输出参数缓冲区, 详见结构体: [NET_DVR_OUT_PARAM](#), 其中参数取值根据不同的 dwCommand 而不同, 详见表 5.50

表 5.49 批量配置

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_ADD_RECORD_PASSBACK_TASK_MANUAL_V50	手动添加录像回传任务 (返回任务 ID)	6254

- 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监控点配置信息的通用接口扩展接口，支持设置接收数据超时时间。

表 5.50 批量参数配置结构

dwCommand 宏定义	取值含义
NET_DVR_ADD_RECORD_PASSBACK_TASK_MANUAL_V50	手动添加录像回传任务（返回任务 ID）
输入输出缓冲区参数	参数取值
lpInParam->struCondBuf.pBuf	输入条件缓冲区，dwCount 个 NET_DVR_ADD_RECORD_PASSBACK_MANUAL_COND
lpInParam->struCondBuf.nLen	输入条件缓冲区长度
lpInParam->struInParamBuf.pBuf	输入参数缓冲区，dwCount 个 NET_DVR_ADD_RECORD_PASSBACK_MANUAL_CFG
lpInParam->struInParamBuf.nLen	输入参数缓冲区长度
lpInParam->dwRecvTimeout	数据接收超时时间，填 0 采用接口默认超时
lpOutParam->struOutBuf.pBuf	输出参数缓冲区，dwCount 个 NET_DVR_RECORD_PASSBACK_TASK_CFG
lpOutParam->struOutBuf.nLen	输出参数缓冲区长度
lpOutParam->lpStatusList	状态缓冲区，对应 dwCount 个参数配置的结果，例如 lpStatusList[2]就对应 lpInParam->struInParamBuf.pBuf[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

[返回目录](#)

5.10.6 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 5.51
[in]lpInBuffer 输入参数，具体内容跟控制命令相关，详见表 5.51
[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 5.51。

表 5.51 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_RECORD_PASSBACK_TASK_MANUAL_CTRL	6236	控制手动添加的录像回传任务	NET_DVR_RECORD_PASSBACK_MANUAL_CTRL

[返回目录](#)

5.10.7 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 5.52
[in] lpInBuffer 输入参数, 具体内容跟配置命令相关, 详见表 5.52
[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 dwCommand、dwType 相关
[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: 调用该接口启动长连接远程配置后, 还需要调用 [NET_DVR_GetNextRemoteConfig](#) 接口获取信息。

表 5.52 长连接参数配置

dwCommand 宏定义	宏定义值	含义	lpInBuffer 对应结构	cbStateCallback
NET_DVR_GET_ALL_RECORD_PASSBACK_TASK_MANUAL	6235	获取所有手动添加的录像回传任务	NET_DVR_RECORD_PASSBACK_MANUAL_COND	NULL

[返回目录](#)

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

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

参 数: [in] IHandle 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 5.54
[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 5.55。获取错误码调用 [NET_DVR_GetLastError](#)。

表 5.53 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_NEXT_STATUS_NEED_WAIT	1001	需等待设备发送数据, 继续调用 NET_DVR_GetNextRemoteConfig
NET_SDK_GET_NEXT_STATUS_FINISH	1002	数据全部取完, 可调用 NET_DVR_StopRemoteConfig 结束长连接
NET_SDK_GET_NEXT_STATUS_FAILED	1003	出现异常, 可调用 NET_DVR_StopRemoteConfig 结束长连接

说 明: 调用 NET_DVR_StartRemoteConfig 时传入不同的命令号(dwCommand), lpOutBuff 对应不同的结

构体，如表 5.54 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 5.54 长连接参数获取

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer
<code>NET_DVR_GET_ALL_RECORD_PASSBACK_TASK_MANUAL</code>	6235	获取所有手动添加的录像回传任务	<code>NET_DVR_RECORD_PASSBACK_MANUAL_TASK_RET</code>

[返回目录](#)

5.10.9 关闭长连接 `NET_DVR_StopRemoteConfig`

函数: `BOOL NET_DVR_StopRemoteConfig(LONG IHandle)`

参数: [in] IHandle 句柄, `NET_DVR_StartRemoteConfig` 的返回值

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

说明: 关闭长连接配置接口所创建的句柄, 释放资源。

[返回目录](#)

5.11 设备热备管理

N+1 模式

5.11.1 启动长连接远程配置 `NET_DVR_StartRemoteConfig`

函数: `LONG NET_DVR_StartRemoteConfig(LONG IUserID, DWORD dwCommand, LPVOID lpInBuffer, DWORD dwInBufferLen, fRemoteConfigCallback cbStateCallback, LPVOID pUserData)`

参数: [in] IUserID 用户 ID 号, `NET_DVR_Login_V30` 返回值

[in] dwCommand 配置命令, 详见表 5.55

[in] lpInBuffer 输入参数, 具体内容跟配置命令相关, 详见表 5.55

[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 dwCommand、dwType 相关

[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：调用该接口启动长连接远程配置后，部分命令还需要调用其他接口获取信息或者设置相关参数，如表 5.56 所示。

表 5.55 长连接参数配置

dwCommand 宏定义	宏定义值	含义	lpInBuffer 对应结构	cbStateCallback
NET_DVR_GET_N_PLUS_ONE_DEVICE_INFO	6222	获取 N+1 设备信息	NULL	NULL

表 5.56 后续接口调用

dwCommand 宏定义	含义	后续接口调用
NET_DVR_GET_N_PLUS_ONE_DEVICE_INFO	获取 N+1 设备信息	NET_DVR_GetNextRemoteConfig

[返回目录](#)

5.11.2 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 5.58
 [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 5.57。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。

表 5.57 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_NEXT_STATUS_NEED_WAIT	1001	需等待设备发送数据，继续调用 NET_DVR_GetNextRemoteConfig
NET_SDK_GET_NEXT_STATUS_FINISH	1002	数据全部取完，可调用 NET_DVR_StopRemoteConfig 结束长连接
NET_SDK_GET_NEXT_STATUS_FAILED	1003	出现异常，可调用 NET_DVR_StopRemoteConfig 结束长连接

说明：调用 NET_DVR_StartRemoteConfig 时传入不同的命令号(dwCommand)，lpOutBuff 对应不同的结构体，如表 5.58 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 5.58 长连接参数获取

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer
NET_DVR_GET_N_PLUS_ONE_DEVICE_INFO	6222	获取 N+1 设备信息	NET_DVR_N_PLUS_ONE_DEVICE_INFO

[返回目录](#)

5.11.3 关闭长连接 **NET_DVR_StopRemoteConfig**

函数: BOOL NET_DVR_StopRemoteConfig(LONG IHandle)
参数: [in] IHandle 句柄, NET_DVR_StartRemoteConfig 的返回值
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说明: 关闭长连接配置接口所创建的句柄, 释放资源。

[返回目录](#)

5.11.4 获取设备的配置信息 **NET_DVR_GetDVRConfig**

函数: BOOL NET_DVR_GetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpOutBuffer, DWORD dwOutBufferSize, LPDWORD lpBytesReturned)
参数: [in] IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in] dwCommand 设备配置命令, 详见表 5.59
[in] IChannel 通道号, 不同的命令对应不同的取值, 如果该参数无效则置为 0xFFFFFFFF 即可, 详见表 5.59
[out] lpOutBuffer 接收数据的缓冲指针, 详见表 5.59
[in] dwOutBufferSize 接收数据的缓冲长度(以字节为单位), 不能为 0
[out] lpBytesReturned 实际收到的数据长度指针, 不能为 NULL
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
说明: 不同的获取功能对应不同的结构体和命令号, 如表 5.59 所示。

表 5.59 获取设备参数

dwCommand 宏定义	dwCommand 含义	IChannel 取值	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_N_PLUS_ONE_WORK_MODE	获取 N+1 工作模式	无效	NET_DVR_N_PLUS_ONE_WORK_MODE_CFG	6168
NET_DVR_GET_MAX_MACHINE_NUM_CFG	获取备机最大个数	无效	NET_DVR_MACHINE_MAX_NUM_CFG	6214

[返回目录](#)

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

函数: BOOL NET_DVR_SetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpInBuffer, DWORD dwInBufferSize)
参数: [in] IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in] dwCommand 设备配置命令, 详见表 5.60
[in] IChannel 通道号, 不同的命令对应不同的取值, 如果该参数无效则置为 0xFFFFFFFF 即可, 详见表 5.60
[in] lpInBuffer 输入数据的缓冲指针, 详见表 5.60
[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 5.60 所示。

表 5.60 参数设置命令

dwCommand 宏定义	dwCommand 含义	lChannel 取值	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_N_PLUS_ONE_WORK_MODE	设置 N+1 工作模式	无效	NET_DVR_N_PLUS_ONE_WORK_MODE_CFG	6169
NET_DVR_SET_MAX_MACHINE_NUM_CFG	设置设备机最大个数	无效	NET_DVR_MACHINE_MAX_NUM_CFG	6215

[返回目录](#)

5.11.6 远程控制 NET_DVR_RemoteControl

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

参数: [in] lUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in] dwCommand 控制命令, 详见表 5.61
[in] lpInBuffer 输入参数, 具体内容跟控制命令相关, 详见表 5.61
[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 5.61。

表 5.61 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_ADD_DEVICE	6216	N+1 模式添加设备	NET_DVR_N_PLUS_ONE_DEVICE_PARAM
NET_DVR_DEL_DEVICE	6217	N+1 模式删除设备	NET_DVR_N_PLUS_ONE_DEVICE_PARAM

[返回目录](#)

N+0 模式

5.11.7 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 5.62
[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 5.63
[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 5.63), 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 5.63 所示。

表 5.62 参数批量获取命令

dwCommand 宏定义	含义	宏定义值
NET_DVR_GET_RELOCATE_INFO	获取 N+0 模式下重定向信息	6026

表 5.63 批量获取设备参数

dwCommand	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_GET_RELOCATE_INFO	dwCount 个 NET_DVR_STREAM_INFO	dwCount 个 NET_DVR_RELOCATE_INFO

[返回目录](#)

5.12 存储设备管理

存储参数配置

5.12.1 获取设备的配置信息 **NET_DVR_GetDVRConfig**

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

参数: [in]lUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 设备配置命令, 详见表 5.64
[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 5.65
[out]lpOutBuffer 接收数据的缓冲指针, 详见表 5.65
[in]dwOutBufferSize 接收数据的缓冲长度(以字节为单位), 不能为 0
[out]lpBytesReturned 实际收到的数据长度指针, 不能为 NULL

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

说明: 不同的获取功能对应不同的结构体和命令号, 如表 5.65 所示。

表 5.64 参数获取命令

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_GET_HDCFG_V40	获取硬盘(录像卷)管理参数	6122
NET_DVR_GET_DATA_CALLBACK_CFG	获取数据回迁状态	6218
NET_DVR_GET_DEV_STORAGE_CFG	获取设备存储信息	6240
NET_DVR_GET_HDVOLUME_CFG	获取硬盘卷信息	4155

表 5.65 获取设备参数

dwCommand 宏定义	Ichannel 取值	lpOutBuffer 对应结构体
NET_DVR_GET_HDCFG_V40	组号, 从 0 开始, 每组 33 个硬盘	NET_DVR_HDCFG
NET_DVR_GET_DATA_CALLBACK_CFG	无效	NET_DVR_DATA_CALLBACK_CFG
NET_DVR_GET_DEV_STORAGE_CFG	无效	NET_DVR_DEV_STORAGE_CFG
NET_DVR_GET_HDVOLUME_CFG	无效	NET_DVR_HARD_DISK_VOLUME_INFO

[返回目录](#)

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

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

参数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 设备配置命令, 详见表 5.66
[in]IChannel 通道号, 不同的命令对应不同的取值, 如果该参数无效则置为 0xFFFFFFFF 即可, 详见表 5.67
[in]lpInBuffer 输入数据的缓冲指针, 详见表 5.67
[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 5.67 所示。

表 5.66 参数设置命令

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_SET_HDCFG_V40	设置硬盘(录像卷)管理参数	6123
NET_DVR_SET_DATA_CALLBACK_CFG	设置数据回迁状态	6219
NET_DVR_SET_HDVOLUME_CFG	设置硬盘卷信息	4156

表 5.67 设置设备参数

dwCommand 宏定义	Ichannel 取值	lpInBuffer 对应结构体
NET_DVR_SET_HDCFG_V40	组号, 从 0 开始, 每组 33 个硬盘	NET_DVR_HDCFG
NET_DVR_SET_DATA_CALLBACK_CFG	无效	NET_DVR_DATA_CALLBACK_CFG
NET_DVR_SET_HDVOLUME_CFG	无效	NET_DVR_HARD_DISK_VOLUME_INFO

[返回目录](#)

5.12.3 数据文件配置及异地备份配置 **NET_DVR_STDXMLConfig**

函数: BOOL NET_DVR_STDXMLConfig(LONG IUserID, NET_DVR_XML_CONFIG_INPUT* lpInputParam, NET_DVR_XML_CONFIG_OUTPUT* lpOutputParam)

参数: [in] IUserID NET_DVR_Login_V40 登录接口的返回值
[in] lpInputParam 输入参数, 详见 NET_DVR_XML_CONFIG_INPUT
[out] lpOutputParam 输出参数, 详见 NET_DVR_XML_CONFIG_OUTPUT

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

错误码判断出错原因。

说明：

Remarks

该接口为 ISAPI 协议透传接口，不同功能对应输入参数和输出参数具体如下表所示。其中所有输入输出参数均为 XML 格式，详细内容请见《设备网络 SDK 使用手册.chm》。

功能描述	IpInputParam->IpRequestUrl	IpInputParam->IpInBuffer	IpOutputParam->IpOutBuffer
数据文件传输速率配置能力集	GET /ISAPI/ContentMgmt/FileData/TransferSpeedConfiguration/capabilities	NULL	TransferSpeedConfiguration (文件传输配置能力 XML 格式)
获取数据文件传输速率	GET /ISAPI/ContentMgmt/FileData/TransferSpeedConfiguration	NULL	TransferSpeedConfiguration (文件传输配置参数 XML 格式)
设置数据文件传输速率	PUT /ISAPI/ContentMgmt/FileData/TransferSpeedConfiguration	TransferSpeedConfiguration (文件传输配置参数 XML 格式)	NULL
数据文件锁定配置能力集	GET /ISAPI/ContentMgmt/FileData/LockConfiguration/capabilities	NULL	LockConfiguration (锁定配置能力 XML 格式)
设置数据文件锁定	PUT /ISAPI/ContentMgmt/FileData/LockConfiguration	NULL	LockConfiguration (锁定配置参数 XML 格式)
获取数据文件信息能力集	GET /ISAPI/ContentMgmt/FileData/SearchFile/capabilities	NULL	SearchFileCondition (数据文件信息获取能力 XML 格式)
获取数据文件信息	POST /ISAPI/ContentMgmt/FileData/SearchFile	SearchFileCondition (数据文件查询条件)	SearchFileResult (数据文件查询结果)
异地备份备机信息配置能力集	GET /ISAPI/ContentMgmt/RemoteBackup/BackupMachineInfo/capabilities	NULL	BackupMachineInfo (异地备份能力集 XML 格式)
获取异地备份备机信息配置	GET /ISAPI/ContentMgmt/RemoteBackup/BackupMachineInfo	NULL	BackupMachineInfo (备机信息 XML 格式)
设置异地备份备机信息配置	PUT /ISAPI/ContentMgmt/RemoteBackup/BackupMachineInfo	BackupMachineInfo (备机信息 XML 格式)	NULL
异地备份任务能力集	GET /ISAPI/ContentMgmt/RemoteBackup/Task/capabilities	NULL	RemoteBackupTaskCap (异地备份任务能力集 XML 格式)
添加异地备份任务	POST /ISAPI/ContentMgmt/RemoteBackup/addTask	RemoteBackupAddTask (XML 格式)	RemoteBackupTaskInfo (XML 格式)
搜索异地备份任务	POST /ISAPI/ContentMgmt/RemoteBackup/SearchTask	RemoteBackupSearchTask (XML 格式)	RemoteBackupTaskData (XML 格式)

		格式)	
删除异地备份单个任务	DELETE /ISAPI/ContentMgmt/RemoteBackup/deleteTasks/<taskID>	NULL	NULL
删除异地备份所有任务	DELETE /ISAPI/ContentMgmt/RemoteBackup/deleteTasks	NULL	NULL

[返回目录](#)

5.12.4 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.
 [in] lpInBuffer 不同的 dwUploadType, 输入参数不同, 详见表 5.
 [in] dwInBufferSize 输入缓冲区大小
 [in] sFileName 上传文件的绝对路径 (包括文件名)
 [out] lpOutBuffer 输出参数, 不同的 dwUploadType, 输出参数不同, 详见表 5.
 [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 5.68 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_FILEVOLUME_DATA	38	上传文件卷数据文件	NET_DVR_UPLOAD_FILEVOLUME_PRARAM	NULL

[返回目录](#)

5.12.5 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; 6- 硬盘错误; 7- 无审讯文件存放盘; 8- 容量不足; 9- 设备资源不足; 10- 文件个数超过 40; 12- 文件类型错误; 19- 文件格式不正确; 20- 文件内容不正确; 21- 上传音频采样率不支持; 34- 安全校验密钥错误。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

[返回目录](#)

5.12.6 获取当前数据上传的结果信息 **NET_DVR_GetUploadResult**

函数: BOOL NET_DVR_GetUploadResult (LONG IUploadHandle, LPVOID lpOutBuffer, DWORD

dwOutBufferSize)

参 数: [in] IUploadHandle 文件上传的句柄, NET_DVR_UploadFile_V40 的返回值
 [out] lpOutBuffer 结果信息缓冲区, 详见表 5.
 [in] dwInBufferSize 输出缓冲区大小

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

说 明:

表 5.69 文件上传结果

dwUploadType 宏定义	功能描述	lpOutBuffer 对应结构体
UPLOAD_FILEVOLUME_DATA	上传文件卷数据文件	NET_DVR_UPLOAD_FILE_RET

[返回目录](#)

5.12.7 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](#)

5.12.8 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.
 [in] lpInBuffer 输入参数。不同的 dwUploadType, 输入参数不同, 详见表 5.
 [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: 上层使用时可选用数据回调或文件路径的方式二选一进行下载, 输入参数中 fnDownloadDataCB、sFileName 同时只能有一个参数生效, sFileName 不为空时会直接将数据写入指定文件中; 为空时则会由回调函数回调数据供用户自行处理。

表 5.70 文件下载类型

dwDownloadType	取值	含义	lpInBuffer 对应结构体
NET_SDK_DOWNLOAD_FILEVOLUME_DATA	20	下载文件卷数据文件	NET_DVR_DOWNLOAD_FILEVOLUME_PARAM

[返回目录](#)

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

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

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

[out] pProgress 返回的进度值，取值范围：0~100

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

说 明：

[返回目录](#)

5.12.10 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](#)

硬盘格式化

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

函 数： LONG NET_DVR_FormatDisk(LONG IUserID, LONG IDiskNumber)

参 数： [in] IUserID NET_DVR_Login_V40 的返回值

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

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

说 明： 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](#)

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

函 数： BOOL NET_DVR_GetFormatProgress(LONG IFormatHandle, 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](#)

5.12.13 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](#)

虚拟磁盘(逻辑卷)操作

5.12.14 获取虚拟磁盘列表 **NET_DVR_GetVDList**

函数: BOOL NET_DVR_GetVDList(LONG IUserID, LPNET_DVR_VD_LIST lpVDList)

参数: [in]IUserID NET_DVR_Login_V40 的返回值

[out]lpVDList 虚拟磁盘信息，包括磁盘个数，每个磁盘的名称、容量、槽位 ID 号、状态等

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: 该接口用于获取存储设备的逻辑卷列表。

[返回目录](#)

5.12.15 创建虚拟磁盘 **NET_DVR_CreateVD**

函数: BOOL NET_DVR_CreateVD(LONG IUserID, LPNET_DVR_OPERATE_VD_PARAM lpVDParam)

参数: [in]IUserID NET_DVR_Login_V40 的返回值

[in]lpVDParam 虚拟磁盘参数，包括阵列 ID、槽位 ID、磁盘大小、名称等

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:

[返回目录](#)

5.12.16 删除虚拟磁盘 **NET_DVR_DeIVD**

函数: BOOL NET_DVR_DeIVD(LONG IUserID, DWORD dwID)

参数: [in]IUserID NET_DVR_Login_V40 的返回值

[in]dwID 虚拟磁盘 ID

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

说明:

[返回目录](#)

5.13 布防、撤防

设置报警等信息上传的回调函数

5.13.1 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。
[out]pAlarmer 报警设备信息
[out]pAlarmInfo 报警信息，详见表 5。
[out]dwBufLen 报警信息缓存大小
[out]pUser 用户数据

表 5.71 报警信息类型

ICommand 宏定义	宏定义值	含义
COMM_ALARM_DEVICE	0x4004	CVR 设备报警信息上传，由于通道值大于 256 而扩展
COMM_ALARM_DEVICE_V40	0x4009	CVR 设备报警信息扩展(增加报警信息子结构)
COMM_ALARM_CVR	0x4005	CVR 外部报警信息上传

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。

表 5.72 报警信息结构

消息类型 ICommand	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM_DEVICE	CVR 设备报警信息，由于通道值大于 256 而扩展	NET_DVR_ALARMINFO_DEV
COMM_ALARM_DEVICE_V40	CVR 设备报警信息扩展(增加报警信息子结构)	NET_DVR_ALARMINFO_DEV_V40
COMM_ALARM_CVR	CVR 外部报警信息	NET_DVR_CVR_ALARM

[返回目录](#)

布防撤防

5.13.2 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](#)

5.13.3 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](#)

5.14 监听报警

5.14.1 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.
[out] pAlarmer 报警设备信息
[out] pAlarmInfo 报警信息, 详见表 5.
[out] dwBufLen 报警信息缓存大小
[out] pUser 用户数据

表 5.73 报警信息类型

ICommand 宏定义	宏定义值	含义
COMM_ALARM_DEVICE	0x4004	CVR 设备报警信息上传, 由于通道值大于 256 而扩展
COMM_ALARM_DEVICE_V40	0x4009	CVR 设备报警信息扩展(增加报警信息子结构)
COMM_ALARM_CVR	0x4005	CVR 外部报警信息上传

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。

表 5.74 报警信息结构

消息类型 ICommand	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM_DEVICE	CVR 设备报警信息，由于通道值大于 256 而扩展	NET_DVR_ALARMINFO_DEV
COMM_ALARM_DEVICE_V40	CVR 设备报警信息扩展(增加报警信息子结构)	NET_DVR_ALARMINFO_DEV_V40
COMM_ALARM_CVR	CVR 外部报警信息	NET_DVR_CVR_ALARM

- 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](#)

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

函数： BOOL NET_DVR_StopListen_V30(LONG IListenHandle)

参数： [in]IListenHandle 监听句柄，NET_DVR_StartListen_V30 的返回值

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

说明：

[返回目录](#)

5.15 手动录像

5.15.1 远程手动启动设备录像 **NET_DVR_StartManualRecord**

函数： BOOL NET_DVR_StartManualRecord(LONG IUserID, LPNET_DVR_MANUAL_RECORD_PARA pRecordPara)

参数： [in]IUserID NET_DVR_Login_V40 的返回值

[in]pRecordPara 手动录像参数，包括通道号或流 ID、录像类型等

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：

[返回目录](#)

5.15.2 远程手动停止设备录像 **NET_DVR_StopManualRecord**

函数： BOOL NET_DVR_StopManualRecord(LONG IUserID, LPNET_DVR_STREAM_INFO pIDInfo)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]pIDInfo 通道号或流 ID 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](#)

5.16 设备维护管理

在线用户信息

5.16.1 获取设备的配置信息 **NET_DVR_GetDVRConfig**

函 数: BOOL NET_DVR_GetDVRConfig(LONG IUserID, DWORD dwCommand, LONG IChannel, LPVOID lpOutBuffer, DWORD dwOutBufferSize, LPDWORD lpBytesReturned)
参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V40 的返回值
[in]dwCommand 设备配置命令, 详见表 5.
[in]IChannel 通道号, 不同的命令对应不同的取值, 如果该参数无效则置为 0xFFFFFFFF 即可, 详见表 5.
[out]lpOutBuffer 接收数据的缓冲指针, 详见表 5.
[in]dwOutBufferSize 接收数据的缓冲长度(以字节为单位), 不能为 0
[out]lpBytesReturned 实际收到的数据长度指针, 不能为 NULL
返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。
说 明: 不同的获取功能对应不同的结构体和命令号, 如表 5.所示。

表 5.75 参数获取命令

dwCommand 宏定义	dwCommand 含义	IChannel	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_ONLINE_USER_CFG	获取在线用户参数	无效	NET_DVR_ONLINE_USER_INFO	6241

[返回目录](#)

设备在线状态

5.16.2 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.
[in]lpInBuffer 输入参数, 具体内容跟控制命令相关, 详见表 5.
[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 5.76 远程控制命令

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

[返回目录](#)

远程升级

5.16.3 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](#)

5.16.4 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](#)

5.16.5 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](#)

5.16.6 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](#)

5.16.7 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.所示。

表 5.77 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](#)

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

函 数： BOOL NET_DVR_CloseUpgradeHandle(LONG IUpgradeHandle)

参 数： [in]IUpgradeHandle NET_DVR_Upgrade 的返回值

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

说 明：

[返回目录](#)

日志查找

5.16.9 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 IUserID, LONG ISelectMode, DWORD dwMajorType, DWORD dwMinorType, LPNET_DVR_TIME lpStartTime, LPNET_DVR_TIME lpStopTime, BOOL bOnlySmart = FALSE)

参数: [in]IUserID NET_DVR_Login_V40 的返回值
[in]ISelectMode 查询方式: 0- 全部, 1- 按类型, 2- 按时间, 3- 按时间和类型
[in]dwMajorType 日志主类型 (S.M.A.R.T 搜索时无效), 0 表示全部类型, 其他类型定义见表 5.
[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.~表 5.
[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 5.78 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 5.79 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 5.80 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_FAILED	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
--------------------------------	------	-----------

表 5.81 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	远程导出录象文件
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 5.82 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](#)

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

函数: LONG NET_DVR_FindNextLog_V30(LONG ILogHandle, LPNET_DVR_LOG_V30 lpLogData)

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

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](#)

5.16.11 释放查找日志的资源 **NET_DVR_FindLogClose_V30**

函数: BOOL NET_DVR_FindLogClose_V30(LONG ILogHandle)

参 数: [in]ILogHandle 日志查找句柄, NET_DVR_FindDVRLog_V30 的返回值
 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说 明:

[返回目录](#)

恢复设备默认参数

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

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

说 明:

[返回目录](#)

远程重启

5.16.13 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](#)

5.16.14 重启 CVR 服务 **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.
 [in]lpInBuffer 输入参数, 具体内容跟控制命令相关, 详见表 5.
 [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.所示。

表 5.83 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_RESTART_SERVICE	6213	重启 CVR 服务	NULL

[返回目录](#)

导入/导出配置文件

5.16.15 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](#)

5.16.16 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](#)

5.16.17 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](#)

5.16.18 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](#)

5.17 获取 IPC 协议列表

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

函 数: BOOL NET_DVR_GetIPCProtoList(LONG IUserID, LPNET_DVR_IPC_PROTO_LIST IpProtoList)

参 数: [in]IUserID NET_DVR_Login_V40 的返回值
[out]IpProtoList 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](#)

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_CREATE_SOCKET_ERROR	44	创建 SOCKET 出错。
NET_DVR_SET_SOCKET_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_ALIAS_DUPLICATE	150	别名重复（EasyDDNS 的配置）
NET_DVR_DEV_NET_OVERFLOW	800	网络流量超过设备能力上限
NET_DVR_STATUS_RECORDFILE_WRITING_NOT_LOCK	801	录像文件在录像，无法被锁定
NET_DVR_STATUS_CANT_FORMAT_LITTLE_DISK	802	由于硬盘太小无法格式化
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
NET_DVR_ERR_OPER_NOT_ALLOWED	824	设备处于 N+0 被接管状态，不允许该操作。

NET_DVR_ERR_NEED_RELOCATE	825	设备处于 N+0 被接管状态，需要获取重定向信息，再重新操作。
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。

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_EXCEPT	810	工作机 CVR 服务异常
NET_SDK_ERR_RD_SERVICE_EXCEPT	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_ANALYZE_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_DESCRIPTERROR	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_DESCRIBERCVTIMEOUT	413	RTSP DESCRIBE 接收超时
NET_DVR_RTSP_DESCRIBERCVDATALOST	414	RTSP DESCRIBE 接收数据错误
NET_DVR_RTSP_DESCRIBERCVERROR	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_SETUPSENDTIMEOUT	421	RTSP SETUP 发送超时
NET_DVR_RTSP_SETUPSENDERROR	422	RTSP SETUP 发送错误
NET_DVR_RTSP_SETUPRCVTIMEOUT	423	RTSP SETUP 接收超时
NET_DVR_RTSP_SETUPRCVDATALOST	424	RTSP SETUP 接收数据错误
NET_DVR_RTSP_SETUPRCVERROR	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_PLAYERROR	430	RTSP PLAY 交互错误
NET_DVR_RTSP_PLAYSENDTIMEOUT	431	RTSP PLAY 发送超时
NET_DVR_RTSP_PLAYSENDERROR	432	RTSP PLAY 发送错误
NET_DVR_RTSP_PLAYRCVTIMEOUT	433	RTSP PLAT 接收超时
NET_DVR_RTSP_PLAYRCVDATALOST	434	RTSP PLAY 接收数据错误
NET_DVR_RTSP_PLAYRCVERROR	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_TEARDOWNSENDTIMEOUT	441	RTSP TEARDOWN 发送超时
NET_DVR_RTSP_TEARDOWNSENDERROR	442	RTSP TEARDOWN 发送错误
NET_DVR_RTSP_TEARDOWNRCVTIMEOUT	443	RTSP TEARDOWN 接收超时
NET_DVR_RTSP_TEARDOWNRCVDATALOST	444	RTSP TEARDOWN 接收数据错误
NET_DVR_RTSP_TEARDOWNRCVERROR	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_NOTSUPPORT	601	不支持
NET_AUDIOINTECOM_ERR_ALLOC_MEMORY	602	内存申请错误
NET_AUDIOINTECOM_ERR_PARAMETER	603	参数错误

NET_AUDIOINTECOM_ERR_CALL_ORDER	604	调用次序错误
NET_AUDIOINTECOM_ERR_FIND_DEVICE	605	未发现设备
NET_AUDIOINTECOM_ERR_OPEN_DEVICE	606	不能打开设备
NET_AUDIOINTECOM_ERR_NO_CONTEXT	607	设备上下文出错
NET_AUDIOINTECOM_ERR_NO_WAVFILE	608	WAV 文件出错
NET_AUDIOINTECOM_ERR_INVALID_TYPE	609	无效的 WAV 参数类型
NET_AUDIOINTECOM_ERR_ENCODE_FAIL	610	编码失败
NET_AUDIOINTECOM_ERR_DECODE_FAIL	611	解码失败
NET_AUDIOINTECOM_ERR_NO_PLAYBACK	612	播放失败
NET_AUDIOINTECOM_ERR_DENOISE_FAIL	613	降噪失败
NET_AUDIOINTECOM_ERR_UNKOWN	619	未知错误
