

设备(IPC)

网络 SDK 编程指南

(for Windows 7/XP/2000/2003/Vista 32bit)

V4.2

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出现，如有任何疑问或争议，请以海康威视最终解释为准。
- 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

目 录	2
1 SDK 简介	8
2 SDK 版本更新	10
3 函数调用顺序.....	11
3.1 SDK 基本调用的主要流程	11
3.2 实时预览模块流程.....	13
3.3 回放和下载模块流程.....	14
3.4 参数配置模块流程.....	15
3.5 远程设备维护模块流程.....	16
3.6 语音对讲转发模块流程.....	17
3.7 报警模块流程.....	18
3.7.1 报警（布防）流程	18
3.7.2 报警（监听）流程	19
3.8 透明通道模块流程.....	20
4 函数调用实例.....	21
4.1 预览模块的示例代码.....	21
4.2 回放和下载模块的示例代码.....	25
4.3 参数配置模块的示例代码.....	33
4.4 远程设备维护模块的示例代码.....	34
4.5 语音对讲转发模块的示例代码.....	37
4.6 报警模块的示例代码.....	38
4.7 透明通道模块的示例代码.....	42
5 函数说明.....	45
5.1 SDK 初始化	45
5.1.1 初始化 SDK NET_DVR_Init	45
5.1.2 释放 SDK 资源 NET_DVR_Cleanup.....	45
5.1.3 设置网络连接超时时间和连接尝试次数 NET_DVR_SetConnectTime.....	45
5.1.4 设置重连功能 NET_DVR_SetReconnect	46
5.1.5 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	46
5.2 异常消息回调.....	46
5.2.1 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.....	46
5.3 SDK 日志和信息	49
5.3.1 获取 SDK 的版本信息 NET_DVR_GetSDKVersion.....	49
5.3.2 获取 SDK 的版本号和 build 信息 NET_DVR_GetSDKBuildVersion	49
5.3.3 获取当前 SDK 的状态信息 NET_DVR_GetSDKState	49
5.3.4 获取当前 SDK 的功能信息 NET_DVR_GetSDKAbility	49
5.3.5 启用写日志文件 NET_DVR_SetLogToFile.....	50
5.4 获取错误信息.....	50

5.4.1	返回最后操作的错误码 NET_DVR_GetLastError	50
5.4.2	返回最后操作的错误码信息 NET_DVR_GetErrorMsg	50
5.5	用户注册	51
5.5.1	用户注册设备 NET_DVR_Login_V30	51
5.5.2	用户注销 NET_DVR_Logout	51
5.6	获取设备能力集	51
5.6.1	获取设备能力集 NET_DVR_GetDeviceAbility	51
5.7	实时预览	53
5.7.1	设置播放显示模式 NET_DVR_SetShowMode	53
5.7.2	主码流动态产生一个关键帧 NET_DVR_MakeKeyFrame	53
5.7.3	子码流动态产生一个关键帧 NET_DVR_MakeKeyFrameSub	54
5.7.4	实时预览 NET_DVR_RealPlay_V30	54
5.7.5	停止预览 NET_DVR_StopRealPlay	55
5.7.6	获取预览时用来解码和显示的播放库句柄 NET_DVR_GetRealPlayerIndex 55	
5.8	预览显示视频参数配置	55
5.8.1	获取预览视频显示参数 NET_DVR_ClientGetVideoEffect	55
5.8.2	获取预览视频显示参数 NET_DVR_GetVideoEffect	56
5.8.3	设置预览视频显示参数 NET_DVR_ClientSetVideoEffect	56
5.8.4	设置预览视频显示参数 NET_DVR_SetVideoEffect	56
5.9	预览画面叠加字符和图像	57
5.9.1	预览画面叠加字符和图像，Linux 下无此接口 NET_DVR_RigisterDrawFun 57	
5.10	预览时解码效果的参数控制	57
5.10.1	设置播放库的帧缓冲区个数 NET_DVR_SetPlayerBufNumber	57
5.10.2	设置解码时丢弃 B 帧的个数 NET_DVR_ThrowBFrame	58
5.11	预览时播放声音控制	58
5.11.1	设置声音播放模式 NET_DVR_SetAudioMode	58
5.11.2	独占声卡模式下开启声音 NET_DVR_OpenSound	58
5.11.3	独占声卡模式下开启声音 NET_DVR_CloseSound	59
5.11.4	共享声卡模式下开启声音 NET_DVR_OpenSoundShare	59
5.11.5	共享声卡模式下关闭声音 NET_DVR_CloseSoundShare	59
5.11.6	调节播放音量 NET_DVR_Volume	59
5.12	实时预览数据捕获	59
5.12.1	注册回调函数，捕获实时码流数据 NET_DVR_SetRealDataCallBack	59
5.12.2	注册回调函数，捕获实时码流数据（标准码流） NET_DVR_SetStandardDataCallBack	60
5.12.3	捕获数据并保存到指定的文件中 NET_DVR_SaveRealData	61
5.12.4	停止数据捕获 NET_DVR_StopSaveRealData	61
5.13	视频抓图	61
5.13.1	设置抓图模式 NET_DVR_SetCapturePictureMode	61
5.13.2	预览时，单帧数据捕获并保存成图片 NET_DVR_CapturePicture	62
5.13.3	单帧数据捕获并保存成 JPEG 图片 NET_DVR_CaptureJPEGPicture	62
5.13.4	单帧数据捕获并保存成 JPEG 存放在指定的内存空间中	

NET_DVR_CaptureJPEGPicture_NEW	62
5.14 录像文件回放、下载、锁定及备份	63
录像文件的查找	63
5.14.1 根据文件类型、时间查找设备录像文件 NET_DVR_FindFile_V40	63
5.14.2 逐个获取查找到的文件信息 NET_DVR_FindNextFile_V30	63
5.14.3 关闭文件查找，释放资源 NET_DVR_FindClose_V30	64
回放录像文件	64
5.14.4 按文件名回放录像文件 NET_DVR_PlayBackByName	64
5.14.5 按时间回放录像文件 NET_DVR_PlayBackByTime_V40	65
5.14.6 控制录像回放的状态 NET_DVR_PlayBackControl_V40	66
5.14.7 停止回放录像文件 NET_DVR_StopPlayBack	68
回放录像文件时的数据捕获	68
5.14.8 捕获回放的录像数据，并保存成文件 NET_DVR_PlayBackSaveData	68
5.14.9 停止保存录像数据 NET_DVR_StopPlayBackSave	68
5.14.10 注册回调函数，捕获录像数据 NET_DVR_SetPlayDataCallBack	69
回放的其他操作	69
5.14.11 获取录像回放时显示的 OSD 时间 NET_DVR_GetPlayBackOsdTime	69
5.14.12 录像回放时抓图，并保存在文件中 NET_DVR_PlayBackCaptureFile	70
5.14.13 刷新显示回放窗口 NET_DVR_RefreshPlay	70
5.14.14 获取回放时用来解码显示的播放库句柄 NET_DVR_GetPlayBackPlayerIndex	70
下载录像文件	71
5.14.15 按文件名下载录像文件 NET_DVR_GetFileByName	71
5.14.16 按时间下载录像文件 NET_DVR_GetFileByTime	71
5.14.17 控制录像下载的状态 NET_DVR_PlayBackControl	72
5.14.18 停止下载录像文件 NET_DVR_StopGetFile	73
5.14.19 获取当前下载录像文件的进度 NET_DVR_GetDownloadPos	73
录像文件锁定和解锁	73
5.14.20 按文件名锁定录像文件 NET_DVR_LockFileByName	73
5.14.21 按文件名解锁录像文件 NET_DVR_UnlockFileByName	73
5.15 手动录像	74
5.15.1 远程手动启动设备录像 NET_DVR_StartDVRRecord	74
5.15.2 远程手动停止设备录像 NET_DVR_StopDVRRecord	74
5.16 布防、撤防	75
设置报警等信息上传的回调函数	75
5.16.1 注册回调函数，接收设备报警消息 NET_DVR_SetDVRMessageCallBack_V30 75	
布防撤防 75	
5.16.2 建立报警上传通道，获取报警等信息 NET_DVR_SetupAlarmChan_V30	75
5.16.3 撤销报警上传通道 NET_DVR_CloseAlarmChan_V30	76
5.17 监听报警	76
5.17.1 启动监听，接收设备主动上传的报警等信息 NET_DVR_StartListen_V30	76
5.17.2 停止监听（支持多线程）NET_DVR_StopListen_V30	77
5.18 云台控制	77

云台控制操作.....	77
5.18.1 云台控制操作（需先启动图像预览）NET_DVR_PTZControl.....	77
5.18.2 云台控制操作（不用启动图像预览）NET_DVR_PTZControl_Other	78
5.18.3 带速度的云台控制操作（需先启动图像预览）NET_DVR_PTZControlWithSpeed.....	79
5.18.4 带速度的云台控制操作（不用启动图像预览）NET_DVR_PTZControlWithSpeed_Other	80
云台预置点操作.....	82
5.18.5 云台预置点操作，需先启动预览 NET_DVR_PTZPreset	82
5.18.6 云台预置点操作 NET_DVR_PTZPreset_Other	82
云台巡航操作.....	83
5.18.7 云台巡航操作，需先启动预览 NET_DVR_PTZPCruise	83
5.18.8 云台巡航操作 NET_DVR_PTZPCruise_Other.....	83
云台轨迹操作.....	84
5.18.9 云台轨迹操作，需先启动预览 NET_DVR_PTZTrack	84
5.18.10 云台轨迹操作 NET_DVR_PTZTrack_Other.....	84
透明云台控制.....	85
5.18.11 透明云台操作，需先启动预览 NET_DVR_TransPTZ	85
5.18.12 透明云台操作 NET_DVR_TransPTZ_Other	85
云台区域缩放控制.....	86
5.18.13 云台图象区域选择放大或缩小 NET_DVR_PTZSelZoomIn	86
5.18.14 云台图像区域选择放大或缩小 NET_DVR_PTZSelZoomIn_Ex.....	86
云台定位 86	
5.18.15 获取设备的配置信息 NET_DVR_GetDVRConfig	86
5.18.16 设置设备的配置信息 NET_DVR_SetDVRConfig	87
获取巡航路径.....	87
5.18.17 获取 IP 快球云台巡航路径 NET_DVR_GetPTZCruise	87
5.19 IPC 远程控制	88
5.19.1 控制一键聚焦 NET_DVR_FocusOnePush	88
5.19.2 恢复镜头电机默认位置 NET_DVR_ResetLens	88
5.19.3 遥控器控制 NET_DVR_RemoteControl.....	88
5.20 语音对讲、转发及广播.....	89
语音对讲 89	
5.20.1 启动语音对讲 NET_DVR_StartVoiceCom_V30	89
5.20.2 设置语音对讲客户端的音量 NET_DVR_SetVoiceComClientVolume	90
5.20.3 停止语音对讲或者语音转发 NET_DVR_StopVoiceCom	90
语音转发 91	
5.20.4 启动语音转发，获取编码后的音频数据 NET_DVR_StartVoiceCom_MR_V30 91	
5.20.5 转发语音数据 NET_DVR_VoiceComSendData	92
5.20.6 停止语音对讲或语音转发 NET_DVR_StopVoiceCom.....	92
语音广播 93	
5.20.7 启动语音广播的 PC 端声音捕获 NET_DVR_ClientAudioStart_V30	93
5.20.8 添加设备的某个语音通道到可以接收 PC 端声音的广播组	

NET_DVR_AddDVR_V30.....	93
5.20.9 从可接收 PC 机声音的广播组里删除该设备的语音通道	
NET_DVR_DelDVR_V30.....	94
5.20.10 停止语音广播的 PC 端声音捕获 NET_DVR_ClientAudioStop	94
音频编解码.....	94
OggVorbis 音频编解码	94
5.20.11 初始化音频编码 NET_DVR_InitG722Encoder	94
5.20.12 OggVorbis 音频编码 NET_DVR_EncodeG722Frame	94
5.20.13 释放音频编码资源 NET_DVR_ReleaseG722Encoder	95
5.20.14 初始化音频解码 NET_DVR_InitG722Decoder	95
5.20.15 OggVorbis 音频解码 NET_DVR_DecodeG722Frame.....	95
5.20.16 释放音频解码资源 NET_DVR_ReleaseG722Decoder.....	96
G711 音频编解码.....	96
5.20.17 G711 音频编码 NET_DVR_EncodeG711Frame.....	96
5.20.18 G711 音频解码 NET_DVR_DecodeG711Frame	96
G726 音频编解码	97
5.20.19 初始化音频编码 NET_DVR_InitG726Encoder	97
5.20.20 G722 音频编码 NET_DVR_EncodeG726Frame	97
5.20.21 释放音频编码资源 NET_DVR_ReleaseG726Encoder	97
5.20.22 初始化音频解码 NET_DVR_InitG726Decoder.....	98
5.20.23 G722 音频解码 NET_DVR_DecodeG726Frame.....	98
5.20.24 释放音频解码资源 NET_DVR_ReleaseG726Decoder.....	98
5.21 透明通道.....	99
5.21.1 建立透明通道 NET_DVR_SerialStart	99
5.21.2 通过透明通道向设备串口发送数据 NET_DVR_SerialSend	99
5.21.3 断开透明通道 NET_DVR_SerialStop	99
5.22 向串口发送数据.....	100
5.22.1 直接向串口发送数据，不需要建立透明通道 NET_DVR_SendToSerialPort	100
5.22.2 直接向 232 串口发送数据，不需要建立透明通道 NET_DVR_SendTo232Port	100
5.23 硬盘管理.....	101
5.23.1 远程格式化设备硬盘 NET_DVR_FormatDisk.....	101
5.23.2 获取格式化硬盘的进度 NET_DVR_GetFormatProgress	101
5.23.3 关闭格式化硬盘句柄，释放资源 NET_DVR_CloseFormatHandle	101
5.24 设备维护管理.....	102
获取设备工作状态.....	102
5.24.1 获取设备的工作状态 NET_DVR_GetDVRWorkState_V30	102
远程升级 102	
5.24.2 设置远程升级时网络环境 NET_DVR_SetNetworkEnvironment	102
5.24.3 远程升级 NET_DVR_Upgrade	103
5.24.4 获取远程升级的进度 NET_DVR_GetUpgradeProgress	103
5.24.5 获取远程升级的状态 NET_DVR_GetUpgradeState.....	103
5.24.6 获取远程升级的阶段信息 NET_DVR_GetUpgradeStep.....	103

5.24.7	关闭远程升级句柄，释放资源 NET_DVR_CloseUpgradeHandle	104
	日志查找	104
5.24.8	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	104
5.24.9	逐条获取查找到的日志信息 NET_DVR_FindNextLog_V30.....	110
5.24.10	释放查找日志的资源 NET_DVR_FindLogClose_V30.....	110
	远程备份	111
5.24.11	获取设备磁盘列表 NET_DVR_GetDiskList	111
5.24.12	备份统一接口 NET_DVR_Backup	111
5.24.13	获取备份的进度 NET_DVR_GetBackupProgress	111
5.24.14	停止备份 NET_DVR_StopBackup	112
	恢复设备默认参数	112
5.24.15	恢复设备默认参数 NET_DVR_RestoreConfig.....	112
	导入/导出配置文件	113
5.24.16	导出配置文件 NET_DVR_GetConfigFile_V30	113
5.24.17	导出配置文件 NET_DVR_GetConfigFile	113
5.24.18	导入配置文件 NET_DVR_SetConfigFile_EX	113
5.24.19	导入配置文件 NET_DVR_SetConfigFile	113
5.25	关机和重启	114
5.25.1	重启设备 NET_DVR_RebootDVR.....	114
5.25.2	关闭设备 NET_DVR_ShutDownDVR	114
5.26	远程参数配置	114
	通用参数配置	114
5.26.1	获取设备的配置信息 NET_DVR_GetDVRConfig	114
5.26.2	设置设备的配置信息 NET_DVR_SetDVRConfig	116
	报警输出配置	118
5.26.3	获取设备报警输出 NET_DVR_GetAlarmOut_V30	118
5.26.4	设置设备报警输出 NET_DVR_SetAlarmOut	118
	RTSP 协议参数配置	118
5.26.5	获取 RTSP 协议参数 NET_DVR_GetRtspConfig	118
5.26.6	设置 RTSP 协议参数 NET_DVR_SetRtspConfig.....	119
5.27	邮件测试	119
5.27.1	测试按已配置的 EMAIL 参数能否收发成功 NET_DVR_StartEmailTest	119
5.27.2	获取邮件测试的进度 NET_DVR_GetEmailTestProgress	119
5.27.3	停止邮件测试 NET_DVR_StopEmailTest.....	120
5.28	热成像仪	120
5.28.1	热成像仪手动快门补偿 NET_DVR_ShutterCompensation	120
5.28.2	热成像仪坏点校正 NET_DVR_CorrectDeadPixel	120
6	错误代码及说明	121
6.1	网络通讯库错误码	121
6.2	RTSP 通讯库错误码	124
6.3	软解码库错误码	125

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、大屏、报警主机等网络产品服务的配套模块，用于远程访问和控制设备软件的二次开发。

本文档仅介绍 IPC(网络摄像机)和 IPD（网络球机）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》。

设备网络 SDK 主要功能：

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

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

Windows 下设备网络 SDK：

网络通讯库	HCNetSDK.h	头文件
	HCNetSDK.lib	LIB 库文件
	HCNetSDK.dll	DLL 库文件
Qos 库	QosControl.dll	DLL 库文件
RTSP 通讯库	StreamTransClient.dll	DLL 库文件
软解码库	PlayM4.h	头文件
	PlayCtrl.lib	LIB 库文件
	PlayCtrl.dll	DLL 库文件
转封装库	SystemTransform.dll	DLL 库文件
硬解码库	DataType.h 和 DecodeCardSdk.h	头文件
	DsSdk.lib	LIB 库文件
	DsSdk.dll	DLL 库文件

Linux 下设备网络 SDK 库：

网络通讯库	hcnet sdk.h	头文件
	libhcnet sdk.so	SO 库文件
Qos 库	libQosControl.so	SO 库文件
RTSP 通讯库	libStreamTransClient.so	SO 库文件
软解码库	playsdkpu.h	头文件
	libm4play.so	SO 库文件
转封装库	libSystemTransform.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，除网络通讯库为必须项外，用户可以根据需要选择其中的一部分组件，以下将对各个组件在 SDK 中的作用和使用条件分别说明。

- 网络通讯库是设备网络 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。
- Qos 库是流控库。推模式下支持，用于无线实时流的码流传输质量优化。
- RTSP 通讯库是支持 RTSP 传输协议的网络库。当需要对支持 RTSP 协议的产品进行取流等操作时就必须加载该项组件。
- 软解码库主要用于对实时码流数据进行解码显示（实现预览功能）和对录像文件进行回放解码等。用户如果需要在 SDK 内部进行对实时流和录像码流播放显示时（即 NET_DVR_RealPlay_V30 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》。
- 转封装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_V30 接口中的回调函数捕获数据或者调用 NET_DVR_SetRealDataCallBack 接口捕获数据时）必须加载该组件。另一种功能是将标准码流转换成其他格式的封装，如 3GPP、PS 等。例如，当用户需要对支持 RTSP 协议的产品实时捕获指定封装格式的码流数据（对应的 SDK 接口为 NET_DVR_SaveRealData）时必须加载该项组件。
- 硬解码库需在配备硬解码卡（MD 卡）的前提下使用，通过解码卡的解码与输出功能实现实时流的解码显示及向监视器上矩阵输出的功能。对于网络摄像机或者球机，不需要使用该库文件。

2 SDK 版本更新

Version 4.1.0 (2012-5-9)

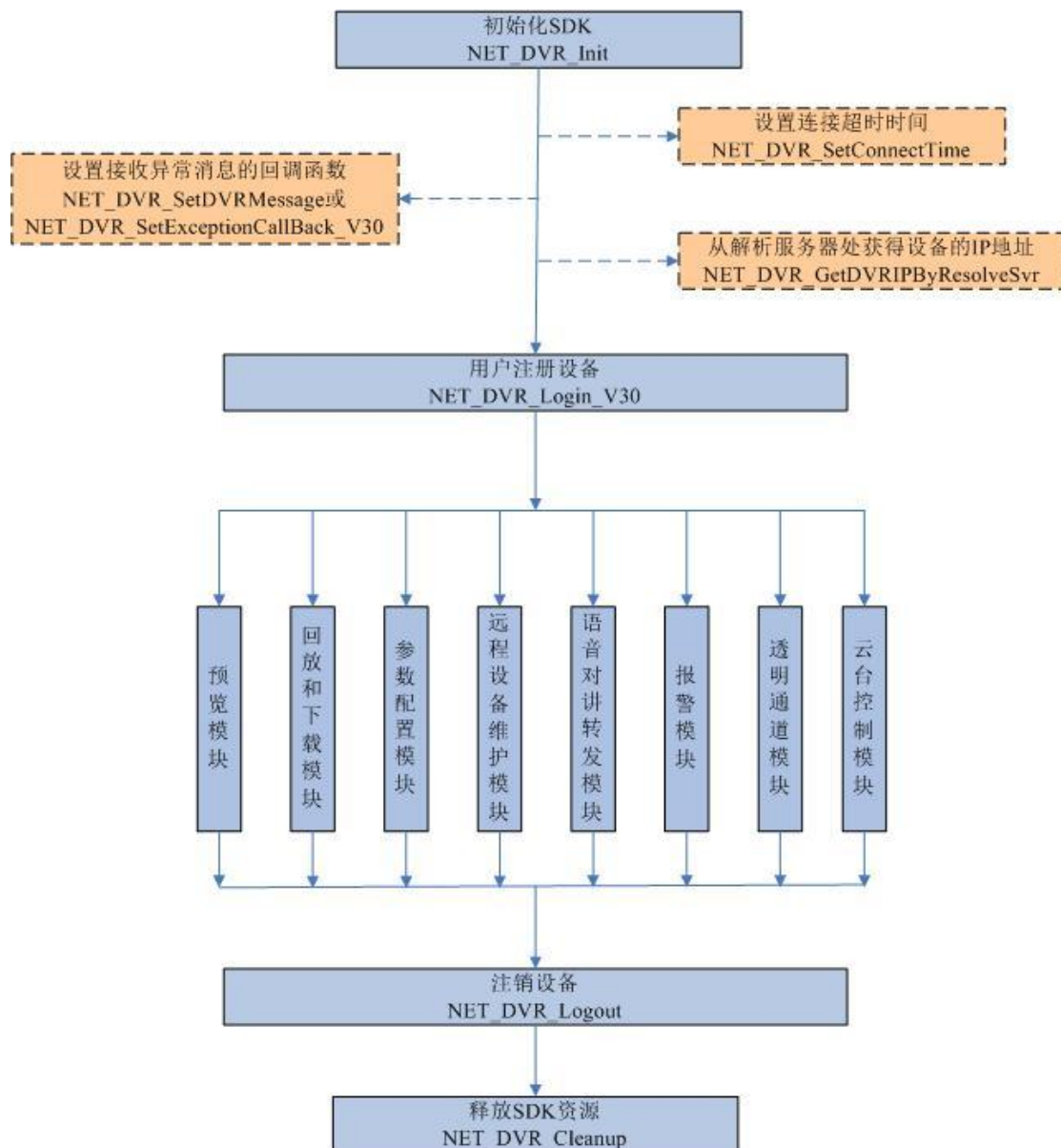
- 红外热成像摄像机接口：
NET_DVR_ShutterCompensation、NET_DVR_CorrectDeadPixel
- IPC v4.0 新增接口：
NET_DVR_FocusOnePush、NET_DVR_ResetLens、NET_DVR_RemoteControl
- 新增配置功能：
NET_DVR_AUDIO_INPUT_PARAM、NET_DVR_CAMERA_DEHAZE_CFG、NET_IPC_AUX_ALARMCFG
- 新增报警上传：
NET_IPC_AUXALARM_RESULT
- 新增能力集：
DEVICE_ALARM_ABILITY

Version 4.1.0 (2012-4-5)

- 新增 G726 音频编解码接口：
NET_DVR_InitG726Encoder、NET_DVR_EncodeG726Frame、NET_DVR_ReleaseG726Encoder、
NET_DVR_InitG726Decoder、NET_DVR_DecodeG726Frame、NET_DVR_ReleaseG726Decoder

3 函数调用顺序

3.1 SDK 基本调用的主要流程



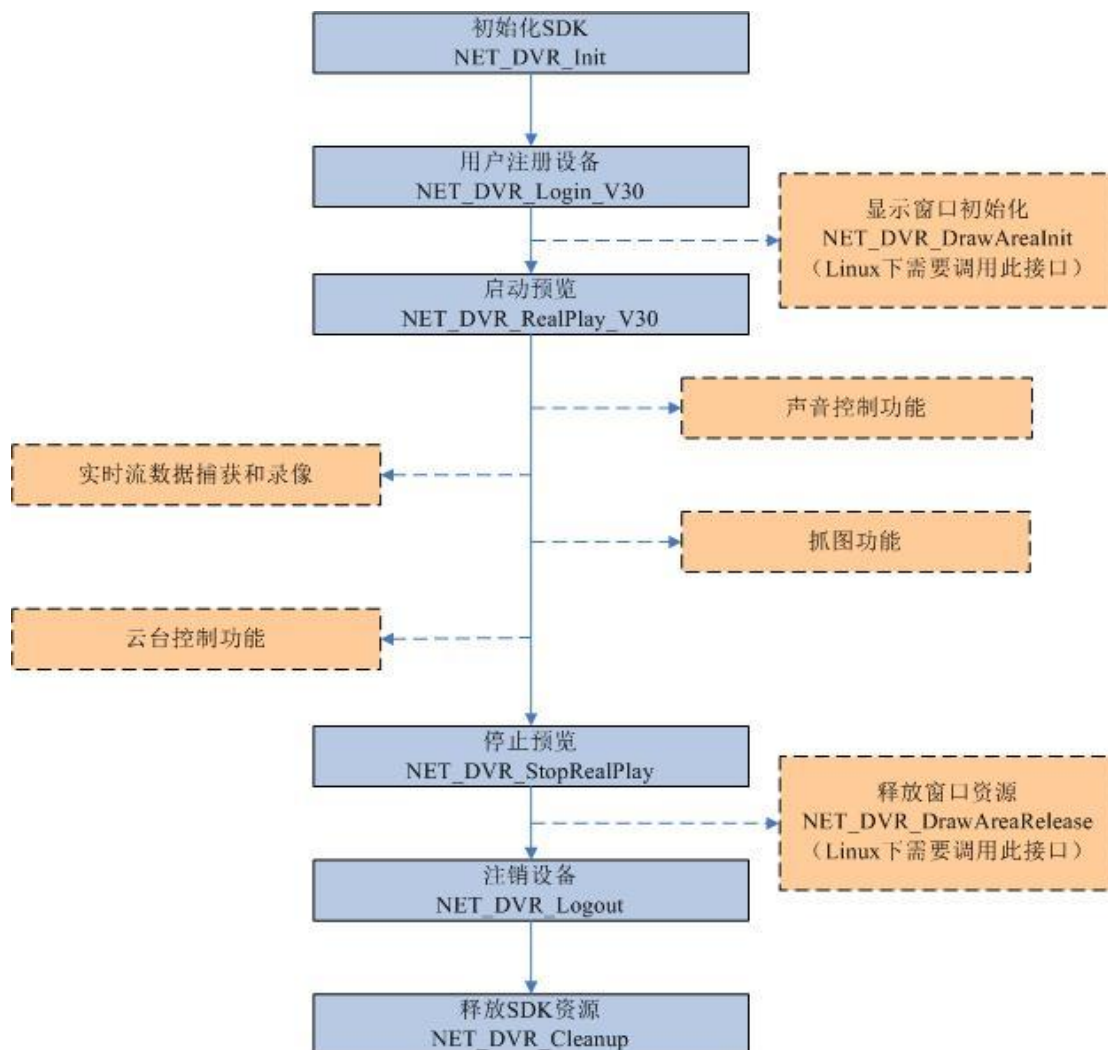
其中虚线框的流程是可选部分，不会影响其他流程和模块的功能使用。按实现功能的不同可以分成十个模块，实现每个模块的功能时初始化 SDK、用户注册设备、注销设备和释放 SDK 资源这 4 个流程是必不可少的，解码器功能模块和行为分析功能模块是针对解码器和智能设备的，在该文档里我们不做描述。

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

情况下，将采用 SDK 中的默认值。

- 设置接收异常消息的回调函数（[NET_DVR_SetDVRMessage](#) 或 [NET_DVR_SetExceptionCallBack_V30](#)）：由于 SDK 中大部分模块的功能都是由异步模式实现，所以我们提供此接口用于接收预览、报警、回放、透明通道和语音对讲等模块发生异常信息。用户可以在初始化 SDK 后就设置该回调函数，在应用层对各个模块异常消息的接收和处理。
- 从解析服务器获得设备的 IP 地址（[NET_DVR_GetDVRIPByResolveSvr_Ex](#)）：该接口提供一种在仅知道设备名称和序列号的情况下，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，我们可以通过此接口输入解析服务器的地址、设备的名称和序列号等信息查询该设备的 IP 地址。IPServer 是我公司提供的一款域名解析服务器软件。
- 用户注册设备（[NET_DVR_Login_V30](#)）：实现用户的注册功能，注册成功后，返回的用户 ID 作为其他功能操作的唯一标识，SDK 允许最大注册用户数为 512 个。网络摄像机和网络球机允许有 16 个注册用户名，而且同时最多允许 128 个用户注册。
- 预览模块：从前端设备取实时码流，解码显示以及播放控制等功能，同时支持软解码和解码卡解码。具体流程详见[预览模块流程](#)。
- 回放和下载模块：可以通过按时间和按文件名的方式远程回放或者下载的录像文件，后续可以进行解码或者存储。同时还支持断点续传功能。具体流程详见[回放和下载模块流程](#)。
- 参数配置模块：设置和获取前端设备的参数，主要包括设备参数、网络参数、通道压缩参数、串口参数、报警参数、异常参数、交易信息和用户配置等参数信息。具体流程详见[参数配置模块流程](#)。
- 远程设备维护模块：实现关闭设备、重启设备、恢复默认值、远程硬盘格式化、远程升级和配置文件导入/导出等维护工作。具体流程详见[远程设备维护模块流程](#)。
- 语音对讲转发模块：实现和前端设备的语音数据对讲和语音数据获取，音频编码格式可以指定。具体流程详见[语音对讲转发模块流程](#)。
- 报警模块：处理设备上传的各种报警信号。报警分为“布防”和“监听”两种方式，在采用监听方式并且不需要获取用户 ID 的情况下，报警模块可以无需进行“用户注册”操作步骤。具体流程详见[报警模块流程](#)。
- 透明通道模块：透明通道是将 IP 数据报文解析后直接发送到串行口的一种技术。实际上起到了延伸串行设备控制距离的作用。可利用 IP 网络控制多种串行设备，如控制解码器、矩阵、报警主机、门禁、仪器仪表等串行设备，对用户来说，只看到点对点传输，无须关心网络传输过程，所以称为串口透明通道。SDK 提供 485 和 232 串口作为透明通道功能，其中要将 232 串口作为透明通道使用，首先必须在 232 串口的配置信息（NET_DVR_RS232CFG）中将工作模式选为透明通道，这样 232 串口才可作为透明通道使用。具体流程详见[透明通道模块流程](#)。
- 云台控制模块：实现对云台的基本操作、预置点、巡航、轨迹和透明云台的控制。SDK 将云台控制分为两种模式：一种是通过图像预览返回的句柄进行控制；另一种是无预览限制，通过用户注册 ID 号进行云台控制。

3.2 实时预览模块流程



图中虚线框部分的模块是与预览模块相关，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。

实时流解码方式：

- 方式一：在预览接口 [NET_DVR_RealPlay_V30](#) 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_V30](#) 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_V30](#) 接口中的回调函数或调用 [NET_DVR_SetRealDataCallBack](#)、[NET_DVR_SetStandardDataCallBack](#) 接口），获取码流数据进行后续解码播放处理。

相关功能：

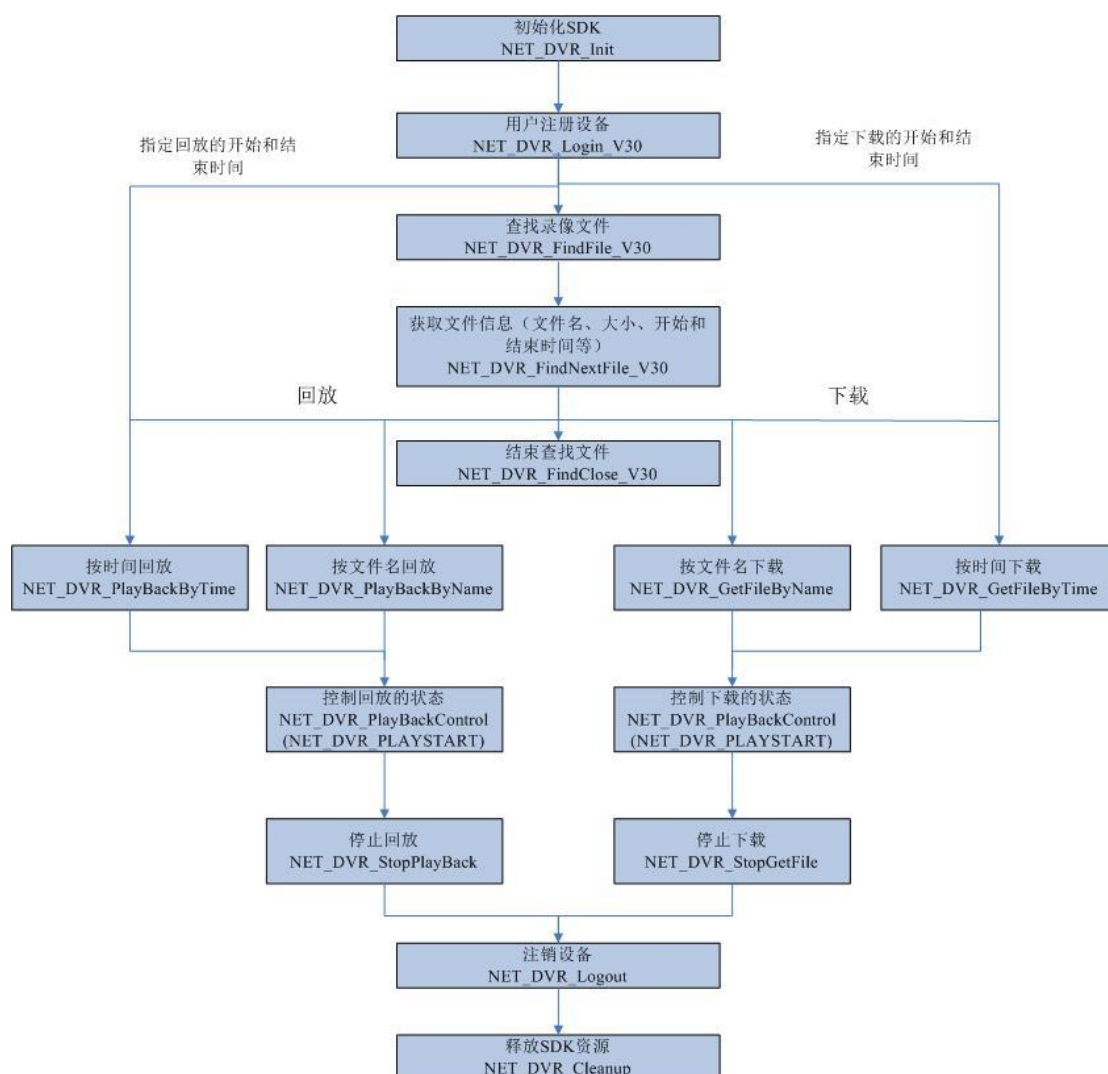
- 声音控制功能主要实现独占、共享声音的打开和关闭；音量的控制。相关接口有：[NET_DVR_OpenSound](#)、[NET_DVR_CloseSound](#)、[NET_DVR_OpenSoundShare](#)、[NET_DVR_CloseSoundShare](#)、[NET_DVR_Volume](#) 等。
- 实时流数据捕获和录像模块主要实现数据回调和本地录像的功能，可以供用户后续处

理。相关接口有：[NET_DVR_SetRealDataCallBack](#)、[NET_DVR_SetStandardDataCallBack](#)、[NET_DVR_SaveRealData](#) 等。

- 抓图功能主要实现对当前解码图像的捕获，保存格式为 BMP。相关接口有：[NET_DVR_CapturePicture](#)。[NET_DVR_CaptureJPEGPicture](#) 支持登录后直接从设备抓取 JPEG 图片。
- 云台控制模块主要是在开启预览的前提下实现对云台控制的操作功能，包括云台预置点、巡航、轨迹和透明云台等。相关接口有：[NET_DVR_PTZControl](#)、[NET_DVR_PTZPreset](#)、[NET_DVR_PTZCruise](#)、[NET_DVR_PTZTrack](#)、[NET_DVR_TransPTZ](#)。

[调用实例代码](#)

3.3 回放和下载模块流程

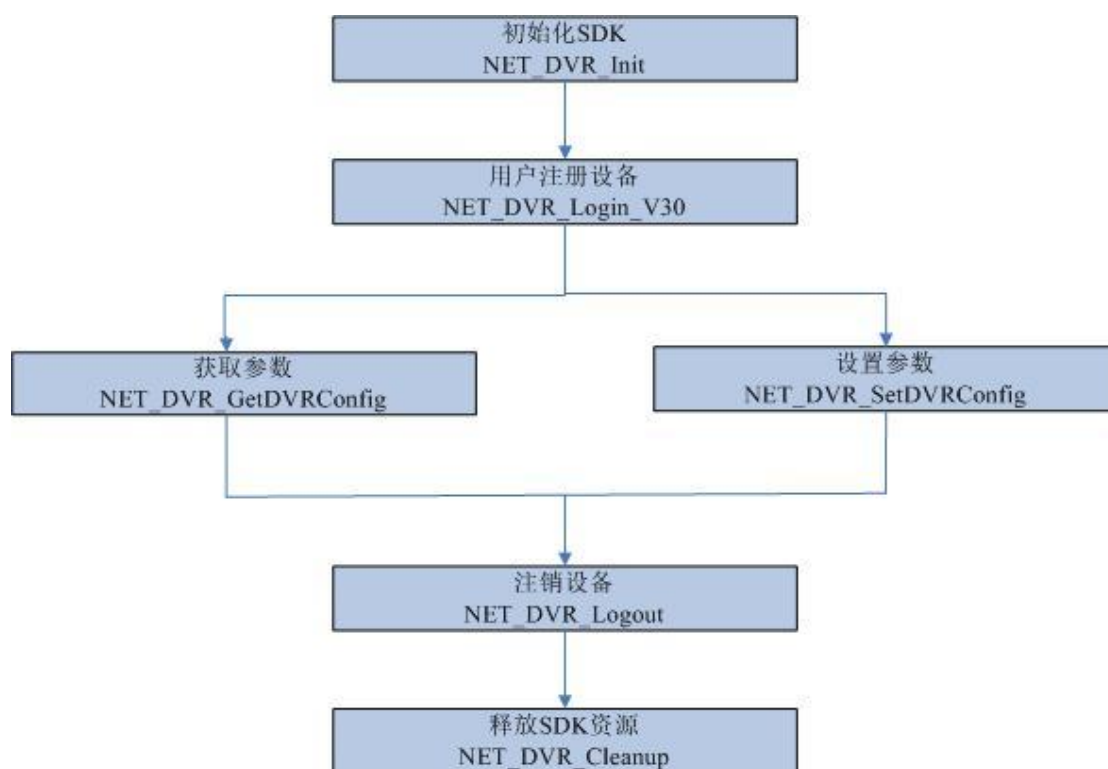


- 按文件回放或下载需要通过查找录像文件功能先获取文件信息（相关接口 [NET_DVR_FindFile_V40](#)、[NET_DVR_FindNextFile_V30](#)），然后根据获取到的文件名开始回放或下载（相关接口 [NET_DVR_PlayBackByName](#)、[NET_DVR_GetFileByName](#)），特别提醒在调用了回放或下载的接口后，还必须调用控制接口（[NET_DVR_PlayBackControl_V40](#)）的开始播放命令（[NET_DVR_PLAYSTART](#)）。

- 按时间回放或下载文件时，用户可以无需调用查找录像文件的相关接口，只要在接口中指定开始和结束时间，调用回放或下载接口（相关接口 [NET_DVR_PlayBackByTime](#)、[NET_DVR_GetFileByTime](#)）后，还必须调用控制接口（[NET_DVR_PlayBackControl_V40](#)）的开始播放命令（NET_DVR_PLAYSTART）。此时，将按照指定时间范围内最近的有录像的时间段开始回放或下载。用户也可以通过调用查找录像文件的相关接口，获取文件的开始和结束时间后，按这个时间范围指定回放或下载接口中的时间参数，最后还必须调用控制接口（[NET_DVR_PlayBackControl_V40](#)）的开始播放命令（NET_DVR_PLAYSTART）。

[调用实例代码](#)

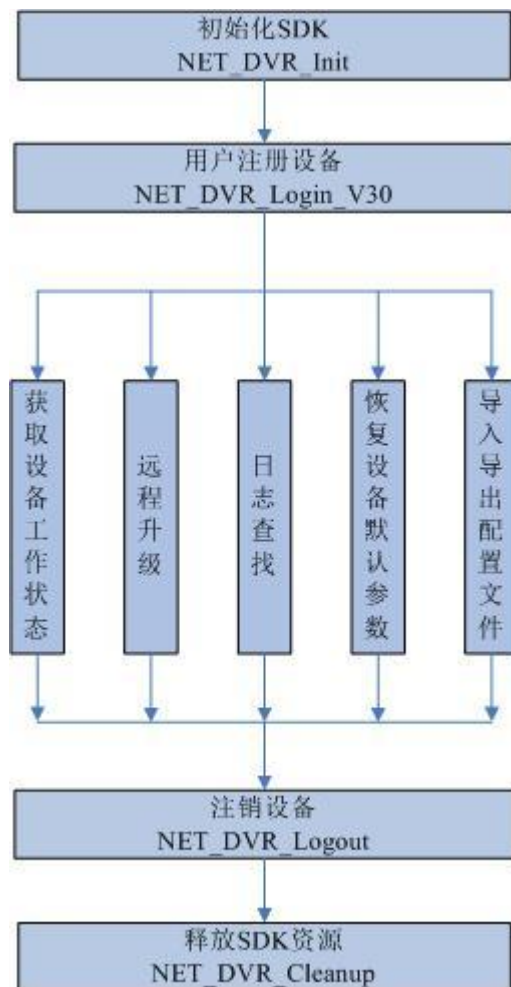
3.4 参数配置模块流程



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

[调用实例代码](#)

3.5 远程设备维护模块流程

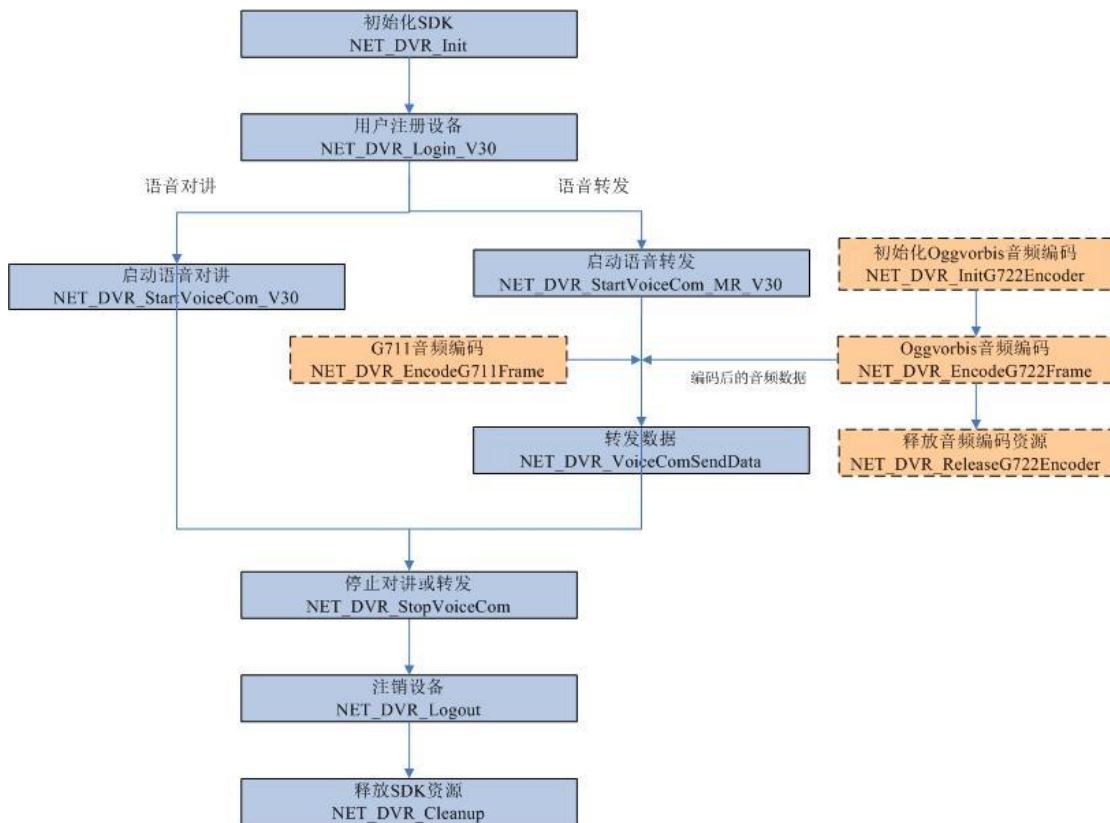


远程设备维护模块包括获取设备工作状态、远程升级、日志查找、恢复设备默认参数和导入、导出配置文件等功能。

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

[调用实例代码](#)

3.6 语音对讲转发模块流程



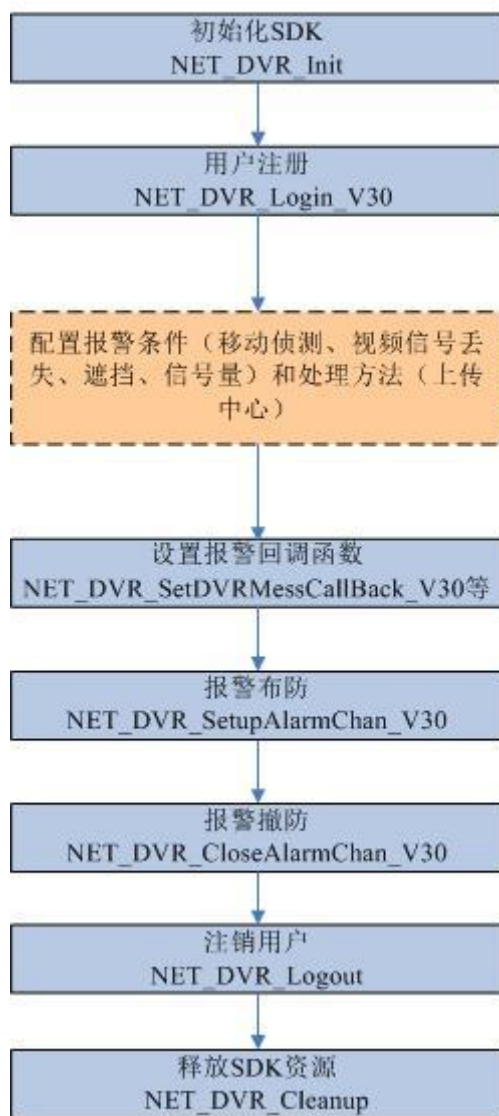
- 语音对讲功能实现 PC 机与设备间音频的发送和接收。在成功注册设备后调用 [NET_DVR_StartVoiceCom_V30](#) 接口完成，同时在该接口中用户可以通过设置回调函数获取当前设备发送或者 PC 机采集的数据（按需要选择回调编码后或者 PCM 数据）。
- 语音转发功能实现将待发送的音频数据（编码后）转发给设备。首先调用 [NET_DVR_StartVoiceCom_MR_V30](#) 接口启动与某台设备的语音转发，再调用 [NET_DVR_VoiceComSendData](#) 接口发送经过编码的音频数据，但是对每次发送的数据大小有限定值，详细说明可见接口。首先调用 [NET_DVR_StartVoiceCom_MR_V30](#) 接口启动与设备的语音转发（此时已建立与设备之间的连接，等待发送数据）。第二，准备好待发送的数据（需要经过编码），相对于上图中虚线框部分，如果数据已按本公司的压缩格式处理这部分就可以省略。数据源可以从 PC 声卡中采集或是从文件中读取，但是需要经过本公司提供的压缩算法进行压缩处理，SDK 提供一套编码接口，方法如下：1）初始化音频编码，调用接口 [NET_DVR_InitG722Encoder](#)；2）OggVorbis 音频编码，调用接口 [NET_DVR_EncodeG722Frame](#)，该接口的参数有一定的要求，详见接口说明；3）当结束所有的编码过程需要调用 [NET_DVR_ReleaseG722Encoder](#) 接口释放编码音频资源。经过第二部的编码操作，我们可以每次得到固定大小的且经过编码后的数据，调用 [NET_DVR_VoiceComSendData](#) 接口发送这些数据给设备。等所有的转发操作完成后，调用 [NET_DVR_StopVoiceCom](#) 接口结束与设备的语音转发连接。
- Linux 下目前只支持语音转发功能，暂不支持语音对讲。

[调用实例代码](#)

3.7 报警模块流程

3.7.1 报警（布防）流程

报警（布防）流程图



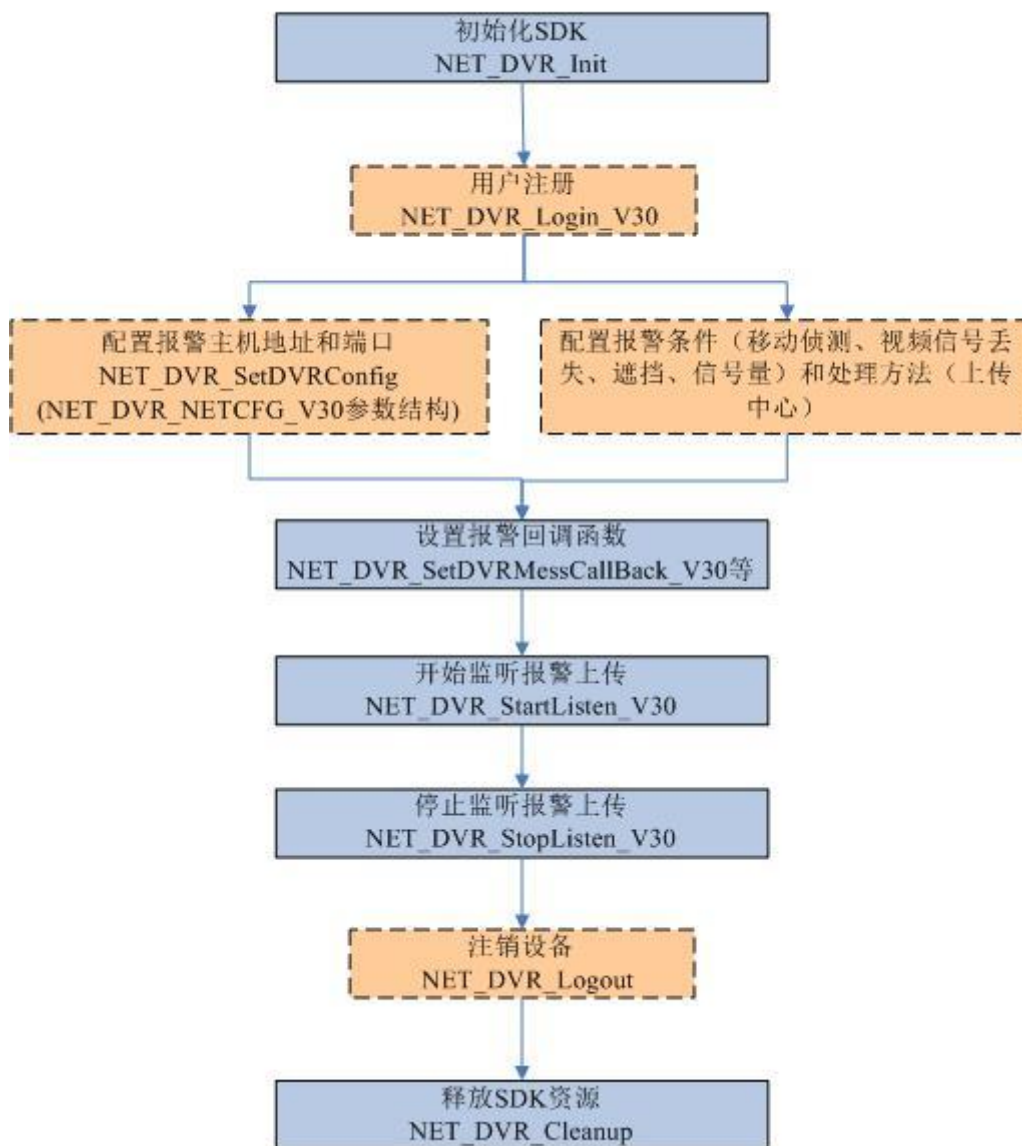
- “布防”报警方式：是指 SDK 主动连接设备，并发起报警上传命令，设备发生报警立即发送给 SDK。
- 由“报警（布防）的流程图”中看出，“布防”方式需要先进行用户注册（[NET_DVR_Login_V30](#)）。虚线框部分是实现报警信息上传的必要条件，主要完成相关的报警条件和处理方法的配置，参数配置的接口为 [NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)。支持的报警类型有移动侦测、视频信号丢失、遮挡和信号量报警，其中前三种报警类型对应的报警条件和处理方法的配置结构体是 [NET_DVR_PICCFG_V30](#)，而信号量报警的配置结构体是 [NET_DVR_ALARMINGCFG_V30](#)。这些参数如果已经配置完成，那么虚线框部分可以省略。接下来就是设置报警回调函数（[NET_DVR_SetDVRMessageCallBack_V30](#) 等），调用成功后还需要设置布防

([NET_DVR_SetupAlarmChan_V30](#))。整个报警上传过程结束后还需要调用撤防接口等操作。

[调用实例代码](#)

3.7.2 报警（监听）流程

报警（监听）流程图



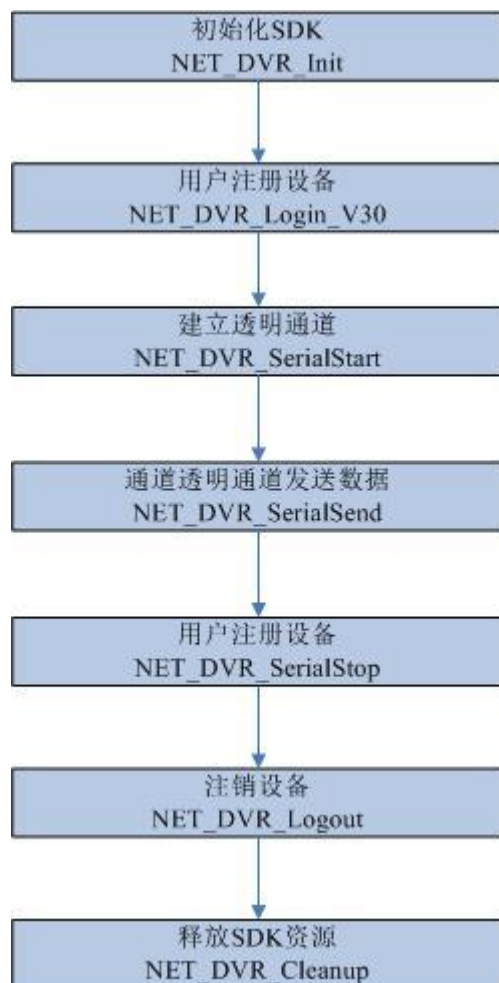
- “监听”报警方式：是指 SDK 不主动发起连接设备，只是在设定的端口上监听接收设备主动上传的报警信息。
- 这个过程需要远程配置设备的报警主机地址（即 PC 机地址）和报警主机端口（即 PC 的监听端口），报警主机就在该端口上监听接收设备主动上传的报警信息。如果报警主机地址和报警主机端口已配置完成，那么“报警（监听）的流程图”中虚线框“用户注册”和“配置报警主机地址和端口”部分就可以省略，但事先没有配置，就必须调用参数配置接口（[NET_DVR_GetDVRConfig](#) 和 [NET_DVR_SetDVRConfig](#)）对设备的网络参数（NET_DVR_NETCFG_V30）进行配置。而虚线框“配置报警条件和处理方法”部分与“布防”

中的一致。对以上需要配置的参数都设置完后，调用 [NET_DVR_StartListen_V30](#) 函数，开启 SDK 的监听端口，准备接收设备上传的报警信息。

- 该方式适用于多个设备向一台客户端上传报警，而且不需要设备登录即可完成，设备重启后不影响报警上传；缺点是设备只支持一个报警主机地址和端口号的配置。

[调用实例代码](#)

3.8 透明通道模块流程



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

[调用实例代码](#)

4 函数调用实例

4.1 预览模块的示例代码

[相关模块流程图](#)

方式一 SDK 直接解码显示

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

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);

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

```

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

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

//-----
//启动预览并设置回调数据流
LONG IRealPlayHandle;
HWND hWnd = GetConsoleWindow();    //获取窗口句柄
NET_DVR_CLIENTINFO ClientInfo = {0};
ClientInfo.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
ClientInfo.IChannel    = 1;        //预览通道号
ClientInfo.ILinkMode    = 0;        /*最高位(31)为 0 表示主码流，为 1 表示子码流。0~30 位表示连接
方式：0—TCP 方式；1—UDP 方式；2—多播方式；3—RTP 方式；4- RTP over RTSP，5- RTSP over HTTP */
ClientInfo.sMultiCastIP = NULL;    //多播地址，需要多播预览时配置

BOOL bPreviewBlock = false;        //请求码流过程是否阻塞，0：否，1：是
IRealPlayHandle = NET_DVR_RealPlay_V30(IUserID, &ClientInfo, NULL, NULL, 0);
if (IRealPlayHandle < 0)
{
    printf("NET_DVR_RealPlay_V30 error\n");
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//-----
//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;

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;
                }
            }
        }
    }
```



```

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

    //-----
    // 注册设备
    LONG lUserID;
    NET_DVR_DEVICEINFO_V30 struDeviceInfo;
    lUserID = NET_DVR_Login_V30("172.0.0.100", 8000, "admin", "12345", &struDeviceInfo);
    if (lUserID < 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_CLIENTINFO ClientInfo = {0};
ClientInfo.hPlayWnd = NULL;           //需要 SDK 解码时句柄设为有效值，仅取流不解码时可设为空
ClientInfo.IChannel = 1;              //预览通道号
ClientInfo.ILinkMode = 0;             /*最高位(31)为 0 表示主码流，为 1 表示子码流。0~30 位表示连接
方式：0—TCP 方式；1—UDP 方式；2—多播方式；3—RTP 方式；4- RTP over RTSP，5- RTSP over HTTP*/
ClientInfo.sMultiCastIP = NULL;      //多播地址，需要多播预览时配置

BOOL bPreviewBlock = false;          //请求码流过程是否阻塞，0：否，1：是
IRealPlayHandle = NET_DVR_RealPlay_V30(IUserID, &ClientInfo, g_RealDataCallBack_V30, NULL, 0);
if (IRealPlayHandle < 0)
{
    printf("NET_DVR_RealPlay_V30 error\n");
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

//-----
//关闭预览
NET_DVR_StopRealPlay(IRealPlayHandle);
//注销用户
NET_DVR_Logout_V30(IUserID);
NET_DVR_Cleanup();

return;
}

```

4.2 回放和下载模块的示例代码

[相关模块流程图](#)

示例一：查找录像文件并下载

```
#include <stdio.h>
```

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

int saveRecordFile(int userId,char * srcfile,char * destfile)
{
    int bRes = 1;
    int hPlayback = 0;
    if( (hPlayback = NET_DVR_GetFileByName(userId, srcfile, destfile)) < 0 )
    {
        printf( "GetFileByName failed. error[%d]\n", NET_DVR_GetLastError());
        bRes= -1;
        return bRes;
    }

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

    int nPos = 0;
    for(nPos = 0;  nPos < 100&&npos>=0; nPos = NET_DVR_GetDownloadPos(hPlayback))
    {
        Sleep(5000);  //millisecond
    }
    printf("have got %d\n", nPos);

    if(!NET_DVR_StopGetFile(hPlayback))
    {
        printf("failed to stop get file [%d]\n",NET_DVR_GetLastError());
        bRes = -1;
        return bRes;
    }
    printf("%s\n",srcfile);

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

```
    else
    {
        return 0;
    }
}

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

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

    NET_DVR_FILECOND struFileCond;
    struFileCond.dwFileType = 0xFF;
    struFileCond.IChannel = 1;
    struFileCond.dwIsLocked = 0xFF;
    struFileCond.dwUseCardNo = 0;
    struFileCond.struStartTime.dwYear    = 2011;
    struFileCond.struStartTime.dwMonth   = 3;
    struFileCond.struStartTime.dwDay     = 1;
    struFileCond.struStartTime.dwHour    = 10;
    struFileCond.struStartTime.dwMinute  = 6;
    struFileCond.struStartTime.dwSecond  = 50;
    struFileCond.struStopTime.dwYear     = 2011;
    struFileCond.struStopTime.dwMonth    = 3;
    struFileCond.struStopTime.dwDay      = 1;
    struFileCond.struStopTime.dwHour     = 11;
    struFileCond.struStopTime.dwMinute   = 7;
    struFileCond.struStopTime.dwSecond   = 0;
    //-----
}
```

```
//查找录像文件
int IFindHandle = NET_DVR_FindFile_V30(IUserID, &struFileCond);
if(IFindHandle < 0)
{
    printf("find file fail,last error %d\n",NET_DVR_GetLastError());
    return;
}
NET_DVR_FINDDATA_V30 struFileData;
while(true)
{
    int result = NET_DVR_FindNextFile_V30(IFindHandle, &struFileData);
    if(result == NET_DVR_ISFINDING)
    {
        continue;
    }
    else if(result == NET_DVR_FILE_SUCCESS)
    {
        char strFileName[256] = {0};
        sprintf(strFileName, "%s", struFileData.sFileName);
        saveRecordFile(IUserID, struFileData.sFileName, strFileName);
        break;
    }
    else if(result == NET_DVR_FILE_NOFIND || result == NET_DVR_NOMOREFILE)
    {
        break;
    }
    else
    {
        printf("find file fail for illegal get file state");
        break;
    }
}
//停止查找
if(IFindHandle > 0)
{
    NET_DVR_FindClose_V30(IFindHandle);
}

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

示例二：按时间播放录像文件

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

void main() {

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

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

    NET_DVR_TIME struStartTime, struStopTime;
    struStartTime.dwYear    = 2011;
    struStartTime.dwMonth   = 3;
    struStartTime.dwDay     = 1;
    struStartTime.dwHour    = 9;
    struStartTime.dwMinute  = 0;
    struStartTime.dwSecond  = 0;
    struStopTime.dwYear    = 2011;
    struStopTime.dwMonth   = 3;
    struStopTime.dwDay     = 1;
    struStopTime.dwHour    = 10;
    struStopTime.dwMinute  = 7;
    struStopTime.dwSecond  = 0;
    HWND hWnd = GetConsoleWindow();    //获取窗口句柄
```

```
//-----  
//按时间回放  
int hPlayback;  
hPlayback = NET_DVR_PlayBackByTime(IUserID, 1, &struStartTime, &struStopTime, hWnd);  
if(hPlayback < 0)  
{  
    printf("NET_DVR_GetFileByTime fail,last error %d\n",NET_DVR_GetLastError());  
    NET_DVR_Logout(IUserID);  
    NET_DVR_Cleanup();  
    return;  
}  
  
//-----  
//开始  
if(!NET_DVR_PlayBackControl(hPlayback, NET_DVR_PLAYSTART, 0, 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_TIME struStartTime, struStopTime;
    struStartTime.dwYear    = 2011;
    struStartTime.dwMonth   = 3;
    struStartTime.dwDay     = 1;
    struStartTime.dwHour    = 9;
    struStartTime.dwMinute  = 0;
    struStartTime.dwSecond  = 0;
    struStopTime.dwYear     = 2011;
    struStopTime.dwMonth    = 3;
    struStopTime.dwDay      = 1;
    struStopTime.dwHour     = 10;
    struStopTime.dwMinute   = 7;
    struStopTime.dwSecond   = 0;

    //-----
    //按时间下载
    int hPlayback;
    hPlayback = NET_DVR_GetFileByTime(IUserID, 1, &struStartTime, &struStopTime, "./test.mp4");
    if(hPlayback < 0)
    {
        printf("NET_DVR_GetFileByTime fail,last error %d\n",NET_DVR_GetLastError());
    }
}
```



```
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

//-----
//开始下载
if(!NET_DVR_PlayBackControl(hPlayback, NET_DVR_PLAYSTART, 0, NULL))
{
    printf("play back control failed [%d]\n", NET_DVR_GetLastError());
    NET_DVR_Logout(lUserID);
    NET_DVR_Cleanup();
    return;
}

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

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

4.3 参数配置模块的示例代码

[相关模块流程图](#)

配置压缩参数（NET_DVR_COMPRESSIONCFG_V30）

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

void main() {

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

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

    int iRet;
    //获取压缩参数
    DWORD dwReturnLen;
    NET_DVR_COMPRESSIONCFG_V30 struParams = {0};
    iRet = NET_DVR_GetDVRConfig(lUserID, 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_V30(lUserID);
        NET_DVR_Cleanup();
    }
}
```

```
        return;
    }

    //设置压缩参数
    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_V30(IUserID);
        NET_DVR_Cleanup();
        return;
    }

    //获取压缩参数
    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_V30(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;
}
```

4.4 远程设备维护模块的示例代码

[相关模块流程图](#)

日志查询

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

```
using namespace std;

void main() {

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

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

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

    struStopTime.dwYear     = 2011;
    struStopTime.dwMonth    = 3;
    struStopTime.dwDay      = 2;
    struStopTime.dwHour     = 9;
    struStopTime.dwMinute   = 10;
    struStopTime.dwSecond   = 0;

    //-----
    //查找日志
    int lFindHandle = NET_DVR_FindDVRLog_V30(lUserID, 0, 0, 0, &struStartTime, &struStopTime, FALSE);
    if(lFindHandle < 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;
}
```

4.5 语音对讲转发模块的示例代码

[相关模块流程图](#)

示例一：语音对讲

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

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

void main() {

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

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

    //语音对讲
    LONG lVoiceHanle;
    lVoiceHanle = NET_DVR_StartVoiceCom_V30(lUserID, 1, 0, fVoiceDataCallBack, NULL);
    if (lVoiceHanle < 0)
    {
```

```
    printf("NET_DVR_StartVoiceCom_V30 error, %d!\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

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

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

4.6 报警模块的示例代码

[相关模块流程图](#)

布防报警的示例代码:

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

void CALLBACK MessageCallback(LONG ICommand, NET_DVR_ALARMER *pAlarmer, char *pAlarmInfo, DWORD
dwBufLen, void* pUser)
{
    int i;
    NET_DVR_ALARMINFO struAlarmInfo;
    memcpy(&struAlarmInfo, pAlarmInfo, sizeof(NET_DVR_ALARMINFO));
    switch(ICommand)
    {

```

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

break;
default:
break;
}
}

void main() {

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

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



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

//启用布防
LONG IHandle;
IHandle = NET_DVR_SetupAlarmChan_V30(IUserID);
if (IHandle < 0)
{
    printf("NET_DVR_SetupAlarmChan_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

Sleep(5000);
//撤销布防上传通道
if (!NET_DVR_CloseAlarmChan_V30(IHandle))
{
    printf("NET_DVR_CloseAlarmChan_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

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

监听报警的示例代码:

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

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

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

void main() {

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

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

```
        return;
    }

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

    //启用监听
    LONG IHandle;
    IHandle = NET_DVR_StartListen_V30(NULL, 7200, MessageCallback, NULL);
    if (IHandle < 0)
    {
        printf("NET_DVR_SetupAlarmChan_V30 error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Logout(IUserID);
        NET_DVR_Cleanup();
        return;
    }

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

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

4.7 透明通道模块的示例代码

[相关模块流程图](#)

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

//回调透传数据函数的外部实现
```

```
void CALLBACK g_fSerialDataCallBack(LONG lSerialHandle, char *pRecvDataBuffer, DWORD dwBufSize, DWORD dwUser)
{
    //..... 处理接收到的透传数据，pRecvDataBuffer 中存放接收到的数据
}

void main() {

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

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

    //设置 232 为透明通道模式（使用 232 透明通道时调用，485 不需要）
    DWORD dwReturned = 0;
    NET_DVR_RS232CFG_V30 struRS232Cfg;
    memset(&struRS232Cfg, 0, sizeof(NET_DVR_RS232CFG_V30));
    if (!NET_DVR_GetDVRConfig(lUserID, NET_DVR_GET_RS232CFG_V30, 0, &struRS232Cfg,
sizeof(NET_DVR_RS232CFG_V30), &dwReturned))
    {
        printf("NET_DVR_GET_RS232CFG_V30 error, %d\n", NET_DVR_GetLastError());
        NET_DVR_Logout(lUserID);
        NET_DVR_Cleanup();
        return;
    }

    struRS232Cfg.struRs232.dwWorkMode = 2;
    //设置 232 为透明通道模式；0：窄带传输，1：控制台，2：透明通道
    if (!NET_DVR_SetDVRConfig(lUserID, NET_DVR_SET_RS232CFG_V30, 0, &(struRS232Cfg),
sizeof(NET_DVR_RS232CFG)))
```

```
{
    printf("NET_DVR_SET_RS232CFG_V30 error, %d\n", NET_DVR_GetLastError());
    NET_DVR_Logout(IUserID);
    NET_DVR_Cleanup();
    return;
}

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

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

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

5 函数说明

5.1 SDK 初始化

5.1.1 初始化 SDK **NET_DVR_Init**

函 数: BOOL NET_DVR_Init()
参 数: 无
返回值: TRUE 表示成功, FALSE 表示失败。
说 明: 调用设备网络 SDK 其他函数的前提

[返回目录](#)

5.1.2 释放 SDK 资源 **NET_DVR_Cleanup**

函 数: BOOL NET_DVR_Cleanup()
参 数: 无
返回值: TRUE 表示成功, FALSE 表示失败。
说 明: 在结束之前最后调用。获取错误码调用 [NET_DVR_GetLastError](#)。

[返回目录](#)

5.1.3 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。获取错误码调用 [NET_DVR_GetLastError](#)。
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制值作为实际的超时时间。

[返回目录](#)

5.1.4 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。获取错误码调用 [NET_DVR_GetLastError](#)。

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](#)

5.1.5 通过解析服务器，获取设备的动态 IP 地址和端口号

NET_DVR_GetDVRIPByResolveSvr_EX

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

参 数： `[in]sServerIP` 解析服务器的 IP 地址
`[in]wServerPort` 解析服务器的端口号，IP Server 解析服务器端口号为 7071，EasyDDNS 服务器的端口号为 80
`[in]sDVRName` 设备名称
`[in]wDVRNameLen` 设备名称的长度
`[in]sDVRSerialNumber` 设备的序列号
`[in]wDVRSerialLen` 设备序列号的长度
`[out]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。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明： 该接口中的设备名称和设备序列号不能同时为空。IPServer 是我公司提供的一款域名解析服务器软件。

[返回目录](#)

5.2 异常消息回调

5.2.1 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 留
接收异常消息的回调函数,回调当前异常的相关信息
[in]pUser 用户数据

```
typedef void(CALLBACK* fExceptionCallBack)(DWORD dwType, LONG  
lUserID, LONG lHandle, void *pUser)
```

[out]dwType 异常或重连等消息的类型,见下文的异常消息宏定义表

[out]lUserID 登录 ID

[out]lHandle 出现异常的相应类型的句柄

[out]pUser 用户数据

dwType 宏定义	宏定义值	含义
EXCEPTION_EXCHANGE	0x8000	用户交互时异常(注册心跳超时,心跳间隔为 2 分钟)
EXCEPTION_AUDIOEXCHANGE	0x8001	语音对讲异常
EXCEPTION_ALARM	0x8002	报警异常
EXCEPTION_PREVIEW	0x8003	网络预览异常
EXCEPTION_SERIAL	0x8004	透明通道异常
EXCEPTION_RECONNECT	0x8005	预览时重连
EXCEPTION_ALARMRECONNECT	0x8006	报警时重连
EXCEPTION_SERIALRECONNECT	0x8007	透明通道重连
SERIAL_RECONNECTSUCCESS	0x8008	透明通道重连成功
EXCEPTION_PLAYBACK	0x8010	回放异常
EXCEPTION_DISKFMT	0x8011	硬盘格式化
EXCEPTION_PASSIVEDECODE	0x8012	被动解码异常
EXCEPTION_EMAILTEST	0x8013	邮件测试异常
EXCEPTION_BACKUP	0x8014	备份异常
PREVIEW_RECONNECTSUCCESS	0x8015	预览时重连成功
ALARM_RECONNECTSUCCESS	0x8016	报警时重连成功
RESUME_EXCHANGE	0x8017	用户交互恢复
NETWORK_FLOWTEST_EXCEPTION	0x8018	网络流量检测异常
EXCEPTION_PICPREVIEWRECONNECT	0x8019	图片预览重连
PICPREVIEW_RECONNECTSUCCESS	0x8020	图片预览重连成功
EXCEPTION_PICPREVIEW	0x8021	图片预览异常

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： Windows 下该函数的 hWnd 和 cbExceptionCallBack 不能同时为 NULL，Linux 下 cbExceptionCallBack 不能设置为 NULL，否则将接收不到异常消息。
如果此结构是以回调方式反馈异常消息，那么应用程序中的异常回调函数实现如下，该函数中的参数 dwType 表示异常消息类型（见上表）；IHandle 表示发生异常的相应类型的句柄。

示例代码：

```
//注册接收异常消息的回调函数
NET_DVR_SetExceptionCallBack_V30(WM_NULL, NULL, g_ExceptionCallBack, NULL);
//接收异常消息的回调函数的外部实现
void CALLBACK g_ExceptionCallBack(DWORD dwType, LONG IUserID, LONG IHandle, void *pUser)
{
    char tempbuf[256];
    ZeroMemory(tempbuf,256);
    switch(dwType)
    {
        case EXCEPTION_AUDIOEXCHANGE:                //语音对讲时网络异常
            sprintf(tempbuf,"语音对讲时网络异常!!!");
            TRACE("%s",tempbuf);
            //TODO: 关闭语音对讲
            break;
        case EXCEPTION_ALARM:                          //报警上传时网络异常
            sprintf(tempbuf,"报警上传时网络异常!!!");
            TRACE("%s",tempbuf);
            //TODO: 关闭报警上传
            break;
        case EXCEPTION_PREVIEW:                        //网络预览时异常
            sprintf(tempbuf,"网络预览时网络异常!!!");
            TRACE("%s",tempbuf);
            //TODO: 关闭网络预览
            break;
        case EXCEPTION_SERIAL:                          //透明通道传输时异常
            sprintf(tempbuf,"透明通道传输时网络异常!!!");
            TRACE("%s",tempbuf);
            //TODO: 关闭透明通道
            break;
        case EXCEPTION_RECONNECT:                      //预览时重连
            break;
        default:
            break;
    }
};
```

[返回目录](#)

5.3 SDK 日志和信息

5.3.1 获取 SDK 的版本信息 **NET_DVR_GetSDKVersion**

函 数: DWORD NET_DVR_GetSDKVersion()
参 数:
返回值: SDK 版本信息。
说 明: SDK 版本信息, 2 个高字节表示主版本, 2 个低字节表示次版本。如 0x00030000: 表示版本为 3.0。

[返回目录](#)

5.3.2 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](#)

5.3.3 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。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

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

函 数: BOOL NET_DVR_GetSDKAbility(LPNET_DVR_SDKABL pSDKAbI)
参 数: [out] pSDKAbI 功能信息结构
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

5.3.5 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。

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，同时最多支持创建 10 个文件，当设置了删除超出的文件时（即 **bAutoDel** 为 **TRUE**），那么将会自动删除超出的文件。更改目录时到下一次写文件时才会使用新的目录写文件。

[返回目录](#)

5.4 获取错误信息

5.4.1 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](#)

5.4.2 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](#)

5.5 用户注册

5.5.1 用户注册设备 **NET_DVR_Login_V30**

函 数： LONG NET_DVR_Login_V30(char *sDVRIP,WORD wDVRPort, char *sUserName, char *sPassword,LPNET_DVR_DEVICEINFO_V30 lpDeviceInfo)

参 数： [in]sDVRIP 设备 IP 地址
[in]wDVRPort 设备端口号
[in]sUserName 登录的用户名
[in]sPassword 用户密码
[out]lpDeviceInfo 设备信息

返回值： -1 表示失败，其他值表示返回的用户 ID 值。该用户 ID 具有唯一性，后续对设备的操作都需要通过此 ID 实现。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明： 网络摄像机和网络球机允许有 16 个注册用户，且同时最多允许 128 个用户注册。SDK 支持 512 个注册，返回 UserID 逐一递增，从 0 到 511 之后又返回 0，注销以及 NET_DVR_Cleanup 均没有初始化为 0。

[返回目录](#)

5.5.2 用户注销 **NET_DVR_Logout**

函 数： BOOL NET_DVR_Logout(LONG lUserID)

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

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

说 明： 建议使用此接口实现注销功能。

[返回目录](#)

5.6 获取设备能力集

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

函 数： BOOL NET_DVR_GetDeviceAbility(LONG lUserID, DWORD dwAbilityType, char* pInBuf, DWORD dwInLength, char* pOutBuf, DWORD dwOutLength)

参 数： [in]lUserID NET_DVR_Login_V30 的返回值
[in]dwAbilityType 能力类型，具体定义见下表：
[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）
[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）

[in]dwOutLength 接收数据的缓冲区的长度

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	0x003	设备所有编码能力
DEVICE_ENCODE_CURRENT	0x004	设备当前编码能力
COMPRESSIONCFG_ABILITY	0x400	压缩参数能力
COMPRESSION_LIMIT	0x401	主子码流压缩参数能力限制
PIC_CAPTURE_ABILITY	0x402	抓图图片分辨率能力集

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

说 明: 接口中 pInBuf 参数的具体定义格式按照不同的设备规定有所不同, 可以是以结构体的形式或者 XML 描述方式。同样地, 参数 pOutBuf 的输出表达格式也按不同的设备规定可以是以结构体的形式或者 XML 描述方式。前 5 种能力描述都是以 XML 格式给出, 详见各产品的能力描述解析方法, 这里不进行详细说明。下表列出了其余能力类型获取时, 需要输入参数和输出参数的格式定义。下表列出了其余能力类型获取时, 需要输入参数和输出参数的格式定义。

能力类型宏定义	能力类型说明	pInBuf	pOutBuf
DEVICE_SOFTHARDWARE_ABILITY	获取设备软硬件能力	无	设备软硬件能力 XML 描述
DEVICE_NETWORK_ABILITY	获取设备网络能力	无	设备网络能力 XML 描述
DEVICE_ENCODE_ALL_ABILITY	获取设备所有编码能力	无	设备所有编码能力 XML 描述
DEVICE_ENCODE_CURRENT	获取设备当前编码能力	当前编码类型和分辨率 XML 描述	设备当前编码能力 XML 描述
DEVICE_RAID_ABILITY	获取设备 RAID 能力	无	设备 RAID 能力 XML 描述
COMPRESSIONCFG_ABILITY	获取压缩参数能力	通道号 (4 个字节)	NET_DVR_COMPRESSIONCFG_ABILITY
COMPRESSION_LIMIT	获取主子码流压缩参数能力限制	NET_DVR_COMPRESSION_LIMIT	NET_DVR_COMPRESSIONCFG_ABILITY
PIC_CAPTURE_ABILITY	获取图片能力	通道号 (4 个字节)	NET_DVR_COMPRESSIONCFG_ABILITY

[返回目录](#)

5.7 实时预览

5.7.1 设置播放显示模式 **NET_DVR_SetShowMode**

函 数: BOOL NET_DVR_SetShowMode (DWORD dwShowType, COLORREF colorKey)

参 数: [in]dwShowType 显示模式

```
enum{
    NORMALMODE = 0,
    OVERLAYMODE
}
```

[in]colorKey

用户设置的透明色，在 OVERLAY 模式下需要设置，colorKey 是一个 32 位的值 0x00bbggrr，最高字节为 0，后三个字节分别表示 b、g、r 的值

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:

- 设置播放器显示模式，需在预览之前设置。
- NORMALMODE 模式可以同时显示多窗口，但是对显卡有一定的要求
- OVERLAYMODE 模式只能同时显示一个窗口，但是对显卡基本没有要求。透明色相当于一层透视膜，显示的画面只能穿过这种颜色，而其他的颜色将挡住显示的画面，用户应该在显示窗口中涂上这种颜色才能看到显示画面，一般应该使用一种不常用的颜色作为透明色，colorKey 是一个 32 位的值 0x00bbggrr，最高字节为 0，后三个字节分别表示 b、g、r 的值。
- 播放器有两种显示模式：普通模式和 OVERLAY 方式，使用 OVERLAY 模式的优点是：大部分显卡都支持 OVERLAY，在一些不支持 BLT 硬件缩放和颜色转换的显卡上(如 SIS 系列显卡)使用 OVERLAY 模式，可以大大降低 CPU 利用率并提高画面质量(相对于软件实现缩放、颜色转换)。缺点是：同时只能播放一路图象，不能实现大规模集中监控。在一块显卡中同一时刻只能有一个 OVERLAY 表面处于活动状态，如果此时系统中已经有程序使用了 OVERLAY，那么播放器就不能再创建 OVERLAY 表面，它将自动改成普通的模式，并不返回 FALSE，一些常用的播放器，例如我司板卡的预览都可能使用了 OVERLAY 表面，同样，如果我们的 SDK 中使用了 OVERLAY 表面，那么其他的程序将不能再使用 OVERLAY 表面。

[返回目录](#)

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

函 数: BOOL NET_DVR_MakeKeyFrame(LONG lUserID, DWORD lChannel)

参 数: [in]lUserID NET_DVR_Login_V30 的返回值
[in]lChannel 通道号，网络摄像机和球机只有一个通道，通道号为 1

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: 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

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。

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](#)

5.7.5 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_V30` 的返回值
返回值： `TRUE` 表示成功，`FALSE` 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明：

[返回目录](#)

5.7.6 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_V30` 的返回值
返回值： `-1` 表示失败，其他值表示播放句柄。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明： 用户可以通过返回的句柄自行实现播放库 SDK 提供的其他功能，详见本公司提供的软解码库函数说明《播放器 SDK 编程指南》。例如使用
`PlayM4_GetBMP(LONG nPort,.....)`、
`PlayM4_GetJPEG(LONG nPort,.....)`这两个接口时，即可实现将当前预览图像以 BMP 或 JPEG 格式抓图保存到内存中：
`PlayM4_GetBMP(NET_DVR_GetRealPlayerIndex(),.....)`
`PlayM4_GetJPEG(NET_DVR_GetRealPlayerIndex(),.....)`

[返回目录](#)

5.8 预览显示视频参数配置

5.8.1 获取预览视频显示参数 `NET_DVR_ClientGetVideoEffect`

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

参 数: [in]IRealHandle NET_DVR_RealPlay_V30 的返回值
 [out]pBrightValue 亮度指针, 取值范围[1,10]
 [out]pContrastValue 对比度指针, 取值范围[1,10]
 [out]pSaturationValue 饱和度指针, 取值范围[1,10]
 [out]pHueValue 色度指针, 取值范围[1,10]
 返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
 说 明: 需要预览才能获取视频参数。

[返回目录](#)

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

函 数: BOOL NET_DVR_GetVideoEffect(LONG IUserID, LONG IChannel,DWORD
 *pBrightValue, DWORD *pContrastValue,DWORD *pSaturationValue,DWORD
 *pHueValue)
 参 数: [in]IRealHandle NET_DVR_Login_V30 的返回值
 [in]IChannel 通道号
 [out] pBrightValue 亮度指针, 取值范围[1,10]
 [out] pContrastValue 对比度指针, 取值范围[1,10]
 [out] pSaturationValue 饱和度指针, 取值范围[1,10]
 [out] pHueValue 色度指针, 取值范围[1,10]
 返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
 说 明: 登录设备获取通道的视频参数。

[返回目录](#)

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

函 数: BOOL NET_DVR_ClientSetVideoEffect(LONG IRealHandle,DWORD pBrightValue,
 DWORD pContrastValue,DWORD pSaturationValue,DWORD pHueValue)
 参 数: [in]IRealHandle NET_DVR_RealPlay_V30 的返回值
 [in]dwBrightValue 亮度, 取值范围[1,10]
 [in]dwContrastValue 对比度, 取值范围[1,10]
 [in]dwSaturationValue 饱和度, 取值范围[1,10]
 [in]dwHueValue 色度, 取值范围[1,10]
 返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
 说 明: 需要预览才能设置视频参数。

[返回目录](#)

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

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

参 数: [in]IRealHandle NET_DVR_Login_V30 的返回值
[in]IChannel 通道号
[in]dwBrightValue 亮度, 取值范围[1,10]
[in]dwContrastValue 对比度, 取值范围[1,10]
[in]dwSaturationValue 饱和度, 取值范围[1,10]
[in]dwHueValue 色度, 取值范围[1,10]

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

说 明: 登录设备设置通道的视频参数。

[返回目录](#)

5.9 预览画面叠加字符和图像

5.9.1 预览画面叠加字符和图像, Linux 下无此接口

NET_DVR_RegisterDrawFun

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

参 数: [in]IRealHandle NET_DVR_RealPlay_V30 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。获取错误码调用 [NET_DVR_GetLastError](#)。

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_V30 进行预览, 参数 *bBlocked* 必须置 1 (TRUE), 否则该接口调用会失败, 获取错误号为 12 (调用次序错误)。

[返回目录](#)

5.10 预览时解码效果的参数控制

5.10.1 设置播放库的帧缓冲区个数 NET_DVR_SetPlayerBufNumber

函 数: BOOL NET_DVR_SetPlayerBufNumber(LONG IRealHandle, DWORD dwBufNum)

参 数: [in]IRealHandle NET_DVR_RealPlay_V30 的返回值

[in]dwBufNum 所要设置的单视频播放时缓冲区最大的帧数，取值范围[1,50]，SDK 默认的帧缓冲区大小为 15

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: 设置网络延时和播放流畅度可以通过此接口来进行调节。dwBufNum 值越大，播放的流畅性越好，相对延时就大；dwBufNum 值越小，播放的延时就小，但是当网络不太顺畅的时候，会有丢帧现象，影响播放的流畅性。若当前为混合流时，为保证音视频同步效果建议设置缓冲帧数大于等于 6 帧。此函数必须紧跟在 [NET_DVR_RealPlay_V30](#) 后使用，在图像播放之后设置则不起作用。

[返回目录](#)

5.10.2 设置解码时丢弃 B 帧的个数 [NET_DVR_ThrowBFrame](#)

函 数: BOOL NET_DVR_ThrowBFrame(LONG lRealHandle,DWORD dwNum)

参 数: [in]lRealHandle NET_DVR_RealPlay_V30 的返回值
[in]dwNum 丢弃 B 帧的个数: 0—不丢; 1—丢 1 个 B 帧;
2—丢 2 个 B 帧

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

说明: 在多路播放时，丢 B 帧可以降低 CPU 的利用率。

[返回目录](#)

5.11 预览时播放声音控制

5.11.1 设置声音播放模式 [NET_DVR_SetAudioMode](#)

函 数: BOOL NET_DVR_SetAudioMode(DWORD dwMode)

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

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

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](#)

5.11.2 独占声卡模式下开启声音 [NET_DVR_OpenSound](#)

函 数: BOOL NET_DVR_OpenSound(LONG lRealHandle)

参 数: [in]lRealHandle NET_DVR_RealPlay_V30 的返回值

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

说明: 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](#)

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

函 数: BOOL NET_DVR_CloseSound()

参 数: 无

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

说 明:

[返回目录](#)

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

函 数: BOOL NET_DVR_OpenSoundShare(LONG IRealHandle)

参 数: [in]IRealHandle NET_DVR_RealPlay_V30 的返回值

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

说 明:

[返回目录](#)

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

函 数: BOOL NET_DVR_CloseSoundShare (LONG IRealHandle)

参 数: [in]IRealHandle NET_DVR_RealPlay_V30 的返回值

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

说 明:

[返回目录](#)

5.11.6 调节播放音量 **NET_DVR_Volume**

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

参 数: [in]IRealHandle NET_DVR_RealPlay_V30 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。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明:

[返回目录](#)

5.12 实时预览数据捕获

5.12.1 注册回调函数, 捕获实时码流数据

NET_DVR_SetRealDataCallBack

函 数: BOOL NET_DVR_SetRealDataCallBack(LONG IRealHandle, fRealDataCallBack

cbRealDataCallBack,DWORD dwUser)

参 数: [in]IRealHandle 预览句柄, NET_DVR_RealPlay_V30 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
[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

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。获取错误码调用 [NET_DVR_GetLastError](#)。

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。

[返回目录](#)

5.12.2 注册回调函数, 捕获实时码流数据（标准码流）

NET_DVR_SetStandardDataCallBack

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

参 数: [in]IRealHandle 预览句柄, NET_DVR_RealPlay_V30 的返回值
[in]cbStdDataCallBack 标准码流回调函数
[in]dwUser 用户数据

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

[out]IRealHandle 当前的预览句柄
[out]dwDataType 数据类型
[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

dwDataType 宏定义	宏定义值	含义
NET_DVR_SYSHEAD	1	系统头数据

NET_DVR_STD_VIDEODATA	4	标准视频流数据
NET_DVR_STD_AUDIODATA	5	标准音频流数据
NET_DVR_PRIVATE_DATA	2 或者 112	私有数据，包括智能信息叠加等

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

说明： 此函数包括开始和停止用户处理 SDK 捕获的数据，当回调函数 cbStdDataCallBack 设为非 NULL 值时，表示回调和处理数据；当设为 NULL 时表示停止回调和处理数据。回调的第一个包是 40 个字节的文件头，供后续解码使用，之后回调的是标准码流（含 12 字节的 RTP 头）。

此函数仅支持对于支持 RTSP 协议取流的设备的标准码流回调。

[返回目录](#)

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

函 数： BOOL NET_DVR_SaveRealData(LONG IRealHandle, char *sFileName)

参 数： [in]IRealHandle NET_DVR_RealPlay_V30 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。获取错误码调用 [NET_DVR_GetLastError](#)。

说明：

[返回目录](#)

5.12.4 停止数据捕获 **NET_DVR_StopSaveRealData**

函 数： BOOL NET_DVR_StopSaveRealData(LONG IRealHandle)

参 数： [in]IRealHandle NET_DVR_RealPlay_V30 的返回值

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

说明：

[返回目录](#)

5.13 视频抓图

5.13.1 设置抓图模式 **NET_DVR_SetCapturePictureMode**

函 数： BOOL NET_DVR_SetCapturePictureMode(DWORD dwCaptureMode)

参 数： [in]dwCaptureMode 抓图模式

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

dwCaptureMode 抓图模式：

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

[}CAPTURE_MODE](#)[返回目录](#)

5.13.2 预览时，单帧数据捕获并保存成图片

NET_DVR_CapturePicture

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

参 数: [in]lRealHandle NET_DVR_RealPlay_V30 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。获取错误码调用 [NET_DVR_GetLastError](#)。

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。

[返回目录](#)

5.13.3 单帧数据捕获并保存成 JPEG 图片

NET_DVR_CaptureJPEGPicture

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

参 数: [in]lUserID NET_DVR_Login_V30 的返回值
[in]lChannel 通道号
[in]lpJpegPara JPEG 图像参数
[in]sPicFileName 保存 JPEG 图的文件路径

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

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

[返回目录](#)

5.13.4 单帧数据捕获并保存成 JPEG 存放在指定的内存空间中

NET_DVR_CaptureJPEGPicture_NEW

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

参 数: [in]lUserID NET_DVR_Login_V30 的返回值

[in]IChannel	通道号
[in]lpJpegPara	JPEG 图像参数
[in]sJpegPicBuffer	保存 JPEG 数据的缓冲区
[in]dwPicSize	输入缓冲区大小
[out]lpSizeReturned	返回图片数据的大小

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

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

[返回目录](#)

5.14 录像文件回放、下载、锁定及备份

录像文件的查找

5.14.1 根据文件类型、时间查找设备录像文件

NET_DVR_FindFile_V40

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

参 数： [in]IUserID NET_DVR_Login_V30 的返回值
 [in]pFindCond 待查找的文件信息结构

返回值： -1 表示失败，其他值作为 NET_DVR_FindClose 等函数的参数。获取错误码调用 [NET_DVR_GetLastError](#)。

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_V30](#) 接口来获取文件信息。

[返回目录](#)

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

函 数： LONG NET_DVR_FindNextFile_V30(LONG IFindHandle,
 LPNET_DVR_FINDDATA_V30 lpFindData)

参 数： [in]IFindHandle 文件查找句柄，NET_DVR_FindFile_V30()的返回值
 [in]lpFindData 保存文件信息的指针

返回值： -1 表示失败，其他值表示当前的获取状态等信息，详见下表。

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_FindFile_V30 得到当前的查找句柄。此接口用于获取一条已查找到的文件信息，若要获取全部的已查找到的文件信息，需要循环调用此接口。通过此接口可以同时获取到与当前录像文件相关的卡号信息和文件是否被锁定的信息。

每次可查询文件最大个数为 4000。

[返回目录](#)

5.14.3 关闭文件查找，释放资源 NET_DVR_FindClose_V30

函 数： BOOL NET_DVR_FindClose_V30(LONG IFindHandle)

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

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

说明：

[返回目录](#)

回放录像文件

5.14.4 按文件名回放录像文件 NET_DVR_PlayBackByName

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

参 数： [in]IUserID NET_DVR_Login_V30 的返回值
[in]sPlayBackFileName 回放的文件名，长度不能超过 100 字节
[in]hWnd 回放的窗口句柄，若置为空，SDK 仍能收到码流数据，但不解码显示

返回值： -1 表示失败，其他值作为 NET_DVR_StopPlayBack 等函数的参数。获取错误码调用 [NET_DVR_GetLastError](#)。

说明： 该接口指定了当前要播放的录像文件，调用成功后，还必须调用 NET_DVR_PlayBackControl_V40 接口的开始播放控制命令（NET_DVR_PLAYSTART）才能实现回放。

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

Linux 下：

对于 4.1 或者以上的版本的 SDK，HWND 表示播放窗口的句柄，定义为：

```
typedef unsigned int HWND;
```

如果使用 Qt 进行界面开发，示例如下：

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

对于 4.1 以前的版本的 SDK，HWND 定义如下：

```
typedef struct __PLAYRECT
{
    int x;           //显示框左上角横坐标
    int y;           //显示框左上角纵坐标
    int uWidth;      //显示框宽度
    int uHeight;     //显示框高度
}PLAYRECT;
typedef PLAYRECT HWND;
```

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

[返回目录](#)

5.14.5 按时间回放录像文件 **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_V30 的返回值
[in]pVodPara 回放参数

返回值： -1 表示失败，其他值作为 NET_DVR_StopPlayBack 等函数的参数。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明： 该接口指定了当前要播放的录像文件，调用成功后，还必须调用 NET_DVR_PlayBackControl_V40 接口的开始播放控制命令 (NET_DVR_PLAYSTART) 才能实现回放。

当回放的是按事件搜索出的录像文件时，由于每个文件都会有预录和延迟的部分，因此在设置本接口的开始和结束时间参数时可以适当提前开始时间和延长结束时间。建议值：最多 10 分钟，最少 5 秒。

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

Linux 下：

对于 4.1 或者以上的版本的 SDK，HWND 表示播放窗口的句柄，定义为：

```
typedef unsigned int HWND;
```

如果使用 Qt 进行界面开发，示例如下：

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

对于 4.1 以前的版本的 SDK，HWND 定义如下：

```
typedef struct __PLAYRECT
{
    int x;          //显示框左上角横坐标
    int y;          //显示框左上角纵坐标
    int uWidth;     //显示框宽度
    int uHeight;    //显示框高度
}PLAYRECT;
typedef PLAYRECT HWND;
```

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

[返回目录](#)

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

函 数： BOOL NET_DVR_PlayBackControl_V40(LONG lPlayHandle,DWORD dwControlCode, LPVOID lpInBuffer = NULL, DWORD dwInLen = 0, LPVOID lpOutBuffer = NULL, DWORD *lpOutLen = NULL)

参 数： [in]lPlayHandle 播放句柄，NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime 的返回值
[in]dwControlCode 控制录像回放状态命令，见下表
[in] lpInBuffer 指向输入参数的指针
[in]dwInLen 输入参数的长度
[out]lpOutBuffer 指向输出参数的指针
[out]lpOutLen 输出参数的长度

dwControlCode 宏定义	宏定义值	含义
NET_DVR_PLAYSTART	1	开始播放
NET_DVR_PLAYPAUSE	3	暂停播放
NET_DVR_PLAYRESTART	4	恢复播放
NET_DVR_PLAYFAST	5	快放
NET_DVR_PLAYSLOW	6	慢放
NET_DVR_PLAYNORMAL	7	正常速度播放（在暂停后调用将恢复暂停前的速度播放）
NET_DVR_PLAYFRAME	8	单帧放（恢复正常回放使用 NET_DVR_PLAYNORMAL 命令）
NET_DVR_PLAYSTARTAUDIO	9	打开声音
NET_DVR_PLAYSTOPAUDIO	10	关闭声音
NET_DVR_PLAYAUDIOVOLUME	11	调节音量，取值范围[0,0xffff]

NET_DVR_PLAYSETPOS	12	改变文件回放的进度
NET_DVR_PLAYGETPOS	13	获取按文件或者按时间回放的进度
NET_DVR_PLAYGETTIME	14	获取当前已经播放的时间(按文件回放的时候有效)
NET_DVR_PLAYGETFRAME	15	获取当前已经播放的帧数(按文件回放的时候有效)
NET_DVR_GETTOTALFRAMES	16	获取当前播放文件总的帧数(按文件回放的时候有效)
NET_DVR_GETTOTALTIME	17	获取当前播放文件总的时间(按文件回放的时候有效)
NET_DVR_THROWBFRAME	20	丢 B 帧
NET_DVR_SETSPEED	24	设置码流速度
NET_DVR_KEEPAIVE	25	保持与设备的心跳(如果回调阻塞, 建议 2 秒发送一次)

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: 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, 详见下表所示。

状态命令宏定义	状态命令说明	IpInBuf	IpOutBuf
NET_DVR_PLAYSTART	开始播放	一个 4 字节整型的偏移量	无
NET_DVR_PLAYSETPOS	改变回放的进度	一个 4 字节整型的进度值(0-100)	无
NET_DVR_PLAYGETPOS	获取回放的进度	无	一个 4 字节整型的进度值(0-100)
NET_DVR_PLAYGETTIME	获取当前已播放的时间 (按文件回放有效)	无	一个 4 字节整型的时间值
NET_DVR_PLAYGETFRAME	获取当前已播放的帧数 (按文件回放有效)	无	一个 4 字节整型的帧数值
NET_DVR_GETTOTALFRAMES	获取当前播放文件总的帧数 (按文件回放有效)	无	一个 4 字节整型的帧数值
NET_DVR_GETTOTALTIME	获取当前播放文件总的时间 (按文件回放有效)	无	一个 4 字节整型的时间值
NET_DVR_THROWBFRAME	丢 B 帧	一个 4 字节整型的 B 帧个数	无
NET_DVR_SETSPEED	设置码流速度	一个 4 字节整型的速度值	无

- 特别指出, 当控制命令是开始播放(即 NET_DVR_PLAYSTART)时, 第三个参数的值表示播放当前文件的偏移量, 若该值为 0 表示从文件的起始位置播放, 若该值不为 0 则表示断点续传的文件位置(Byte)。

- 该接口中的第五个参数表示当前控制命令操作所获取到的相应的参数, 控

制命令中的 NET_DVR_PLAYGETPOS、NET_DVR_PLAYGETTIME、NET_DVR_PLAYGETFRAME、NET_DVR_GETTOTALFRAMES 和 NET_DVR_GETTOTALTIME 都能通过该参数得到对应的值，详见上表。

- 当命令值为 NET_DVR_PLAYGETPOS 时，获取文件回放或者下载进度时，0-100 表示正常的进度值，大于 100 的值表示回放或者下载异常；获取按时间回放或下载进度时，只能获取的进度值是 0、100（结束）、200（异常）。

[返回目录](#)

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

函 数： BOOL NET_DVR_StopPlayBack(LONG IPlayHandle)
参 数： [in]IPlayHandle 回放句柄，NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime 的返回值
返回值： TRUE 表示成功，FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明：

[返回目录](#)

回放录像文件时的数据捕获

5.14.8 捕获回放的录像数据，并保存成文件

NET_DVR_PlayBackSaveData

函 数： BOOL NET_DVR_PlayBackSaveData(LONG IPlayHandle,char *sFileName)
参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime 的返回值
[in]sFileName 保存数据的文件路径
返回值： TRUE 表示成功，FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明：

[返回目录](#)

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

函 数： BOOL NET_DVR_StopPlayBackSave(LONG IPlayHandle)
参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或 NET_DVR_PlayBackByTime 的返回值
返回值： TRUE 表示成功，FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明：

[返回目录](#)

5.14.10 注册回调函数，捕获录像数据 **NET_DVR_SetPlayDataCallBack**

函 数： BOOL NET_DVR_SetPlayDataCallBack(LONG IPlayHandle, fPlayDataCallBack
cbPlayDataCallBack,DWORD dwUser)

参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或
NET_DVR_PlayBackByTime 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，详见下表
[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

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。获取错误码调用 [NET_DVR_GetLastError](#)。

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。

[返回目录](#)

回放的其他操作

5.14.11 获取录像回放时显示的 OSD 时间

NET_DVR_GetPlayBackOsdTime

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

参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或
NET_DVR_PlayBackByTime 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。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明：

[返回目录](#)

5.14.12 录像回放时抓图，并保存在文件中

NET_DVR_PlayBackCaptureFile

函 数： BOOL NET_DVR_PlayBackCaptureFile(LONG IPlayHandle,char *sFileName)
 参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或
 NET_DVR_PlayBackByTime 的返回值
 [in]sFileName 保存图片数据的文件路径
 返回值： TRUE 表示成功，FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
 说 明： 回放时抓下来的图片时间要比抓图时间延后，这是因为预览画面上的 OSD 时间是解码完成的显示时间，而解码缓冲区会有将近 1M 左右的数据还没有解出来，要抓取的图片数据是网络缓冲里面的。目前解码库没有直接从解码缓冲区中取出数据的接口。

[返回目录](#)

5.14.13 刷新显示回放窗口 NET_DVR_RefreshPlay

函 数： BOOL NET_DVR_RefreshPlay(LONG IPlayHandle)
 参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或
 NET_DVR_PlayBackByTime 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： 当用户暂停或者单帧回放时，如果刷新了窗口，则窗口中的图像因为刷新而消失，此时调用这个接口可以重新显示最后一帧画面。此接口只在暂停和单帧播放时有效。

[返回目录](#)

5.14.14 获取回放时用来解码显示的播放库句柄

NET_DVR_GetPlayBackPlayerIndex

函 数： int NET_DVR_GetPlayBackPlayerIndex(LONG IPlayHandle)
 参 数： [in]IPlayHandle 播放句柄，NET_DVR_PlayBackByName 或
 NET_DVR_PlayBackByTime 的返回值
 返回值： -1 表示失败，其他值表示播放句柄。获取错误码调用 [NET_DVR_GetLastError](#)。
 说 明： 用户可以通过返回的句柄自行实现播放库 SDK 提供的其他功能，详见本公司提供的软解码库函数说明《播放器 SDK 编程指南》。
 例如使用 PlayM4_GetBMP(LONG nPort,.....)、PlayM4_GetJPEG(LONG nPort,.....) 这两个接口时，即可实现将当前预览图像以 BMP 或 JPEG 格式抓图保存到内存中： PlayM4_GetBMP(NET_DVR_GetRealPlayerIndex(),.....)
 PlayM4_GetJPEG(NET_DVR_GetRealPlayerIndex(),.....) 。

[返回目录](#)

下载录像文件

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

函 数: LONG NET_DVR_GetFileByName(LONG IUserID,char *sDVRFileName,char *sSavedFileName)

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

返回值: -1 表示失败,其他值作为 NET_DVR_StopGetFile 等函数的参数。获取错误码调用 [NET_DVR_GetLastError](#)。

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

[返回目录](#)

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

函 数: LONG NET_DVR_GetFileByTime(LONG IUserID,LONG IChannel, LPNET_DVR_TIME lpStartTime, LPNET_DVR_TIME lpStopTime, char *sSavedFileName)

参 数: [in]IUserID NET_DVR_Login_V30 的返回值
[in]IChannel 通道号
[in]lpStartTime 开始时间
[in]lpStopTime 结束时间
[in]sSavedFileName 下载后保存到 PC 机的文件路径,需为绝对路径

返回值: -1 表示失败,其他值作为 NET_DVR_StopGetFile 等函数的参数。获取错误码调用 [NET_DVR_GetLastError](#)。

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](#) 接口的开始播放控制命令 (NET_DVR_PLAYSTART) 才能实现下载。

[返回目录](#)

5.14.17 控制录像下载的状态 NET_DVR_PlayBackControl

函 数： BOOL NET_DVR_PlayBackControl(LONG IPlayHandle,DWORD dwControlCode,DWORD dwInValue,DWORD *LPOutValue)

参 数： [in]IPlayHandle 播放句柄，NET_DVR_GetFileByName 或 NET_DVR_GetFileByTime 的返回值

[in]dwControlCode 控制录像下载状态命令，见下表

[in]dwInValue 设置的参数，如设置文件下载的进度(命令值 NET_DVR_PLAYSETPOS)时，此参数表示进度值；如开始下载(命令值 NET_DVR_PLAYSTART)时，此参数表示断点续传的文件位置（Byte）

[out]LPOutValue 获取的参数，如获取当前下载文件总的时间（命令值 NET_DVR_GETTOTALTIME ），此参数就是得到的总时间

dwControlCode 宏定义	宏定义值	含义
NET_DVR_PLAYSTART	1	开始下载
NET_DVR_PLAYPAUSE	3	暂停下载
NET_DVR_PLAYRESTART	4	恢复下载
NET_DVR_PLAYSETPOS	12	改变文件下载的进度（按文件下载时有效）
NET_DVR_PLAYGETPOS	13	获取文件下载的进度（按文件下载时有效）
NET_DVR_GETTOTALFRAMES	16	获取当前下载文件总的帧数(按文件下载时有效)
NET_DVR_GETTOTALTIME	17	获取当前下载文件总的时间(按文件下载时有效)
NET_DVR_SET_DOWNLOAD_SPEED	28	设置下载速度

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

- 说 明：
- 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。
 - NET_DVR_PLAYSETPOS 命令下此参数表示下载的进度；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）(目前只有 DS-90xx 和 DS-81xx 硬盘录像机支持断点续传功能)。
 - 该接口中的第四个参数表示当前控制命令操作所获取到的相应的参数，控制命令中的 NET_DVR_PLAYGETPOS、NET_DVR_GETTOTALFRAMES 和 NET_DVR_GETTOTALTIME 都能通过该参数得到对应的值。
 - 进度值的取值范围为 0-100，获取文件下载进度时，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。

[返回目录](#)

5.14.18 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。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

5.14.19 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,
如返回 200 表明出现网络异常。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明: 该接口用于获取按文件名下载录像文件时的下载进度。

[返回目录](#)

录像文件锁定和解锁

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

函 数: BOOL NET_DVR_LockFileByName(LONG IUserID, char *sLockFileName)
参 数: [in]IUserID NET_DVR_Login_V30 的返回值
要锁定的录像文件名, 文件名长度需小于 100
[in]sLockFileName 字节
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明: 在使用该接口锁定录像文件前, 可以先调用录像文件查找的接口获取文件名。
当文件被锁定后, 将不会被覆盖。

[返回目录](#)

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

函 数: BOOL NET_DVR_UnlockFileByName(LONG IUserID, char *sUnlockFileName)
参 数: [in]IUserID NET_DVR_Login_V30 的返回值
[in]sUnlockFileName 要解锁的录像文件名
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明: 在使用该接口锁定录像文件前, 可以先调用录像文件查找的接口获取文件名。

[返回目录](#)

5.15 手动录像

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

函 数: `BOOL NET_DVR_StartDVRRecord(LONG IUserID, LONG IChannel, LONG IRecordType)`

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

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

说 明:
➤ 录像类型设置需要设备支持, 不支持默认为手动录像。
➤ 当某通道已经开启定时录像的前提下首次开启手动录像, 此次操作未生效, 仍保持定时录像状态, 且查询设备状态(见 `NET_DVR_GetDVRWorkState_V30` 和 `NET_DVR_GetDVRWorkState`; 结构体 `NET_DVR_WORKSTATE_V30` 和 `NET_DVR_WORKSTATE`) 中的录像状态仍为录像; 此时关闭手动录像, 停止了定时录像, 且查询录像状态为不录像; 第二次开启手动录像, 此时手动录像开始; 停止手动录像后, 重启设备, 定时录像重新打开。

[返回目录](#)

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

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

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

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

说 明:

[返回目录](#)

5.16 布防、撤防

设置报警等信息上传的回调函数

5.16.1 注册回调函数，接收设备报警消息

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

ICommand 宏定义	宏定义值	含义
COMM_ALARM	0x1100	V3.0 以下版本支持的设备的报警信息上传
COMM_IPC_AUXALARM_RESULT	0x2820	PIR 报警、无线报警、呼救报警上传

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

消息类型 ICommand	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM	V3.0 以下版本支持的设备的报警信息	NET_DVR_ALARMINFO
COMM_IPC_AUXALAR M_RESULT	PIR 报警、无线报警、呼救报警信息	NET_IPC_AUXALARM_RESU LT

[返回目录](#)

布防撤防

5.16.2 建立报警上传通道，获取报警等信息

NET_DVR_SetupAlarmChan_V30

函 数： BOOL NET_DVR_SetupAlarmChan_V30(LONG IUserID)

参 数: [in]IUserID NET_DVR_Login_V30 的返回值
返回值: -1 表示失败, 其他值作为 NET_DVR_CloseAlarmChan_V30 函数的句柄参数。
获取错误码调用 [NET_DVR_GetLastError](#)。
说 明: 使用该接口支持上传 V3.0 以上版本支持的设备的报警结构。启动布防前, 需要调用注册回调函数的接口 (如 [NET_DVR_SetDVRMessageCallBack_V30](#)) 才能获取到上传的报警等信息。

[返回目录](#)

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

函 数: BOOL NET_DVR_CloseAlarmChan_V30(LONG IAlarmHandle)
参 数: [in]IAlarmHandle NET_DVR_SetupAlarmChan_V30 的返回值
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

5.17 监听报警

5.17.1 启动监听, 接收设备主动上传的报警等信息

NET_DVR_StartListen_V30

函 数: LONG NET_DVR_StartListen_V30(char *sLocalIP, WORD wLocalPort, MSGCallBack DataCallback, void* pUserData = NULL)
参 数: [in]sLocalIP PC 机本地 IP 地址, 可以置为 NULL
[in]wLocalPort PC 本地监听端口号。由用户设置, 必须和设备端设置的一致
[in]DataCallback 回调函数, 不能为 NULL
[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
[out]pAlarmer 报警设备信息
[out]pAlarmInfo 报警信息
[out]dwBufLen 报警信息缓存大小
[out]pUser 用户数据

ICommand 宏定义	宏定义值	含义
COMM_ALARM	0x1100	V3.0 以下版本支持的设备的报警信息上传
COMM_IPC_AUXALARM_RESULT	0x2820	PIR 报警、无线报警、呼救报警上传

返回值: -1 表示失败, 其他值作为 NET_DVR_CloseAlarmChan_V30 函数的句柄参数。
获取错误码调用 [NET_DVR_GetLastError](#)。

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，其关系见下表：

消息类型（ICommand）	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM	V3.0 以下版本支持的设备的报警信息	NET_DVR_ALARMINFO
COMM_IPC_AUXALARM_RESULT	PIR 报警、无线报警、呼救报警信息	NET_IPC_AUXALARM_RESULT

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

[返回目录](#)

5.17.2 停止监听（支持多线程）NET_DVR_StopListen_V30

函 数： BOOL NET_DVR_StopListen_V30(LONG lListenHandle)
参 数： [in]lListenHandle 监听句柄，NET_DVR_StartListen_V30 的返回值
返回值： TRUE 表示成功，FALSE 表示失败。获取错误码调用 NET_DVR_GetLastError。
说 明：

[返回目录](#)

5.18 云台控制

云台控制操作

5.18.1 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_V30 的返回值
[in]dwPTZCommand 云台控制命令
[in]dwStop 云台停止动作或开始动作：0- 开始；1- 停止

dwPTZCommand 宏定义	宏定义值	含义
LIGHT_PWRON	2	接通灯光电源
WIPER_PWRON	3	接通雨刷开关

FAN_PWRON	4	接通风扇开关
HEATER_PWRON	5	接通加热器开关
AUX_PWRON1	6	接通辅助设备开关
AUX_PWRON2	7	接通辅助设备开关
ZOOM_IN	11	焦距变大(倍率变大)
ZOOM_OUT	12	焦距变小(倍率变小)
FOCUS_NEAR	13	焦点前调
FOCUS_FAR	14	焦点后调
IRIS_OPEN	15	光圈扩大
IRIS_CLOSE	16	光圈缩小
TILT_UP	21	云台上仰
TILT_DOWN	22	云台下俯
PAN_LEFT	23	云台左转
PAN_RIGHT	24	云台右转
UP_LEFT	25	云台上仰和左转
UP_RIGHT	26	云台上仰和右转
DOWN_LEFT	27	云台下俯和左转
DOWN_RIGHT	28	云台下俯和右转
PAN_AUTO	29	云台左右自动扫描

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

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

[返回目录](#)

5.18.2 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_V30 的返回值
[in]IChannel 通道号
[in]dwPTZCommand 云台控制命令
[in]dwStop 云台停止动作或开始动作：0- 开始；1- 停止

dwPTZCommand 宏定义	宏定义值	含义
LIGHT_PWRON	2	接通灯光电源
WIPER_PWRON	3	接通雨刷开关
FAN_PWRON	4	接通风扇开关
HEATER_PWRON	5	接通加热器开关
AUX_PWRON1	6	接通辅助设备开关
AUX_PWRON2	7	接通辅助设备开关
ZOOM_IN	11	焦距变大(倍率变大)
ZOOM_OUT	12	焦距变小(倍率变小)
FOCUS_NEAR	13	焦点前调
FOCUS_FAR	14	焦点后调
IRIS_OPEN	15	光圈扩大
IRIS_CLOSE	16	光圈缩小
TILT_UP	21	云台上仰
TILT_DOWN	22	云台下俯
PAN_LEFT	23	云台左转
PAN_RIGHT	24	云台右转
UP_LEFT	25	云台上仰和左转
UP_RIGHT	26	云台上仰和右转
DOWN_LEFT	27	云台下俯和左转
DOWN_RIGHT	28	云台下俯和右转
PAN_AUTO	29	云台左右自动扫描

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

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

[返回目录](#)

5.18.3 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 lRealHandle, DWORD

dwPTZCommand, DWORD dwStop, DWORD dwSpeed)

参 数: [in]IRealHandle NET_DVR_RealPlay_V30 的返回值
[in] dwPTZCommand 云台控制命令
[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]

dwPTZCommand 宏定义	宏定义值	含义
LIGHT_PWRON	2	接通灯光电源
WIPER_PWRON	3	接通雨刷开关
FAN_PWRON	4	接通风扇开关
HEATER_PWRON	5	接通加热器开关
AUX_PWRON1	6	接通辅助设备开关
AUX_PWRON2	7	接通辅助设备开关
ZOOM_IN	11	焦距变大(倍率变大)
ZOOM_OUT	12	焦距变小(倍率变小)
FOCUS_NEAR	13	焦点前调
FOCUS_FAR	14	焦点后调
IRIS_OPEN	15	光圈扩大
IRIS_CLOSE	16	光圈缩小
TILT_UP	21	云台上仰
TILT_DOWN	22	云台下俯
PAN_LEFT	23	云台左转
PAN_RIGHT	24	云台右转
UP_LEFT	25	云台上仰和左转
UP_RIGHT	26	云台上仰和右转
DOWN_LEFT	27	云台下俯和左转
DOWN_RIGHT	28	云台下俯和右转
PAN_AUTO	29	云台左右自动扫描

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

[返回目录](#)

5.18.4 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]UserID NET_DVR_Login_V30 的返回值
[in]Channel 通道号
[in]dwPTZCommand 云台控制命令
[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]

dwPTZCommand 宏定义	宏定义值	含义
LIGHT_PWRON	2	接通灯光电源
WIPER_PWRON	3	接通雨刷开关
FAN_PWRON	4	接通风扇开关
HEATER_PWRON	5	接通加热器开关
AUX_PWRON1	6	接通辅助设备开关
AUX_PWRON2	7	接通辅助设备开关
ZOOM_IN	11	焦距变大(倍率变大)
ZOOM_OUT	12	焦距变小(倍率变小)
FOCUS_NEAR	13	焦点前调
FOCUS_FAR	14	焦点后调
IRIS_OPEN	15	光圈扩大
IRIS_CLOSE	16	光圈缩小
TILT_UP	21	云台上仰
TILT_DOWN	22	云台下俯
PAN_LEFT	23	云台左转
PAN_RIGHT	24	云台右转
UP_LEFT	25	云台上仰和左转
UP_RIGHT	26	云台上仰和右转
DOWN_LEFT	27	云台下俯和左转
DOWN_RIGHT	28	云台下俯和右转
PAN_AUTO	29	云台左右自动扫描

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

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

[返回目录](#)

云台预置点操作

5.18.5 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_V30 的返回值
`[in]dwPTZPresetCmd` 操作云台预置点命令，见下表
`[in]dwPresetIndex` 预置点的序号（从 1 开始），最多支持 255 个预置点

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

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

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

[返回目录](#)

5.18.6 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_V30 的返回值
`[in]IChannel` 通道号
`[in]dwPTZPresetCmd` 操作云台预置点命令，见下表
`[in]dwPresetIndex` 预置点的序号（从 1 开始），最多支持 255 个预置点

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

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

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

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

[返回目录](#)

云台巡航操作

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

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

参 数： [in]lRealHandle NET_DVR_RealPlay_V30 的返回值
[in]dwPTZCruiseCmd 操作云台巡航命令，见下表
[in]byCruiseRoute 巡航路径，最多支持 32 条路径（序号从 1 开始）
[in]byCruisePoint 巡航点，最多支持 32 个点（序号从 1 开始）
[in]wInput 不同巡航命令时的值不同，预置点(最大 255)、时间(最大 255)、速度(最大 40)

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

返回值： TRUE 表示成功，FALSE 表示失败。获取错误码调用 NET_DVR_GetLastError。

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

[返回目录](#)

5.18.8 云台巡航操作 NET_DVR_PTZPCruise_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_V30 的返回值
[in]lChannel 通道号
[in]dwPTZCruiseCmd 操作云台巡航命令，见下表
[in]byCruiseRoute 巡航路径，最多支持 32 条路径（序号从 1 开

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

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

返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说明: 与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码, 设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配, 需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持, 则无法用该接口控制。

[返回目录](#)

云台轨迹操作

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

函 数: BOOL NET_DVR_PTZTrack(LONG lRealHandle, DWORD dwPTZTrackCmd)
参 数: [in]lRealHandle NET_DVR_RealPlay_V30 的返回值
[in]dwPTZTrackCmd 操作云台巡航命令, 见下表

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

返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说明: 与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码, 设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配, 需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持, 则无法用该接口控制。

[返回目录](#)

5.18.10 云台轨迹操作 **NET_DVR_PTZTrack_Other**

函 数: BOOL NET_DVR_PTZTrack_Other(LONG lUserID, LONG lChannel, DWORD

dwPTZTrackCmd)
参 数: [in]IUserID NET_DVR_Login_V30 的返回值
[in]IChannel 通道号
[in]dwPTZTrackCmd 操作云台巡航命令, 见下表

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

返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明: 与设备之间的云台各项操作的命令都对应于设备与云台之间的控制码, 设备会根据目前设置的解码器种类和解码器地址向云台发送控制码。如果目前设备上设置的解码器与云台设备的不匹配, 需要重新配置设备的解码器。如果云台设备所需的解码器设备不支持, 则无法用该接口控制。

[返回目录](#)

透明云台控制

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

函 数: BOOL NET_DVR_TransPTZ(LONG IRealHandle,char *pPTZCodeBuf,DWORD dwBufSize)
参 数: [in]IRealHandle NET_DVR_RealPlay_V30 的返回值
[in]pPTZCodeBuf 存放云台控制码缓冲区的指针
[in]dwBufSize 云台控制码的长度
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明: 使用该接口能直接通过设备将云台控制码信息直接传输给云台设备, 而无需配置解码器。

[返回目录](#)

5.18.12 透明云台操作 **NET_DVR_TransPTZ_Other**

函 数: BOOL NET_DVR_TransPTZ(LONG IUserID, LONG IChannel, char *pPTZCodeBuf, DWORD dwBufSize)
参 数: [in]IUserID NET_DVR_Login_V30 的返回值
[in]IChannel 通道号
[in]pPTZCodeBuf 存放云台控制码缓冲区的指针
[in]dwBufSize 云台控制码的长度
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明: 使用该接口能直接通过设备将云台控制码信息直接传输给云台设备, 而无需配置解码器。

[返回目录](#)

云台区域缩放控制

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

函 数： BOOL NET_DVR_PTZSelZoomIn(LONG IRealHandle, LPNET_DVR_POINT_FRAME pStruPointFrame);

参 数： [in]IRealHandle NET_DVR_RealPlay_V30 的返回值
[in]pStruPointFrame 云台图像区域位置信息

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

说 明： 该接口实现 3D 定位功能，需要前端设备的支持。
假设当前预览显示图像的框为 352*288，原点即该显示框的左上角的顶点。
参数 pStruPointFrame 中各坐标值的计算方法（以 X 轴方向上为例）：xTop=鼠标当前所选区域的左上点的值*255/352。缩小条件：xBottom 减去 xTop 的值大于 2。放大条件：xBottom 减去 xTop 的值大于 0，且 yBottom 减去 yTop 的值大于 0。

[返回目录](#)

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

函 数： BOOL NET_DVR_PTZSelZoomIn_EX(LONG IUserID, LONG IChannel, LPNET_DVR_POINT_FRAME pStruPointFrame)

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

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： 该接口实现 3D 定位功能，需要前端设备的支持。
假设当前预览显示图像的框为 352*288，原点即该显示框的左上角的顶点。
参数 pStruPointFrame 中各坐标值的计算方法（以 X 轴方向上为例）：xTop=鼠标当前所选区域的左上点的值*255/352。缩小条件：xBottom 减去 xTop 的值大于 2。放大条件：xBottom 减去 xTop 的值大于 0，且 yBottom 减去 yTop 的值大于 0。

[返回目录](#)

云台定位

5.18.15 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_V30 的返回值

[in]dwCommand 设备配置命令，参见配置命令

[in]lChannel 通道号，如果命令不需要通道号，该参数无效，置为 0xFFFFFFFF 即可

[out]lpOutBuffer 接收数据的缓冲指针

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

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

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

说明： 不同的获取功能对应不同的结构体和命令号，如下表所示：

dwCommand 宏定义	dwCommand 含义	通道号	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_PTZPOS	获取 IP 快球 PTZ 参数	有效	NET_DVR_PTZPOS	293
NET_DVR_GET_PTZSCOPE	获取 IP 快球 PTZ 范围参数	有效	NET_DVR_PTZSCOPE	294

[返回目录](#)

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

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

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

[in]dwCommand 设备配置命令，参见配置命令

[in]lChannel 通道号，如果命令不需要通道号，该参数无效，置为 0xFFFFFFFF 即可

[in]lpInBuffer 输入数据的缓冲指针

[in]dwInBufferSize 输入数据的缓冲长度(以字节为单位)

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

说明： 不同的获取功能对应不同的结构体和命令号，如下表所示：

dwCommand 宏定义	dwCommand 含义	通道号	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_PTZPOS	设置 IP 快球 PTZ 参数	有效	NET_DVR_PTZPOS	292

[返回目录](#)

获取巡航路径

5.18.17 获取 IP 快球云台巡航路径 **NET_DVR_GetPTZCruise**

函 数： BOOL NET_DVR_GetPTZCruise(LONG lUserID, LONG lChannel, LONG lCruiseRoute, LPNET_DVR_CRUISE_RET lpCruiseRet)

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

[in]lChannel 通道号

[in]lCruiseRoute 巡航路径号

[out]dwInBufferSize 巡航轨迹

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

说 明:

[返回目录](#)

5.19 IPC 远程控制

5.19.1控制一键聚焦 **NET_DVR_FocusOnePush**

函 数: BOOL NET_DVR_FocusOnePush(LONG IUserID, LONG IChannel)
参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V30 的返回值
 [in]IChannel 通道号
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

5.19.2恢复镜头电机默认位置 **NET_DVR_ResetLens**

函 数: BOOL NET_DVR_ResetLens(LONG IUserID, LONG IChannel)
参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V30 的返回值
 [in]IChannel 通道号
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

5.19.3遥控器控制 **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_V30 的返回值
 [in]dwCommand 控制命令, 详见列表
 [in]lpInBuffer 输入参数, 具体内容跟控制命令相关
 [in]dwInBufferSize 输入参数长度
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明: 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:

dwCommand 宏定义	宏值	控制功能	lpInBuffer 对应结构体
NET_DVR_REMOTECONTROL_ALARM	3205	远程控制遥控器布防	NET_DVR_REMOTECONTROL_ALARM_PARAM
NET_DVR_REMOTECONTROL_DISALARM	3206	远程控制遥控器撤防	NET_DVR_REMOTECONTROL_ALARM_PARAM
NET_DVR_REMOTECONTROL_STUDY	3207	远程控制遥控器学习	NET_DVR_REMOTECONTROL_STUDY_

			PARAM
NET_DVR_WIRELESS_ALARM_STUDY	3208	远程控制无线报警学习	NET_DVR_WIRELESS_ALARM_STUDY_PARAM

[返回目录](#)

5.20 语音对讲、转发及广播

语音对讲

5.20.1 启动语音对讲 **NET_DVR_StartVoiceCom_V30**

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

参 数：

[in]IUserID	NET_DVR_Login_V30 的返回值
[in]dwVoiceChan	语音通道号，从 1 开始
[in]bNeedCBNoEncData	需要回调的语音数据类型： 0—编码后的语音数据， 1—编码前的 PCM 原始数据
[in]cbVoiceDataCallBack	音频数据回调函数
[in]pUser	用户数据指针

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

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

返回值： -1 表示失败，其他值作为 NET_DVR_StopVoiceCom 等函数的句柄参数。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明： Windows 7 操作系统下，如果不外接音频设备，该接口将返回失败。在调用开始语音对讲之前可先配置设备的语音对讲音频编码类型，即可先调用参数配置中的 NET_DVR_COMPRESSION_AUDIO 结构配置。
当前音频为 OggVorbis 编码时，音频数据的采样频率为 16000，16 位采样且是单通道的。因此，音频播放格式应如下定义：

```
const int SAMPLES_PER_SECOND = 16000;
const int CHANNEL = 1;
const int BITS_PER_SAMPLE = 16;
WAVEFORMATEX m_wavFormatEx;
```

```

m_wavFormatEx.cbSize = sizeof(m_wavFormatEx);
m_wavFormatEx.nBlockAlign = CHANNEL * BITS_PER_SAMPLE / 8;
m_wavFormatEx.nChannels = CHANNEL;
m_wavFormatEx.nSamplesPerSec = SAMPLES_PER_SECOND;
m_wavFormatEx.wBitsPerSample = BITS_PER_SAMPLE;
m_wavFormatEx.nAvgBytesPerSec =
SAMPLES_PER_SECOND*m_wavFormatEx.nBlockAlign

```

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

```

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

```

[返回目录](#)

5.20.2 设置语音对讲客户端的音量

NET_DVR_SetVoiceComClientVolume

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

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

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

说 明：

[返回目录](#)

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

函 数： BOOL NET_DVR_StopVoiceCom(LONG lVoiceComHandle)

参 数： [in]lVoiceComHandle NET_DVR_StartVoiceCom_V30 或
NET_DVR_StartVoiceCom_MR_V30 的返回值

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

说 明：

[返回目录](#)

语音转发

5.20.4 启动语音转发，获取编码后的音频数据

NET_DVR_StartVoiceCom_MR_V30

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

参 数：

[in]IUserID	NET_DVR_Login_V30 的返回值
[in]dwVoiceChan	语音通道号，从 1 开始
[in]cbVoiceDataCallBack	音频数据回调函数，得到的数据是编码以后的音频数据，需调用我们提供的音频解码函数（详见音频编解码章节的说明）后可得到 PCM 数据
[in]pUser	用户数据指针

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

[out]IVoiceComHandle	NET_DVR_StartVoiceCom_MR_V30 的返回值
[out]pRecvDataBuffer	存放音频数据的缓冲区指针
[out]dwBufSize	音频数据大小
[out]byAudioFlag	音频数据类型：1-设备发送过来的音频数据
[out]pUser	用户数据指针

返回值： -1 表示失败，其他值作为 NET_DVR_VoiceComSendData、NET_DVR_StopVoiceCom 等函数的句柄参数。获取错误码调用 [NET_DVR_GetLastError](#)。

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

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

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

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

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

[返回目录](#)

5.20.5 转发语音数据 **NET_DVR_VoiceComSendData**

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

参 数: [in]lVoiceComHandle NET_DVR_StartVoiceCom_MR_V30 的返回值
[in]pSendBuf 存放语音数据的缓冲区
[in]dwBufSize 语音数据大小。当前是 OggVorbis 音频编码类型时，每次发送的数据为 80 字节；当前是 G711 音频编码类型时，每次发送的数据为 160 字节。

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

说 明: 该接口实现将获取到的经过编码后的音频数据转发给设备。

[返回目录](#)

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

函 数: BOOL NET_DVR_StopVoiceCom (LONG lVoiceComHandle)

参 数: [in]lVoiceComHandle NET_DVR_StartVoiceCom_V30 或
NET_DVR_StartVoiceCom_MR_V30 的返回值

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

说 明:

[返回目录](#)

语音广播

5.20.7 启动语音广播的 PC 端声音捕获

NET_DVR_ClientAudioStart_V30

函 数: `BOOL NET_DVR_ClientAudioStart_V30(fVoiceDataCallBack cbVoiceDataCallBack, void *pUser)`

参 数:

<code>[in]cbVoiceDataCallBack</code>	音频数据回调函数
<code>[in]pUser</code>	用户数据

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

<code>[out]pRecvDataBuffer</code>	存放 PC 本地采集的音频数据（PCM）的缓冲区指针
<code>[out]dwBufSize</code>	音频数据大小
<code>[out]pUser</code>	用户数据指针

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

[返回目录](#)

5.20.8 添加设备的某个语音通道到可以接收 PC 端声音的广播组

NET_DVR_AddDVR_V30

函 数: `LONG NET_DVR_AddDVR_V30(LONG IUserID, DWORD dwVoiceChan)`

参 数:

<code>[in]IUserID</code>	<code>NET_DVR_Login_V30</code> 的返回值
<code>[in]dwVoiceChan</code>	语音通道号, 从 1 开始

返回值: -1 表示失败, 其他值作为 `NET_DVR_DelDVR_V30` 等函数的参数。获取错误码调用 [NET_DVR_GetLastError](#)。

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

[返回目录](#)

5.20.9 从可接收 PC 机声音的广播组里删除该设备的语音通道

NET_DVR_DeIDVR_V30

函 数: LONG NET_DVR_DeIDVR_V30(LONG IUserID)
参 数: [in]IUserID NET_DVR_Login_V30 的返回值
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

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

函 数: BOOL NET_DVR_ClientAudioStop()
参 数:
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

音频编解码

OggVorbis 音频编解码

5.20.11 初始化音频编码 NET_DVR_InitG722Encoder

函 数: void* NET_DVR_InitG722Encoder()
参 数:
返回值: -1 表示失败,其他值为音频编码句柄。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

5.20.12 OggVorbis 音频编码 NET_DVR_EncodeG722Frame

函 数: BOOL NET_DVR_EncodeG722Frame(void *pEncodeHandle,unsigned char* pInBuffer, unsigned char* pOutBuffer)
参 数: [in]pEncodeHandle 音频编码句柄, NET_DVR_InitG722Encoder 的返回值
[in]pInBuffer 输入缓冲区, 按采样标准(采样频率为 16000, 16 位采样, 单通道)获取的 PCM 音频数据, 规定输入数据的大小为 1280 字节
[out]pOutBuffer 输出缓冲区, 编码后的输出数据大小为 80

字节

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

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

[返回目录](#)

5.20.13 释放音频编码资源 NET_DVR_ReleaseG722Encoder

函数: void NET_DVR_ReleaseG722Encoder(void *pEncodeHandle)

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

返回值: 无。获取错误码调用 [NET_DVR_GetLastError](#)。

说明:

[返回目录](#)

5.20.14 初始化音频解码 NET_DVR_InitG722Decoder

函数: void* NET_DVR_InitG722Decoder(int nBitrate = 16000)

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

返回值: -1 表示失败, 其他值为音频解码句柄。获取错误码调用 [NET_DVR_GetLastError](#)。

说明:

[返回目录](#)

5.20.15 OggVorbis 音频解码 NET_DVR_DecodeG722Frame

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

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

[in]pInBuffer 输入缓冲区, 编码数据大小为 80 字节

[out]pOutBuffer 输出缓冲区, 按采样标准(采样频率为 16000, 16 位采样, 单通道)获取的 PCM 音频数据, 规定输出数据的大小为 1280 字节

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

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

在结束调用后还需要做释放资源的操作。

[返回目录](#)

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

函 数: void NET_DVR_ReleaseG722Decoder(void *pDecHandle)

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

返回值: 无返回值。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明:

[返回目录](#)

G711 音频编解码

5.20.17 G711 音频编码 **NET_DVR_EncodeG711Frame**

函 数: BOOL NET_DVR_EncodeG711Frame(unsigned int iType, unsigned char
*pInBuffer, unsigned char *pOutBuffer)

参 数: [in]iType 编码类型: 0-Mu law 编码, 非 0-A law 编码
[in]pInBuffer 输入缓冲区, 按采样标准(采样频率为 8000,
16 位采样, 单通道)获取的 PCM 音频数据,
规定输入数据的大小为 320 字节
[out]pOutBuffer 输出缓冲区, 编码后输出数据大小为 160 字
节

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

说 明: 主要为配合语音对讲功能而设定, 当需将客户端的原始音频数据发送至设备
端, 可采用音频编码函数将原始数据压缩编码后再发往设备端; 客户端获取
设备端发送过来的压缩码流, 可调用音频解码函数
NET_DVR_DecodeG711Frame 进行数据解码。在调用编解码函数之前无需做初
始化操作。

[返回目录](#)

5.20.18 G711 音频解码 **NET_DVR_DecodeG711Frame**

函 数: BOOL NET_DVR_DecodeG711Frame(unsigned int iType, unsigned char
*pInBuffer, unsigned char *pOutBuffer)

参 数: [in]iType 编码类型: 0-Mu law 编码, 非 0-A law 编码
[in]pInBuffer 输入缓冲区, 编码数据大小为 160 字节
[out]pOutBuffer 输出缓冲区, 按采样标准(采样频率为 8000,
16 位采样, 单通道)获取的 PCM 音频数据,
规定输出数据的大小为 320 字节

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

说明: 主要为配合语音对讲、转发功能而设定, 当需将客户端的原始音频数据发送至设备端, 可采用音频编码函数 NET_DVR_EncodeG711Frame 将原始数据压缩编码后再发往设备端; 客户端获取设备端发送过来的压缩码流, 可调用音频解码函数进行数据解码。在调用编解码函数之前无需做初始化操作。

[返回目录](#)

G726 音频编解码

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

函数: void* NET_DVR_InitG726Encoder(void **pEncMoudle)

参数: [out]pEncMoudle 编码模块句柄, 编码时作为输入参数

返回值: -1 表示失败, 其他值为音频编码句柄。获取错误码调用 [NET_DVR_GetLastError](#)。

说明:

[返回目录](#)

5.20.20 G722 音频编码 **NET_DVR_EncodeG726Frame**

函数: BOOL NET_DVR_EncodeG726Frame(void *pEncMoudle, unsigned char *pInBuffer, unsigned char *pOutBuffer, BYTE byReset)

参数: [in]pEncMoudle 音频编码句柄, NET_DVR_InitG726Encoder 的输出参数

[in]pInBuffer 输入缓冲区, 按采样标准(采样频率为 8000, 16 位采样, 单通道)获取的 PCM 音频数据, 规定输入数据的大小为 640 字节

[out]pOutBuffer 输出缓冲区, 编码后输出数据大小为 80 字节

[in]byReset 是否重置, 第一帧需要重置

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

说明: 主要为配合语音对讲、转发功能而设定, 当需将客户端的原始音频数据发送至设备端, 可采用音频编码函数将原始数据压缩编码后再发往设备端; 客户端获取设备端发送过来的压缩码流, 可调用音频解码函数 NET_DVR_DecodeG726Frame 进行数据解码。在调用编解码函数之前都需要做相应的初始化操作, 在结束调用后还需要做释放资源的操作。

[返回目录](#)

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

函数: void NET_DVR_ReleaseG726Encoder(void *pEncHandle)

参数: [in]pEncHandle 音频编码句柄, NET_DVR_InitG726Encoder 的返回值

返回值： 无。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明：

[返回目录](#)

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

函 数： void* NET_DVR_InitG726Decoder(void **pDecMoudle)

参 数： [out]pDecMoudle 解码模块句柄，解码时作为输入参数

返回值： -1 表示失败，其他值为音频解码句柄。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明：

[返回目录](#)

5.20.23 G722 音频解码 **NET_DVR_DecodeG726Frame**

函 数： BOOL NET_DVR_DecodeG722Frame(void *pDecMoudle, unsigned char *pInBuffer, unsigned char *pOutBuffer, BYTE byReset)

参 数： [in]pDecMoudle 音频解码句柄，NET_DVR_InitG726Decoder 的输出参数

[in]pInBuffer 输入缓冲区，编码数据大小为 80 字节

[out]pOutBuffer 输出缓冲区，按采样标准（采样频率为 8000，16 位采样，单通道）获取的 PCM 音频数据，规定输出数据的大小为 640 字节

[in]byReset 是否重置，第一帧需要重置

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

说 明： 主要为配合语音对讲、转发功能而设定，当需将客户端的原始音频数据发送至设备端，可采用音频编码函数 NET_DVR_EncodeG726Frame 将原始数据压缩编码后再发往设备端；客户端获取设备端发送过来的压缩码流，可调用音频解码函数进行数据解码。在调用编解码函数之前都需要做相应的初始化操作，在结束调用后还需要做释放资源的操作。

[返回目录](#)

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

函 数： void NET_DVR_ReleaseG726Decoder(void *pDecHandle)

参 数： [in]pDecHandle 音频解码句柄，NET_DVR_InitG726Decoder 的返回值

返回值： 无返回值。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明：

[返回目录](#)

5.21 透明通道

5.21.1 建立透明通道 **NET_DVR_SerialStart**

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

参 数: [in]IUserID NET_DVR_Login_V30 的返回值
[in]ISerialPort 串口号: 1- 232 串口; 2- 485 串口
[in]cbSerialDataCallBack 透明通道数据回调函数
[in]dwUser 用户数据

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

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

返回值: -1 表示失败, 其他值作为 NET_DVR_SerialSend 等函数的句柄参数。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明: 需要从回调函数得到数据解码器必须支持数据回传, 否则发送成功, 回调依然不会有返回。

[返回目录](#)

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

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

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

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

说 明:

[返回目录](#)

5.21.3 断开透明通道 **NET_DVR_SerialStop**

函 数: BOOL NET_DVR_SerialStop (LONG ISerialHandle)

参 数: [in]ISerialHandle NET_DVR_SerialStart 的返回值

返回值： TRUE 表示成功，FALSE 表示失败。

说 明：

[返回目录](#)

5.22 向串口发送数据

5.22.1 直接向串口发送数据，不需要建立透明通道

NET_DVR_SendToSerialPort

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

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

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

说 明：

[返回目录](#)

5.22.2 直接向 232 串口发送数据，不需要建立透明通道

NET_DVR_SendTo232Port

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

参 数： [in]lUserID NET_DVR_Login_V30 的返回值
[in]pSendBuf 发送数据的缓冲区指针
[in]dwBufSize 缓冲区的大小，最多 1016 字节

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

说 明：

[返回目录](#)

5.23 硬盘管理

5.23.1 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_V30 的返回值
 [in]IDiskNumber 硬盘号, 从 0 开始, 0xff 表示对所有硬盘有效 (不包括只读硬盘)

返回值: -1 表示失败, 其他值作为 NET_DVR_CloseFormatHandle 等函数的参数。获取错误码调用 [NET_DVR_GetLastError](#)。

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](#)

5.23.2 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。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明:

[返回目录](#)

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

函 数: BOOL NET_DVR_CloseFormatHandle(LONG IFormatHandle)

参 数: [in]IFormatHandle 格式化硬盘句柄, NET_DVR_FormatDisk 的返

返回值

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

说明:

[返回目录](#)

5.24 设备维护管理

获取设备工作状态

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

函数: BOOL NET_DVR_GetDVRWorkState_V30(LONG lUserID,
LPNET_DVR_WORKSTATE_V30 lpWorkState)

参数: [in]lUserID NET_DVR_Login_V30 的返回值
[out]lpWorkState 获取的设备工作状态结构体参数

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

说明: 此处获取的是 IPC 的硬盘状态, 其他设备的硬盘状态通过
NET_DVR_GetDVRConfig 的 NET_DVR_GET_HDCFG 命令获取。

[返回目录](#)

远程升级

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

函数: BOOL NET_DVR_SetNetworkEnvironment(DWORD dwEnvironmentLevel)

参数: [in]dwEnvironmentLevel 网络环境级别

enum{

LOCAL_AREA_NETWORK = 0, //局域网环境

WIDE_AREA_NETWORK //广域网环境

}

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

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](#)

5.24.3 远程升级 **NET_DVR_Upgrade**

函 数: LONG NET_DVR_Upgrade(LONG IUserID, char *sFileName)
参 数: [in]IUserID NET_DVR_Login_V30 的返回值
[in]sFileName 升级的文件路径（包括文件名）。路径长度和操作系统有关，sdk 不做限制，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。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明:

[返回目录](#)

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

函 数: Int NET_DVR_GetUpgradeProgress(LONG IUpgradeHandle)
参 数: [in]IUpgradeHandle NET_DVR_Upgrade 的返回值
返回值: -1 表示失败，0~100 表示升级进度。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明:

[返回目录](#)

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

函 数: Int NET_DVR_GetUpgradeState(LONG IUpgradeHandle)
参 数: [in]IUpgradeHandle NET_DVR_Upgrade()的返回值
返回值: -1 表示失败，其他值定义如下：
1—升级成功
2—正在升级
3—升级失败
4—网络断开，状态未知
5—升级文件语言版本不匹配
获取错误码调用 [NET_DVR_GetLastError](#)。

说 明:

[返回目录](#)

5.24.6 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，其他值定义如下：

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

获取错误码调用 [NET_DVR_GetLastError](#)。

说 明：

[返回目录](#)

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

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

[返回目录](#)

日志查找

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

NET_DVR_FindDVRLog_V30

函 数： LONG NET_DVR_FindDVRLog_V30(LONG lUserID, LONG lSelectMode, DWORD dwMajorType, DWORD dwMinorType, LPNET_DVR_TIME lpStartTime, LPNET_DVR_TIME lpStopTime, BOOL bOnlySmart = FALSE)
参 数： [in]lUserID NET_DVR_Login_V30 的返回值
[in]lSelectMode 查询方式：
0—全部；
1—按类型；
2—按时间；
3—按时间和类型
[in]dwMajorType 日志主类型（S.M.A.R.T 搜索时无效），0 表示全部类型，其他类型定义如下：
[in]dwMinorType 日志次类型（S.M.A.R.T 搜索时无效），0 表示全部类型，根据不同的主类型的次类型定义如下
[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。获取错

误码调用 [NET_DVR_GetLastError](#)。

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。

dwMajorType

日志主类型（S.M.A.R.T 搜索时无效），0 表示全部类型，其他类型定义如下：

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

dwMinorType

日志次类型（S.M.A.R.T 搜索时无效），0 表示全部类型，根据不同的主类型的次类型定义如下：

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

主类型的宏定义	宏定义值	含义
MAJOR_EXCEPTION	0x2	异常
次类型的宏定义	宏定义值	含义
MINOR_RAID_ERROR	0x20	阵列异常

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

主类型的宏定义	宏定义值	含义
MAJOR_OPERATION	0x3	操作
次类型的宏定义	宏定义值	含义
MINOR_START_DVR	0x41	开机
MINOR_STOP_DVR	0x42	关机
MINOR_STOP_ABNORMAL	0x43	异常关机
MINOR_REBOOT_DVR	0x44	本地重启设备
MINOR_LOCAL_LOGIN	0x50	本地登陆
MINOR_LOCAL_LOGOUT	0x51	本地注销登陆
MINOR_LOCAL_CFG_PARM	0x52	本地配置参数
MINOR_LOCAL_PLAYBYFILE	0x53	本地按文件回放或下载
MINOR_LOCAL_PLAYBYTIME	0x54	本地按时间回放或下载
MINOR_LOCAL_START_REC	0x55	本地开始录像
MINOR_LOCAL_STOP_REC	0x56	本地停止录像
MINOR_LOCAL_PTZCTRL	0x57	本地云台控制
MINOR_LOCAL_PREVIEW	0x58	本地预览(保留不使用)
MINOR_LOCAL_MODIFY_TIME	0x59	本地修改时间(保留不使用)
MINOR_LOCAL_UPGRADE	0x5a	本地升级
MINOR_LOCAL_RECFILE_OUTPUT	0x5b	本地备份录像文件
MINOR_LOCAL_FORMAT_HDD	0x5c	本地初始化硬盘

MINOR_LOCAL_CFGFILE_OUTPUT	0x5d	导出本地配置文件
MINOR_LOCAL_CFGFILE_INPUT	0x5e	导入本地配置文件
MINOR_LOCAL_COPYFILE	0x5f	本地备份文件
MINOR_LOCAL_LOCKFILE	0x60	本地锁定录像文件
MINOR_LOCAL_UNLOCKFILE	0x61	本地解锁录像文件
MINOR_LOCAL_DVR_ALARM	0x62	本地手动清除和触发报警
MINOR_IPC_ADD	0x63	本地添加 IPC
MINOR_IPC_DEL	0x64	本地删除 IPC
MINOR_IPC_SET	0x65	本地设置 IPC
MINOR_LOCAL_START_BACKUP	0x66	本地开始备份
MINOR_LOCAL_STOP_BACKUP	0x67	本地停止备份
MINOR_LOCAL_COPYFILE_START_TIME	0x68	本地备份开始时间
MINOR_LOCAL_COPYFILE_END_TIME	0x69	本地备份结束时间
MINOR_LOCAL_ADD_NAS	0x6a	本地添加网络硬盘
MINOR_LOCAL_DEL_NAS	0x6b	本地删除 NAS 盘
MINOR_LOCAL_SET_NAS	0x6c	本地设置 NAS 盘
MINOR_REMOTE_LOGIN	0x70	远程登录
MINOR_REMOTE_LOGOUT	0x71	远程注销登陆
MINOR_REMOTE_START_REC	0x72	远程开始录像
MINOR_REMOTE_STOP_REC	0x73	远程停止录像
MINOR_START_TRANS_CHAN	0x74	开始透明传输
MINOR_STOP_TRANS_CHAN	0x75	停止透明传输
MINOR_REMOTE_GET_PARM	0x76	远程获取参数
MINOR_REMOTE_CFG_PARM	0x77	远程配置参数
MINOR_REMOTE_GET_STATUS	0x78	远程获取状态
MINOR_REMOTE_ARM	0x79	远程布防
MINOR_REMOTE_DISARM	0x7a	远程撤防
MINOR_REMOTE_REBOOT	0x7b	远程重启
MINOR_START_VT	0x7c	开始语音对讲
MINOR_STOP_VT	0x7d	停止语音对讲
MINOR_REMOTE_UPGRADE	0x7e	远程升级
MINOR_REMOTE_PLAYBYFILE	0x7f	远程按文件回放
MINOR_REMOTE_PLAYBYTIME	0x80	远程按时间回放
MINOR_REMOTE_PTZCTRL	0x81	远程云台控制
MINOR_REMOTE_FORMAT_HDD	0x82	远程格式化硬盘

MINOR_REMOTE_STOP	0x83	远程关机
MINOR_REMOTE_LOCKFILE	0x84	远程锁定文件
MINOR_REMOTE_UNLOCKFILE	0x85	远程解锁文件
MINOR_REMOTE_CFGFILE_OUTPUT	0x86	远程导出配置文件
MINOR_REMOTE_CFGFILE_INTPUT	0x87	远程导入配置文件
MINOR_REMOTE_RECFILE_OUTPUT	0x88	远程导出录像文件
MINOR_REMOTE_DVR_ALARM	0x89	远程手动清除和触发报警
MINOR_REMOTE_IPC_ADD	0x8a	远程添加 IPC
MINOR_REMOTE_IPC_DEL	0x8b	远程删除 IPC
MINOR_REMOTE_IPC_SET	0x8c	远程设置 IPC
MINOR_REBOOT_VCA_LIB	0x8d	重启智能库
MINOR_REMOTE_ADD_NAS	0x8e	远程添加 NAS 盘
MINOR_REMOTE_DEL_NAS	0x8f	远程删除 NAS 盘
MINOR_REMOTE_SET_NAS	0x90	远程设置 NAS 盘
MINOR_LOCAL_START_REC_CDRW	0x91	本地开始刻录
MINOR_LOCAL_STOP_REC_CDRW	0x92	本地停止刻录
MINOR_REMOTE_START_REC_CDRW	0x93	远程开始刻录
MINOR_REMOTE_STOP_REC_CDRW	0x94	远程停止刻录
MINOR_LOCAL_PIC_OUTPUT	0x95	本地备份图片文件
MINOR_REMOTE_SET_NAS	0x96	远程备份图片文件
MINOR_LOCAL_INQUEST_RESUME	0x97	本地恢复审讯事件
MINOR_REMOTE_INQUEST_RESUME	0x98	远程恢复审讯事件
MINOR_REMOTE_BYPASS	0xd0	远程旁路
MINOR_REMOTE_UNBYPASS	0xd1	远程旁路恢复
MINOR_REMOTE_SET_ALARMIN_CFG	0xd2	远程设置报警输入参数
MINOR_REMOTE_GET_ALARMIN_CFG	0xd3	远程获取报警输入参数
MINOR_REMOTE_SET_ALARMOUT_CFG	0xd4	远程设置报警输出参数
MINOR_REMOTE_GET_ALARMOUT_CFG	0xd5	远程获取报警输出参数
MINOR_REMOTE_ALARMOUT_OPEN_MAN	0xd6	远程手动开启报警输出
MINOR_REMOTE_ALARMOUT_CLOSE_MAN	0xd7	远程手动关闭报警输出
MINOR_REMOTE_ALARM_ENABLE_CFG	0xd8	远程设置报警主机的 RS485 串口使能状态
MINOR_DBDATA_OUTPUT	0xd9	导出数据库记录
MINOR_DBDATA_INPUT	0xda	导入数据库记录
MINOR_MU_SWITCH	0xdb	级联切换

MINOR_MU_PTZ	0xdc	级联 PTZ 控制
MINOR_LOCAL_CONF_REB_RAID	0x101	本地配置自动重建
MINOR_LOCAL_CONF_SPARE	0x102	本地配置热备
MINOR_LOCAL_ADD_RAID	0x103	本地创建阵列
MINOR_LOCAL_DEL_RAID	0x104	本地删除阵列
MINOR_LOCAL_MIG_RAID	0x105	本地迁移阵列
MINOR_LOCAL_REB_RAID	0x106	本地手动重建阵列
MINOR_LOCAL_QUICK_CONF_RAID	0x107	本地一键配置
MINOR_LOCAL_ADD_VD	0x108	本地创建虚拟磁盘
MINOR_LOCAL_DEL_VD	0x109	本地删除虚拟磁盘
MINOR_LOCAL_RP_VD	0x10a	本地修复虚拟磁盘
MINOR_LOCAL_FORMAT_EXPANDVD	0x10b	本地扩展虚拟磁盘扩容
MINOR_LOCAL_RAID_UPGRADE	0x10c	本地 raid 卡升级
MINOR_LOCAL_STOP_RAID	0x10d	本地暂停 RAID 操作(即安全拔盘)
MINOR_REMOTE_CONF_REB_RAID	0x111	远程配置自动重建
MINOR_REMOTE_CONF_SPARE	0x112	远程配置热备
MINOR_REMOTE_ADD_RAID	0x113	远程创建阵列
MINOR_REMOTE_DEL_RAID	0x114	远程删除阵列
MINOR_REMOTE_MIG_RAID	0x115	远程迁移阵列
MINOR_REMOTE_REB_RAID	0x116	远程手动重建阵列
MINOR_REMOTE_QUICK_CONF_RAID	0x117	远程一键配置
MINOR_REMOTE_ADD_VD	0x118	远程创建虚拟磁盘
MINOR_REMOTE_DEL_VD	0x119	远程删除虚拟磁盘
MINOR_REMOTE_RP_VD	0x11a	远程修复虚拟磁盘
MINOR_REMOTE_FORMAT_EXPANDVD	0x11b	远程虚拟磁盘扩容
MINOR_REMOTE_RAID_UPGRADE	0x11c	远程 raid 卡升级
MINOR_REMOTE_STOP_RAID	0x11d	远程暂停 RAID 操作(即安全拔盘)
MINOR_LOCAL_START_PIC_REC	0x121	本地开始抓图
MINOR_LOCAL_STOP_PIC_REC	0x122	本地停止抓图
MINOR_LOCAL_SET_SNMP	0x125	本地配置 SNMP
MINOR_LOCAL_TAG_OPT	0x126	本地标签操作
MINOR_REMOTE_START_PIC_REC	0x131	远程开始抓图
MINOR_REMOTE_STOP_PIC_REC	0x132	远程停止抓图
MINOR_REMOTE_SET_SNMP	0x135	远程配置 SNMP
MINOR_REMOTE_TAG_OPT	0x136	远程标签操作

MINOR_REMOTE_TAG_OPT	0x136	远程标签操作
MINOR_LOCAL_VOUT_SWITCH	0x140	本地输出口切换操作

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

[返回目录](#)

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

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

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

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

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](#)

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

函 数： BOOL NET_DVR_FindLogClose_V30(LONG lLogHandle)

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

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

说 明：

[返回目录](#)

远程备份

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

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

参 数： [in]lUserID NET_DVR_Login_V30 的返回值
[out]lpDiskList 设备可用备份磁盘信息结构

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

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

[返回目录](#)

5.24.12 备份统一接口 **NET_DVR_Backup**

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

参 数： [in]lUserID 用户 ID 号，NET_DVR_Login_V30 的返回值
[in]dwBackupType 备份类型：1- 按文件名备份录像文件，2- 按时间段备份录像文件，3- 备份图片，4- 恢复审讯事件，5- 备份日志
[in]lpBackupBuff 指向备份参数指针
[in]dwBackupBuffSize 备份参数大小

返回值： -1 失败，0-510 的返回句柄值作为 NET_DVR_GetBackupProgress，NET_DVR_StopBackup 的参数。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明： 不同的备份类型(dwBackupType)对应不同的备份参数结构体(lpBackupBuff)：

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

[返回目录](#)

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

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

参 数: [in]IHandle NET_DVR_Backup 的返回值
[out] pState 当前备份的进度，进度值的取值范围为 (0,100)，其他值的定义见下表

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

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

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

[返回目录](#)

5.24.14 停止备份 **NET_DVR_StopBackup**

函 数: BOOL NET_DVR_StopBackup(LONG IHandle)

参 数: [in]IHandle NET_DVR_Backup 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](#)。

[返回目录](#)

恢复设备默认参数

5.24.15 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_V30 的返回值

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

说 明:

[返回目录](#)

导入/导出配置文件

5.24.16 导出配置文件 **NET_DVR_GetConfigFile_V30**

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

参 数： [in]IUserID 用户 ID 号，NET_DVR_Login_V30 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。获取错误码调用 [NET_DVR_GetLastError](#)。

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](#)

5.24.17 导出配置文件 **NET_DVR_GetConfigFile**

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

参 数： [in]IUserID 用户 ID 号，NET_DVR_Login_V30 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。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明：

[返回目录](#)

5.24.18 导入配置文件 **NET_DVR_SetConfigFile_EX**

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

参 数： [in]IUserID 用户 ID 号，NET_DVR_Login_V30 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。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明：

[返回目录](#)

5.24.19 导入配置文件 **NET_DVR_SetConfigFile**

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

参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V30 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。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

5.25 关机和重启

5.25.1 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_V30 的返回值
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

5.25.2 关闭设备 **NET_DVR_ShutDownDVR**

函 数: BOOL NET_DVR_ShutDownDVR(LONG IUserID)
参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V30 的返回值
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

5.26 远程参数配置

通用参数配置

5.26.1 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_V30 的返回值
[in]dwCommand 设备配置命令, 参见配置命令
[in]IChannel 通道号, 如果命令不需要通道号, 该参数无效, 置为 0xFFFFFFFF 即可
[out]lpOutBuffer 接收数据的缓冲指针

[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。获取错误码调用 [NET_DVR_GetLastError](#)。

说明: 不同的获取功能对应不同的结构体和命令号, 如下表所示:

■ 对于配置命令 NET_DVR_GET_ALARMINGCFG_V30 和 NET_DVR_GET_ALARMOUTCFG_V30, IChannel 为 0, 建议应用界面上通道号显示为 1 以保持和设备本地日志的一致。

■ 其他配置命令, 如果 IChannel 有效, 通道号为 1。

dwCommand 宏定义	dwCommand 含义	通道号	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_DEVICECFG_V40	获取设备参数(扩展)	无效	NET_DVR_DEVICECFG_V40	1100
NET_DVR_GET_TIMECFG	获取时间参数	无效	NET_DVR_TIME	118
NET_DVR_GET_NETCFG_V30	获取网络参数	无效	NET_DVR_NETCFG_V30	1000
NET_DVR_GET_PICCFG_V30	获取图像参数	有效	NET_DVR_PICCFG_V30	1002
NET_DVR_GET_COMPRESSCFG_V30	获取压缩参数	有效	NET_DVR_COMPRESSIONCFG_V30	1040
NET_DVR_GET_RECORDCFG_V30	获取录像参数	有效	NET_DVR_RECORD_V30	1004
NET_DVR_GET_DECODERCFG_V30	获取(云台)解码器参数	有效	NET_DVR_DECODERCFG_V30	1042
NET_DVR_GET_RS232CFG_V30	获取 232 串口参数	无效	NET_DVR_RS232CFG_V30	1036
NET_DVR_GET_ALARMINGCFG_V30	获取报警输入参数	有效	NET_DVR_ALARMINGCFG_V30	1024
NET_DVR_GET_ALARMOUTCFG_V30	获取报警输出参数	有效	NET_DVR_ALARMOUTCFG_V30	1026
NET_DVR_GET_USERCFG_V30	获取用户参数	无效	NET_DVR_USER_V30	1006
NET_DVR_GET_EXCEPTIONCFG_V30	获取异常参数	无效	NET_DVR_EXCEPTION_V30	1034
NET_DVR_GET_ZONEANDDST	获取时区和夏时制参数	无效	NET_DVR_ZONEANDDST	128
NET_DVR_GET_SHOWSTRING_V30	获取叠加字符参数	有效	NET_DVR_SHOWSTRING_V30	1030
NET_DVR_GET_NETAPPCFG	获取网络应用参数(NTP/DDNS)	无效	NET_DVR_NETAPPCFG	222
NET_DVR_GET_NTPCFG	获取网络应用参数(NTP)	无效	NET_DVR_NTPPARA	224
NET_DVR_GET_DDNSCFG_V30	获取网络应用参数(DDNS)	无效	NET_DVR_DDNSPARA_V30	1010
NET_DVR_GET_EMAILCFG_V30	获取网络应用参数(EMAIL)	无效	NET_DVR_EMAILCFG_V30	1012
NET_DVR_GET_NFSCFG	获取 NFS(网络文件系统)参数	无效	NET_DVR_NFSCFG	230
NET_DVR_GET_HDCFG	获取硬盘管理参数	无效	NET_DVR_HDCFG	1054
NET_DVR_GET_COMPRESSCFG_AUDIO	获取语音对讲音频参	无效	NET_DVR_COMPRESSION_AUDIO	1058

D	数			
NET_DVR_GET_AP_INFO_LIST	获取无线网络资源参数	无效	NET_DVR_AP_INFO_LIST	305
NET_DVR_GET_WIFI_CFG	获取 IP 监控设备无线参数	无效	NET_DVR_WIFI_CFG	307
NET_DVR_GET_WIFI_WORKMODE	获取 IP 监控设备网口参数	无效	NET_DVR_WIFI_WORKMODE	309
NET_DVR_GET_NET_DISKCFG	获取网络硬盘接入参数	无效	NET_DVR_NET_DISKCFG	1038
NET_DVR_GET_CCDPARAMCFG	获取前端参数	无效	NET_DVR_CAMERAPARAMCFG	1067
NET_DVR_GET_SNMPCFG	获取 SNMP 参数	无效	NET_DVR_SNMPCFG	1112
NET_DVR_GET_AUDIO_INPUT	获取音频输入参数	有效	NET_DVR_AUDIO_INPUT_PARAM	3201
NET_DVR_GET_CAMERA_DEHAZE_CFG	获取去雾参数	有效	NET_DVR_CAMERA_DEHAZE_CFG	3203
NET_IPC_GET_AUX_ALARMCFG	获取辅助报警参数	有效	NET_IPC_AUX_ALARMCFG	3209

[返回目录](#)

5.26.2 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_V30 的返回值
- [in]dwCommand 设备配置命令，参见配置命令
- [in]IChannel 通道号，如果命令不需要通道号，该参数无效，置为 0xFFFFFFFF 即可
- [in]lpInBuffer 输入数据的缓冲指针
- [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。获取错误码调用 [NET_DVR_GetLastError](#)。

说 明： 不同的获取功能对应不同的结构体和命令号，如下表所示：

- 对于配置命令 NET_DVR_SET_ALARMINGCFG_V30 和 NET_DVR_SET_ALARMOUTCFG_V30，IChannel 为 0，建议应用界面上通道号显示为 1 以保持和设备本地日志的一致。
- 其他配置命令，如果 IChannel 有效，通道号为 1。

dwCommand 宏定义	dwCommand 含义	通道号	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_DEVICECFG_V40	设置设备参数(扩展)	无效	NET_DVR_DEVICECFG_V40	1101
NET_DVR_SET_TIMECFG	设置时间参数	无效	NET_DVR_TIME	119
NET_DVR_SET_NETCFG_V30	设置网络参数	无效	NET_DVR_NETCFG_V30	1001
NET_DVR_SET_PICCFG_V30	设置图像参数	有效	NET_DVR_PICCFG_V30	1003
NET_DVR_SET_COMPRESSCFG_V30	设置压缩参数	有效	NET_DVR_COMPRESSIONCFG_V30	1041

NET_DVR_SET_RECORDCFG_V30	设置录像参数	有效	NET_DVR_RECORD_V30	1005
NET_DVR_SET_DECODERCFG_V30	设置（云台）解码器参数	有效	NET_DVR_DECODERCFG_V30	1043
NET_DVR_SET_RS232CFG_V30	设置 232 串口参数	无效	NET_DVR_RS232CFG_V30	1037
NET_DVR_SET_ALARMINCFG_V30	设置报警输入参数	有效	NET_DVR_ALARMINCFG_V30	1025
NET_DVR_SET_ALARMOUTCFG_V30	设置报警输出参数	有效	NET_DVR_ALARMOUTCFG_V30	1027
NET_DVR_SET_USERCFG_V30	设置用户参数	无效	NET_DVR_USER_V30	1007
NET_DVR_SET_EXCEPTIONCFG_V30	设置异常参数	无效	NET_DVR_EXCEPTION_V30	1035
NET_DVR_SET_ZONEANDDST	设置时区和夏时制参数	无效	NET_DVR_ZONEANDDST	129
NET_DVR_SET_SHOWSTRING_V30	设置叠加字符参数	有效	NET_DVR_SHOWSTRING_V30	1031
NET_DVR_SET_NETAPPCFG	设置网络应用参数 (NTP/DDNS)	无效	NET_DVR_NETAPPCFG	223
NET_DVR_SET_NTPCFG	设置网络应用参数 (NTP)	无效	NET_DVR_NTPPARA	225
NET_DVR_SET_DDNSCFG_V30	设置网络应用参数 (DDNS)	无效	NET_DVR_DDNSPARA_V30	1011
NET_DVR_SET_EMAILCFG_V30	设置网络应用参数 (EMAIL)	无效	NET_DVR_EMAILCFG_V30	1013
NET_DVR_SET_NFSCFG	设置 NFS（网络文件系统）参数	无效	NET_DVR_NFSCFG	231
NET_DVR_SET_HDCFG	设置硬盘管理参数	无效	NET_DVR_HDCFG	1055
NET_DVR_SET_COMPRESSCFG_AUD	设置语音对讲音频参数	无效	NET_DVR_COMPRESSION_AUDIO	1059
NET_DVR_SET_WIFI_CFG	获取 IP 监控设备无线参数	无效	NET_DVR_WIFI_CFG	306
NET_DVR_SET_WIFI_WORKMODE	获取 IP 监控设备网口参数	无效	NET_DVR_WIFI_WORKMODE	308
NET_DVR_SET_NET_DISKCFG	设置网络硬盘接入参数	无效	NET_DVR_NET_DISKCFG	1039
NET_DVR_SET_CCDPARAMCFG	设置前端参数	无效	NET_DVR_CAMERAPARAMCFG	1068
NET_DVR_SET_SNMPCFG	设置 SNMP 参数	无效	NET_DVR_SNMPCFG	1113
NET_DVR_SET_AUDIO_INPUT	设置音频输入参数	有效	NET_DVR_AUDIO_INPUT_PARAM	3202
NET_DVR_SET_CAMERA_DEHAZE_CFG	设置去雾参数	有效	NET_DVR_CAMERA_DEHAZE_CFG	3204
NET_IPC_SET_AUX_ALARMCFG	设置辅助报警参数	有效	NET_IPC_AUX_ALARMCFG	3210

[返回目录](#)

报警输出配置

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

函 数: BOOL NET_DVR_GetAlarmOut_V30(LONG IUserID,
LPNET_DVR_ALARMOUTSTATUS_V30 lpAlarmOutState)
参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V30 的返回值
[out]lpAlarmOutState 报警输出状态
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

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

函 数: BOOL NET_DVR_SetAlarmOut(LONG IUserID, LONG IAlarmOutPort, LONG
IAlarmOutStatic)
参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V30 的返回值
[in]IAlarmOutPort 报警输出口。
初始输出口从 0 开始,
0x00ff 表示全部模拟输出,
0xff00 表示全部数字输出,
DS-90xx 系列设备同时支持对 IP 接入的报警输出进行
处理, 对应 32-95 为数字报警输出
[in]IAlarmOutStatic 报警输出状态: 0—停止输出, 1—输出
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

RTSP 协议参数配置

5.26.5 获取 RTSP 协议参数 **NET_DVR_GetRtspConfig**

函 数: BOOL NET_DVR_GetRtspConfig(LONG IUserID, DWORD dwCommand,
LPNET_DVR_RTSPCFG lpOutBuffer, DWORD dwOutBufferSize)
参 数: [in]IUserID 用户 ID 号, NET_DVR_Login_V30 的返回值
[in]dwCommand 保留, 置为 0
[out]lpOutBuffer 输出缓存
[in]dwOutBufferSize 存放输出数据的缓冲区大小
返回值: TRUE 表示成功, FALSE 表示失败。获取错误码调用 [NET_DVR_GetLastError](#)。
说 明:

[返回目录](#)

5.26.6 设置 RTSP 协议参数 **NET_DVR_SetRtspConfig**

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

参 数： [in]lUserID 用户 ID 号，NET_DVR_Login_V30 的返回值
 [in]dwCommand 保留，置为 0
 [in]lpInBuffer 输入缓存
 [in]dwOutBufferSize 输入缓存的大小，大小为结构体
 NET_DVR_RTSPCFG 的大小

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

说 明：

[返回目录](#)

5.27 邮件测试

5.27.1 测试按已配置的 EMAIL 参数能否收发成功

NET_DVR_StartEmailTest

函 数： LONG NET_DVR_StartEmailTest(LONG lUserID)

参 数： [in]lUserID NET_DVR_Login_V30 的返回值

返回值： -1 表示失败，其他值作为 NET_DVR_GetEmailTestProcess 、
NET_DVR_StopEmailTest 等函数的参数。获取错误码调用
[NET_DVR_GetLastError](#)。

说 明： 在调用此接口测试前，需要配置相关的 EMAIL 参数，详见
NET_DVR_GetDVRConfig、NET_DVR_SetDVRConfig 的网络应用参数（EMAIL）
配置。

[返回目录](#)

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

函 数： BOOL NET_DVR_GetEmailTestProgress(LONG lEmailTestHandle, DWORD*
pState)

参 数： [in]lEmailTestHandle NET_DVR_StartEmailTest 的返回值
 [out]pState 邮件测试的进度，进度值的取值范围(0,100)，
 其他值定义如下

pState 宏定义	宏定义值	含义
PROCESSING	0	正在处理

PROCESS_SUCCESS	100	过程完成
PROCESS_EXCEPTION	400	过程异常
PROCESS_FAILED	500	过程失败

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

[返回目录](#)

5.27.3 停止邮件测试 **NET_DVR_StopEmailTest**

函 数： BOOL NET_DVR_StopEmailTest(LONG IEmailTestHandle)

参 数： [in]IEmailTestHandle NET_DVR_StartEmailTest 的返回值

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

[返回目录](#)

5.28热成像仪

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

函 数： BOOL NET_DVR_ShutterCompensation(LONG IUserID)

参 数： [in]IUserID NET_DVR_Login_V30 的返回值

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

说 明：

[返回目录](#)

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

函 数： BOOL NET_DVR_CorrectDeadPixel(LONG IUserID, LONG IChannel,
LPNET_DVR_CORRECT_DEADPIXEL_PARAM lpInParam)

参 数： [in]IUserID NET_DVR_Login_V30 的返回值

[in]IChannel 通道号

[in]lpInParam 坏点校正参数

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

[返回目录](#)

6 错误代码及说明

6.1 网络通讯库错误码

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

NET_DVR_PASSWORD_FORMAT_ERROR	26	密码输入格式不正确
NET_DVR_DISK_FORMATING	27	硬盘正在格式化，不能启动操作。
NET_DVR_DVRNORESOURCE	28	设备资源不足。
NET_DVR_DVROPRATEFAILED	29	设备操作失败。
NET_DVR_OPENHOSTSOUND_FAIL	30	语音对讲、语音广播操作中采集本地音频或打开音频输出失败。
NET_DVR_DVRVOICEOPENED	31	设备语音对讲被占用。
NET_DVR_TIMEINPUTERROR	32	时间输入不正确。
NET_DVR_NOSPECFILE	33	回放时设备没有指定的文件。
NET_DVR_CREATEFILE_ERROR	34	创建文件出错。本地录像、保存图片、获取配置文件和远程下载录像时创建文件失败。
NET_DVR_FILEOPENFAIL	35	打开文件出错。设置配置文件、设备升级、上传审讯文件时打开文件失败。
NET_DVR_OPERNOTFINISH	36	上次的操作还没有完成
NET_DVR_GETPLAYTIMEFAIL	37	获取当前播放的时间出错。
NET_DVR_PLAYFAIL	38	播放出错。
NET_DVR_FILEFORMAT_ERROR	39	文件格式不正确。
NET_DVR_DIR_ERROR	40	路径错误
NET_DVR_ALLOC_RESOURCE_ERROR	41	SDK 资源分配错误。
NET_DVR_AUDIO_MODE_ERROR	42	声卡模式错误。当前打开声音播放模式与实际设置的模式不符出错。
NET_DVR_NOENOUGH_BUF	43	缓冲区太小。接收设备数据的缓冲区或存放图片缓冲区不足。
NET_DVR_CREATESOCKET_ERROR	44	创建 SOCKET 出错。
NET_DVR_SETSOCKET_ERROR	45	设置 SOCKET 出错。
NET_DVR_MAX_NUM	46	个数达到最大。分配的注册连接数、预览连接数超过 SDK 支持的最大数。
NET_DVR_USERNOTEXIST	47	用户不存在。注册的用户 ID 已注销或不可用。
NET_DVR_WRITEFLASHERROR	48	写 FLASH 出错。设备升级时写 FLASH 失败。
NET_DVR_UPGRADEFAIL	49	设备升级失败。网络或升级文件语言不匹配等原因升级失败。
NET_DVR_CARDHAVEINIT	50	解码卡已经初始化过。
NET_DVR_PLAYERFAILED	51	调用播放库中某个函数失败。
NET_DVR_MAX_USERNUM	52	登录设备的用户数达到最大。
NET_DVR_GETLOCALIPANDMACFAIL	53	获得本地 PC 的 IP 地址或物理地址失败。
NET_DVR_NOENCODEING	54	设备该通道没有启动编码。
NET_DVR_IPMISMATCH	55	IP 地址不匹配。
NET_DVR_MACMISMATCH	56	MAC 地址不匹配。
NET_DVR_UPGRADELANGMISMATCH	57	升级文件语言不匹配。
NET_DVR_MAX_PLAYERPORT	58	播放器路数达到最大。
NET_DVR_NOSPACEBACKUP	59	备份设备中没有足够空间进行备份。
NET_DVR_NODEVICEBACKUP	60	没有找到指定的备份设备。
NET_DVR_PICTURE_BITS_ERROR	61	图像素位数不符，限 24 色。

NET_DVR_PICTURE_DIMENSION_ERROR	62	图片高*宽超限，限 128*256。
NET_DVR_PICTURE_SIZ_ERROR	63	图片大小超限，限 100K。
NET_DVR_LOADPLAYERSDKFAILED	64	载入当前目录下 Player Sdk 出错。
NET_DVR_LOADPLAYERSDKPROC_ERROR	65	找不到 Player Sdk 中某个函数入口。
NET_DVR_LOADDSSDKFAILED	66	载入当前目录下 DsSdk 出错。
NET_DVR_LOADDSSDKPROC_ERROR	67	找不到 DsSdk 中某个函数入口。
NET_DVR_DSSDK_ERROR	68	调用硬解码库 DsSdk 中某个函数失败。
NET_DVR_VOICEMONOPOLIZE	69	声卡被独占。
NET_DVR_JOINMULTICASTFAILED	70	加入多播组失败。
NET_DVR_CREATEDIR_ERROR	71	建立日志文件目录失败。
NET_DVR_BINDSOCKET_ERROR	72	绑定套接字失败。
NET_DVR_SOCKETCLOSE_ERROR	73	socket 连接中断，此错误通常是由于连接中断或目的地不可达。
NET_DVR_USERID_ISUSING	74	注销时用户 ID 正在进行某操作。
NET_DVR_SOCKETLISTEN_ERROR	75	监听失败。
NET_DVR_PROGRAM_EXCEPTION	76	程序异常。
NET_DVR_WRITEFILE_FAILED	77	写文件失败。本地录像、远程下载录像、下载图片等操作时写文件失败。
NET_DVR_FORMAT_READONLY	78	禁止格式化只读硬盘。
NET_DVR_WITHSAMEUSERNAME	79	远程用户配置结构中存在相同的用户名。
NET_DVR_DEVICETYPE_ERROR	80	导入参数时设备型号不匹配。
NET_DVR_LANGUAGE_ERROR	81	导入参数时语言不匹配。
NET_DVR_PARAVERSION_ERROR	82	导入参数时软件版本不匹配。
NET_DVR_IPCHAN_NOTALIVE	83	预览时外接 IP 通道不在线。
NET_DVR_RTSP_SDK_ERROR	84	加载标准协议通讯库 StreamTransClient 失败。
NET_DVR_CONVERT_SDK_ERROR	85	加载转封装库失败。
NET_DVR_IPC_COUNT_OVERFLOW	86	超出最大的 IP 接入通道数。
NET_DVR_MAX_ADD_NUM	87	添加录像标签或者其他操作超出最多支持的个数。
NET_DVR_PARAMMODE_ERROR	88	图像增强仪，参数模式错误（用于硬件设置时，客户端进行软件设置时错误值）。
NET_DVR_CODESPITTER_OFFLINE	89	码分器不在线。
NET_DVR_BACKUP_COPYING	90	设备正在备份。
NET_DVR_CHAN_NOTSUPPORT	91	通道不支持该操作。
NET_DVR_CALLINEINVALID	92	高度线位置太集中或长度线不够倾斜。
NET_DVR_CALCANCELCONFLICT	93	取消标定冲突，如果设置了规则及全局的实际大小尺寸过滤。
NET_DVR_CALPOINTOUTRANGE	94	标定点超出范围。
NET_DVR_FILTERRECTINVALID	95	尺寸过滤器不符合要求。
NET_DVR_DDNS_DEVOFFLINE	96	设备没有注册到 ddns 上。
NET_DVR_DDNS_INTER_ERROR	97	DDNS 服务器内部错误。
NET_DVR_ALIAS_DUPLICATE	150	别名重复（EasyDDNS 的配置）

NET_DVR_DEV_NET_OVERFLOW	800	网络流量超过设备能力上限
NET_DVR_STATUS_RECORDFILE_WRITING_NOT_LOCK	801	录像文件在录像，无法被锁定
NET_DVR_STATUS_CANT_FORMAT_LITTLE_DISK	802	由于硬盘太小无法格式化
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。

6.2 RTSP 通讯库错误码

错误名称	错误值	说明
NET_DVR_RTSP_GETPORTFAILED	407	获取 RTSP 端口错误
NET_DVR_RTSP_DESCRIBESENDTIMEOUT	411	RTSP DESCRIBE 发送超时
NET_DVR_RTSP_DESCRIBESENDERERROR	412	RTSP DESCRIBE 发送失败
NET_DVR_RTSP_DESCRIBERECEVTIMEOUT	413	RTSP DESCRIBE 接收超时
NET_DVR_RTSP_DESCRIBERECEVDATALOST	414	RTSP DESCRIBE 接收数据错误
NET_DVR_RTSP_DESCRIBERECEVERROR	415	RTSP DESCRIBE 接收失败
NET_DVR_RTSP_DESCRIBESERVERERR	416	RTSP DESCRIBE 服务器返回 401,501 等错误

NET_DVR_RTSP_SETUPSENDDTIMEOUT	421	RTSP SETUP 发送超时
NET_DVR_RTSP_SETUPSENDERERROR	422	RTSP SETUP 发送错误
NET_DVR_RTSP_SETUPRECVTIMEOUT	423	RTSP SETUP 接收超时
NET_DVR_RTSP_SETUPRECVDATALOST	424	RTSP SETUP 接收数据错误
NET_DVR_RTSP_SETUPRECVERROR	425	RTSP SETUP 接收失败
NET_DVR_RTSP_OVER_MAX_CHAN	426	设备超过最大连接数
NET_DVR_RTSP_PLAYSENDDTIMEOUT	431	RTSP PLAY 发送超时
NET_DVR_RTSP_PLAYSENDERERROR	432	RTSP PLAY 发送错误
NET_DVR_RTSP_PLAYRECVTIMEOUT	433	RTSP PLAT 接收超时
NET_DVR_RTSP_PLAYRECVDATALOST	434	RTSP PLAY 接收数据错误
NET_DVR_RTSP_PLAYRECVERROR	435	RTSP PLAY 接收失败
NET_DVR_RTSP_PLAYSERVERERR	436	RTSP PLAY 设备返回错误状态
NET_DVR_RTSP_TEARDOWNSENDDTIMEOUT	441	RTSP TEARDOWN 发送超时
NET_DVR_RTSP_TEARDOWNSENDERERROR	442	RTSP TEARDOWN 发送错误
NET_DVR_RTSP_TEARDOWNRECVTIMEOUT	443	RTSP TEARDOWN 接收超时
NET_DVR_RTSP_TEARDOWNRECVDATALOST	444	RTSP TEARDOWN 接收数据错误
NET_DVR_RTSP_TEARDOWNRECVERROR	445	RTSP TEARDOWN 接收失败
NET_DVR_RTSP_TEARDOWNSENDERERR	446	RTSP TEARDOWN 设备返回错误状态

6.3 软解码库错误码

错误名称	错误值	说明
NET_PLAYM4_NOERROR	500	没有错误
NET_PLAYM4_PARA_OVER	501	输入参数非法
NET_PLAYM4_ORDER_ERROR	502	调用顺序不对
NET_PLAYM4_TIMER_ERROR	503	多媒体时钟设置失败
NET_PLAYM4_DEC_VIDEO_ERROR	504	视频解码失败
NET_PLAYM4_DEC_AUDIO_ERROR	505	音频解码失败
NET_PLAYM4_ALLOC_MEMORY_ERROR	506	分配内存失败
NET_PLAYM4_OPEN_FILE_ERROR	507	文件操作失败
NET_PLAYM4_CREATE_OBJ_ERROR	508	创建线程事件等失败
NET_PLAYM4_CREATE_DDRAW_ERROR	509	创建 directDraw 失败
NET_PLAYM4_CREATE_OFFSCREEN_ERROR	510	创建后端缓存失败
NET_PLAYM4_BUF_OVER	511	缓冲区满，输入流失败
NET_PLAYM4_CREATE_SOUND_ERROR	512	创建音频设备失败
NET_PLAYM4_SET_VOLUME_ERROR	513	设置音量失败
NET_PLAYM4_SUPPORT_FILE_ONLY	514	只能在播放文件时才能使用此接口

NET_PLAYM4_SUPPORT_STREAM_ONLY	515	只能在播放流时才能使用此接口
NET_PLAYM4_SYS_NOT_SUPPORT	516	系统不支持，解码器只能工作在 Pentium 3 以上
NET_PLAYM4_FILEHEADER_UNKNOWN	517	没有文件头
NET_PLAYM4_VERSION_INCORRECT	518	解码器和编码器版本不对应
NET_PLAYM4_INIT_DECODER_ERROR	519	初始化解码器失败
NET_PLAYM4_CHECK_FILE_ERROR	520	文件太短或码流无法识别
NET_PLAYM4_INIT_TIMER_ERROR	521	初始化多媒体时钟失败
NET_PLAYM4_BLT_ERROR	522	位拷贝失败
NET_PLAYM4_UPDATE_ERROR	523	显示 overlay 失败
NET_PLAYM4_OPEN_FILE_ERROR_MULTI	524	打开混合流文件失败
NET_PLAYM4_OPEN_FILE_ERROR_VIDEO	525	打开视频流文件失败
NET_PLAYM4_JPEG_COMPRESS_ERROR	526	JPEG 压缩错误
NET_PLAYM4_EXTRACT_NOT_SUPPORT	527	不支持该文件版本.
NET_PLAYM4_EXTRACT_DATA_ERROR	528	提取文件数据失败