

设备(LCD 屏)
网络 SDK 编程指南
V5.1

声 明

非常感谢您购买我公司的产品，如果您有什么疑问或需要请随时联系我们。

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

目 录

声 明	I
目 录	II
1 SDK 简介	1
2 SDK 版本更新	4
3 函数调用顺序	5
3.1 LCD 屏主要流程	5
3.2 LCD 拼接屏功能接口流程	7
3.2.1 LCD 屏服务器	7
3.2.2 带解码功能的 LCD 屏	8
3.3 报警模块流程	10
3.3.1 报警布防	10
3.3.2 报警监听	11
3.4 透明通道	12
4 函数调用实例	13
4.1 LCD 屏服务器拼接示例代码	13
5 函数说明	23
5.1 SDK 初始化	23
5.1.1 初始化 SDK NET_DVR_Init	23
5.1.2 释放 SDK 资源 NET_DVR_Cleanup	23
5.1.3 设置网络连接超时时间和连接尝试次数 NET_DVR_SetConnectTime	23
5.1.4 设置重连功能 NET_DVR_SetReconnect	23
5.1.5 设置接收超时时间 NET_DVR_SetRecvTimeOut	24
5.1.6 获取所有 IP，用于支持多网卡接口 NET_DVR_GetLocalIP	24
5.1.7 设置 IP 绑定 NET_DVR_SetValidIP	24
5.1.8 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	24
5.2 异常消息回调	25
5.2.1 注册接收异常、重连等消息回调函数 NET_DVR_SetExceptionCallBack_V30	25
5.3 SDK 日志和信息	27
5.3.1 获取 SDK 的版本号和 build 信息 NET_DVR_GetSDKBuildVersion	27
5.3.2 获取当前 SDK 的状态信息 NET_DVR_GetSDKState	27
5.3.3 获取当前 SDK 的功能信息 NET_DVR_GetSDKAbility	27
5.3.4 获取 SDK 本地参数 NET_DVR_GetSDKLocalCfg	28
5.3.5 设置 SDK 本地参数 NET_DVR_SetSDKLocalCfg	28
5.3.6 启用写日志文件 NET_DVR_SetLogToFile	29
5.4 获取错误信息	29
5.4.1 返回最后操作的错误码 NET_DVR_GetLastError	29
5.4.2 返回最后操作的错误码信息 NET_DVR_GetErrorMsg	29
5.5 用户注册	30
5.5.1 用户注册设备 NET_DVR_Login_V30	30
5.5.2 用户注销 NET_DVR_Logout	30
5.6 设备能力集	30

5.6.1	获取设备能力集 NET_DVR_GetDeviceAbility	30
5.7	布防、撤防.....	32
	设置报警等信息上传的回调函数	32
5.7.1	注册回调函数, 接收设备报警消息 NET_DVR_SetDVRMessageCallBack_V30.....	32
	布防撤防	33
5.7.2	建立报警上传通道, 获取报警等信息 NET_DVR_SetupAlarmChan_V41	33
5.7.3	撤销报警上传通道 NET_DVR_CloseAlarmChan_V30	33
5.8	监听报警.....	33
5.8.1	启动监听, 接收设备主动上传的报警等信息 NET_DVR_StartListen_V30	33
5.8.2	停止监听 (支持多线程) NET_DVR_StopListen_V30	34
5.9	通用参数配置.....	34
5.9.1	获取设备的配置信息 NET_DVR_GetDVRConfig	34
5.9.2	设置设备的配置信息 NET_DVR_SetDVRConfig	36
5.10	LCD 屏远程参数配置	36
5.10.1	启动远程配置 (屏幕拼接、配置参数) NET_DVR_StartRemoteConfig.....	36
5.10.2	发送长连接数据 (设置屏幕拼接参数) NET_DVR_SendRemoteConfig.....	38
5.10.3	关闭长连接 NET_DVR_StopRemoteConfig.....	39
5.11	解码功能.....	39
	显示通道配置	39
5.11.1	获取显示通道配置参数 NET_DVR_MatrixGetDisplayCfg_V41	39
5.11.2	设置显示通道配置参数 NET_DVR_MatrixSetDisplayCfg_V41.....	39
5.12	窗口解码操作.....	40
	动态解码.....	40
5.12.1	启动动态解码 NET_DVR_MatrixStartDynamic_V41.....	40
5.12.2	停止动态解码 NET_DVR_MatrixStopDynamic	40
	远程文件回放	40
5.12.3	远程文件回放解码 NET_DVR_RemoteControl.....	40
5.12.4	远程回放文件控制 NET_DVR_MatrixSetRemotePlayControl	41
5.12.5	获取回放状态 NET_DVR_MatrixGetRemotePlayStatus.....	41
	解码开关控制	42
5.12.6	获取当前解码通道开关 NET_DVR_MatrixGetDecChanEnable.....	42
5.12.7	设置当前解码通道开关 NET_DVR_MatrixSetDecChanEnable	42
	轮巡解码.....	43
5.12.8	获取轮巡解码通道 NET_DVR_MatrixGetLoopDecChanInfo_V41	43
5.12.9	设置轮巡解码通道 NET_DVR_MatrixSetLoopDecChanInfo_V41.....	43
5.12.10	获取单个解码通道轮巡开关 NET_DVR_MatrixGetLoopDecChanEnable.....	43
5.12.11	设置单个解码通道轮巡开关 NET_DVR_MatrixSetLoopDecChanEnable	44
5.12.12	获取所有解码通道轮巡开关 NET_DVR_MatrixGetLoopDecEnable.....	44
	获取解码信息	44
5.12.13	获取当前解码通道信息 NET_DVR_MatrixGetDecChanInfo_V41.....	44
5.12.14	获取当前解码状态 NET_DVR_MatrixGetDeviceStatus_V41.....	45
5.13	场景配置、操作.....	45
5.13.1	远程控制 (场景、屏幕控制以及矩阵输入输出切换) NET_DVR_RemoteControl	45
5.13.2	批量获取信息 NET_DVR_GetDeviceConfig	45

5.13.3	批量设置参数 NET_DVR_SetDeviceConfig	47
5.14	透明通道.....	49
5.14.1	建立透明通道 NET_DVR_SerialStart_V40	49
5.14.2	通过透明通道向设备串口发送数据 NET_DVR_SerialSend.....	49
5.14.3	断开透明通道 NET_DVR_SerialStop	50
5.15	设备维护.....	50
远程升级.....		50
5.15.1	设置远程升级时网络环境 NET_DVR_SetNetworkEnvironment.....	50
5.15.2	远程升级 NET_DVR_Upgrade	50
5.15.3	获取远程升级的进度 NET_DVR_GetUpgradeProgress	51
5.15.4	获取远程升级的状态 NET_DVR_GetUpgradeState	51
5.15.5	获取远程升级的阶段信息 NET_DVR_GetUpgradeStep.....	51
5.15.6	关闭远程升级句柄, 释放资源 NET_DVR_CloseUpgradeHandle	52
日志查找.....		52
5.15.7	查找设备的日志信息 NET_DVR_FindDVRLog_V30.....	52
5.15.8	逐条获取查找到的日志信息 NET_DVR_FindNextLog_V30.....	52
5.15.9	释放查找日志的资源 NET_DVR_FindLogClose_V30.....	53
远程恢复默认参数、重启		53
5.15.10	恢复设备默认参数 NET_DVR_RestoreConfig	53
5.15.11	重启设备 NET_DVR_RebootDVR	53
配置文件导入导出		54
5.15.12	导出配置文件 NET_DVR_GetConfigFile	54
5.15.13	导入配置文件 NET_DVR_SetConfigFile	54
6	错误代码及说明.....	55
6.1	网络通讯库错误码.....	55
6.2	RTSP 通讯库错误码.....	59
6.3	软解码库错误码.....	59
7	结构体说明.....	61
7.1	宏定义.....	61
7.2	NET_DVR_ADC_CFG: ADC 参数结构体.....	62
7.3	NET_DVR_ALARMER:报警设备信息	63
7.4	NET_DVR_AREA_ZOOM_CFG:区域放大还原参数结构体	64
7.5	NET_DVR_ASSOCIATE_INPUT_PARAM:关联的输入设备信息结构体	65
7.6	NET_DVR_ASSOCIATE_OUTPUT_PARAM: 关联的输出显示屏信息结构体	65
7.7	NET_DVR_BOOT_LOGO_CFG:开机 LOGO 参数结构体	66
7.8	NET_DVR_COLOR_TEMPERATURE_CFG:色温参数结构体	66
7.9	NET_DVR_COMPRESSIONCFG_V30:通道压缩参数结构体.....	67
7.10	NET_DVR_COMPRESSION_INFO_V30:码流压缩参数	67
7.11	NET_DVR_DAYTIME:时间参数结构体.....	69
7.12	NET_DVR_DDNS_ADDRESS:设备 DDNS 域名信息	69
7.13	NET_DVR_DEC_DDNS_DEV:动态域名取流配置	70
7.14	NET_DVR_DEC_STREAM_DEV_EX:设备取流配置.....	70
7.15	NET_DVR_DEC_STREAM_MODE:取流模式配置	71
7.16	NET_DVR_DECODER_WORK_STATUS_V41:解码设备状态参数结构体.....	71

7.17	NET_DVR_DEFOG_LCD:去雾参数结构体.....	72
7.18	NET_DVR_DELAY_TIME:延时开机参数结构体.....	72
7.19	NET_DVR_DEV_CHAN_INFO_EX:前端编码设备信息.....	73
7.20	NET_DVR_DEV_DDNS_INFO:动态域名参数配置.....	74
7.21	NET_DVR_DEV_IP_INFO:屏幕 IP 信息结构体.....	75
7.22	NET_DVR_DEVICECFG_V40:设备参数.....	76
7.23	NET_DVR_DEVICEINFO_V30:设备参数结构体.....	79
7.24	NET_DVR_DISP_CHAN_STATUS_V41:显示通道状态结构体.....	81
7.25	NET_DVR_DISPLAY_COLOR_CTRL:显示单元颜色控制结构体.....	82
7.26	NET_DVR_DISPLAY_POSITION_CTRL:显示单元位置控制结构体.....	83
7.27	NET_DVR_ETHERNET_V30:以太网配置参数.....	83
7.28	NET_DVR_EXCEPTION_V30:异常参数.....	84
7.29	NET_DVR_EXCEPTION_V40:异常参数配置(扩展).....	85
7.30	NET_DVR_EXTERNAL_MATRIX_CFG:矩阵信息结构体.....	85
7.31	NET_DVR_HANDLEEXCEPTION:报警和异常处理结构体.....	87
7.32	NET_DVR_HANDLEEXCEPTION_V30:报警和异常处理.....	87
7.33	NET_DVR_HANDLEEXCEPTION_V41:异常参数信息结构体.....	88
7.34	NET_DVR_HIDEALARM:遮挡报警参数结构体.....	88
7.35	NET_DVR_HOLIDAY_PARAM_CFG:假日参数配置结构体.....	89
7.36	NET_DVR_HOLIDAY_PARAM:假日参数结构体.....	90
7.37	NET_DVR_HOLIDATE_UNION:假日日期结构联合体.....	90
7.38	NET_DVR_HOLIDATE_MODEA:假日日期模式 A.....	91
7.39	NET_DVR_HOLIDATE_MODEB:假日日期模式 B.....	91
7.40	NET_DVR_HOLIDATE_MODEC:假日日期模式 C.....	92
7.41	NET_DVR_INPUT_INTERFACE_CTRL:输入接口控制结构体.....	92
7.42	NET_DVR_IO_RELATION_INFO:矩阵输入输出关联关系信息结构体.....	93
7.43	NET_DVR_IP_ADDRESS:设备 IP 地址信息.....	94
7.44	NET_DVR_IPADDR:IP 地址.....	94
7.45	NET_DVR_LCD_ALARM:LCD 屏幕报警信息结构体.....	94
7.46	NET_DVR_LCD_AUDIO_CFG:LCD 音频参数结构体.....	95
7.47	NET_DVR_LOCAL_ABILITY_PARSE_CFG:能力集解析库配置.....	96
7.48	NET_DVR_LOCAL_BYTE_ENCODE_CONVERT:字符编码转换参数结构体.....	96
7.49	NET_DVR_LOCAL_CHECK_DEV:设备在线巡检参数结构体.....	97
7.50	NET_DVR_LOCAL_MEM_POOL_CFG:内存池本地配置.....	98
7.51	NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG:按模块配置超时时间.....	98
7.52	NET_DVR_LOCAL_PROTECT_KEY_CFG:密钥配置.....	99
7.53	NET_DVR_LOCAL_TALK_MODE_CFG:对讲模式配置.....	99
7.54	NET_DVR_LOCAL_TCP_PORT_BIND_CFG:本地 TCP 端口绑定配置.....	99
7.55	NET_DVR_LOCAL_UDP_PORT_BIND_CFG:本地 UDP 端口绑定配置.....	100
7.56	NET_DVR_LOG_V30:日志信息.....	101
7.57	NET_DVR_MATRIX_ABILITY:解码显示和解码能力输出参数结构体.....	102
7.58	NET_DVR_MATRIX_CHAN_INFO_V41:动态解码轮巡通道信息.....	104
7.59	NET_DVR_MATRIX_CHAN_STATUS:解码通道状态结构体.....	105
7.60	NET_DVR_MATRIX_DEC_CHAN_INFO_V41:当前窗口解码信息.....	107

7.61	NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX:远程文件回放解码参数	108
7.62	NET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS:回放状态信息	109
7.63	NET_DVR_MATRIX_LOOP_DECINFO_V41:动态解码轮巡通道参数	110
7.64	NET_DVR_MATRIX_PASSIVEMODE:被动解码参数.....	110
7.65	NET_DVR_MATRIX_SWITCH_INFO:矩阵切换信息结构体	111
7.66	NET_DVR_MATRIX_VOUTCFG:显示通道配置结构体	112
7.67	NET_DVR_MOTION:移动侦测参数结构体	113
7.68	NET_DVR_MSC_SCREEN_BACKLIGHT_CFG:屏幕背光参数结构体	114
7.69	NET_DVR_MSC_SCREEN_INTERFACE_CFG:输入源参数结构体.....	114
7.70	NET_DVR_MSC_SCREEN_PIP_CFG:屏幕画中画参数结构体	115
7.71	NET_DVR_MSC_SCREEN_PARAM:屏幕参数联合体.....	116
7.72	NET_DVR_MSC_SCREEN_PARAM_CFG:屏幕参数结构体	117
7.73	NET_DVR_MSC_SCREEN_PARAM_COND:屏幕相关参数配置条件结构体	118
7.74	NET_DVR_MSC_SCREEN_REMOTE_CFG:屏幕相关参数配置结构体	118
7.75	NET_DVR_MSC_SPLICE_CFG:屏幕拼接配置结构体	119
7.76	NET_DVR_NETCFG_V30:网络配置	120
7.77	NET_DVR_NTTPARA:网络应用参数(NTP)	121
7.78	NET_DVR_PICCFG:通道图像结构体.....	122
7.79	NET_DVR_PLAY_BACK_BY_TIME:回放时间信息	124
7.80	NET_DVR_PPPCFG_V30:PPP 参数结构体	124
7.81	NET_DVR_PPPOECFG:PPPoE 配置结构体	125
7.82	NET_DVR_PU_STREAM_CFG_V41:动态解码参数.....	125
7.83	NET_DVR_PU_STREAM_URL:URL 取流配置.....	126
7.84	NET_DVR_RECTCFG:窗口设置结构体	127
7.85	NET_DVR_RECTCFG_EX:矩形窗口参数	127
7.86	NET_DVR_RS232CFG_V30:RS232 串口参数结构体.....	128
7.87	NET_DVR_SCENE_CONTROL_INFO:场景控制信息	128
7.88	NET_DVR_SCHEDTIME:起止时间段参数	129
7.89	NET_DVR_SCREEN_BASE_INFO:屏幕基本信息联合体	129
7.90	NET_DVR_SCREEN_CONTROL_PARAM:屏幕控制参数联合体	129
7.91	NET_DVR_SCREEN_CONTROL_V41:屏幕控制结构体.....	130
7.92	NET_DVR_SCREEN_DISPLAY_CFG:显示参数配置结构体.....	132
7.93	NET_DVR_SCREEN_EDGE_CFG:屏幕边缘参数结构体.....	133
7.94	NET_DVR_SCREEN_FAN_WORK_MODE_CFG:风扇工作方式配置结构体	133
7.95	NET_DVR_SCREEN_INDEX_CFG:屏幕索引相关参数配置结构体.....	134
7.96	NET_DVR_SCREEN_MENU_CFG:屏幕菜单配置结构体	135
7.97	NET_DVR_SCREEN_VGA_CFG:VGA 信号配置结构体.....	135
7.98	NET_DVR_SCREEN_WALL_CTRL: 屏幕维墙参数结构体.....	136
7.99	NET_DVR_SCREEN_WORK_STATE: 屏幕工作状态参数结构体.....	136
7.100	NET_DVR_SDKSTATE:SDK 状态信息	137
7.101	NET_DVR_SDKABL:SDK 功能信息	138
7.102	NET_DVR_SERIAL_CONTROL:串口信息结构体.....	139
7.103	NET_DVR_SETUPALARM_PARAM:报警布防参数	139
7.104	NET_DVR_SIMULATE_REMOTE_CTRL:模拟遥控按键参数结构体	140

7.105	NET_DVR_SINGLE_RS232:RS232 串口参数.....	140
7.106	NET_DVR_SNAP_ABILITY:智能交通摄像机能力集参数结构体	141
7.107	NET_DVR_STREAM_MEDIA_SERVER:流媒体服务器参数.....	144
7.108	NET_DVR_SWITCH_DAY_TIME: 开关机计划参数结构体	145
7.109	NET_DVR_SWITCH_TIME_POINT: 按时间点开关机参数结构体	145
7.110	NET_DVR_SYSTEM_TIME:时间信息结构体	146
7.111	NET_DVR_TIME:时间参数.....	146
7.112	NET_DVR_TIMEPOINT:时间点参数.....	147
7.113	NET_DVR_TIME_SWITCH_CFG:时间点参数.....	148
7.114	NET_DVR_USING_SERIALPORT: 当前使用串口信息结构体	148
7.115	NET_VCA_BEHAVIOR_ABILITY:异常行为识别能力输出参数结构体	149
7.116	NET_VCA_DEV_ABILITY:智能设备能力集结构体	150
7.117	NET_DVR_VILOST:信号丢失报警参数结构体	152
7.118	NET_DVR_VCS_USER_INFO:用户信息结构体.....	153
7.119	NET_DVR_VIDEO_OUT_CFG: 视频参数结构体	154
7.120	NET_DVR_VIDEO_PLATFORM: 显示通道配置联合体.....	155
7.121	NET_DVR_VIDEO_WALL_AREA: 电视墙区域信息结构体	155
7.122	NET_DVR_VIDEO_WALL_INFO:电视墙信息	156
7.123	NET_DVR_VIDEOEFFECT:视频参数.....	156
7.124	NET_DVR_VIDEOWALLDISPLAYPOSITION:电视墙输出位置配置.....	157
7.125	NET_DVR_VIDEOWALLWINDOWPOSITION:电视墙窗口信息	158
7.126	NET_DVR_WALLOUTPUTPARAM:电视墙显示输出参数	158
7.127	NET_DVR_WALLSCENECFG:电视墙场景信息	159
7.128	NET_DVR_ZONEANDDST:夏令时参数	160
7.129	NET_MATRIX_ANALOGMATRIX:模拟矩阵参数结构体	160
7.130	NET_MATRIX_DIGITALMATRIX:数字矩阵参数结构体.....	161
7.131	NET_MATRIX_UNION:矩阵参数联合体	161
7.132	LCD 屏幕能力 XML 描述	162
7.133	日志类型	168

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、视频服务器、网络摄像机、网络球机、解码器、多屏控制器、报警主机、LCD 屏等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。本文档主要介绍 LCD 屏相关的功能。

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

表 1.1 Windows SDK 组件

网络通讯库	外部接口	HCNetSDK.h	头文件	
		HCNetSDK.lib	LIB 库文件	
		HCNetSDK.dll	DLL 库文件	
	核心组件	HCCore.lib	LIB 库文件	
		HCCore.dll	DLL 库文件	
组件库	设备配置核心组件	HCCoreDevCfg.dll	DLL 库文件	HCNetSDKCom 文件夹
	预览组件	HCPreview.lib	LIB 库文件	HCNetSDKCom 文件夹
		HCPreview.dll	DLL 库文件	HCNetSDKCom 文件夹
	回放组件	HCPlayBack.dll	DLL 库文件	HCNetSDKCom 文件夹
	语音组件	HCVoiceTalk.dll	DLL 库文件	HCNetSDKCom 文件夹
	报警组件	HCAalarm.lib	LIB 库文件	HCNetSDKCom 文件夹
		HCAalarm.dll	DLL 库文件	HCNetSDKCom 文件夹
	显示组件	HCDisplay.dll	DLL 库文件	HCNetSDKCom 文件夹
	行业应用管理配置组件	HCIIndustry.dll	DLL 库文件	HCNetSDKCom 文件夹
	维护管理配置组件	HCGeneralCfgMgr.lib	LIB 库文件	HCNetSDKCom 文件夹
HCGeneralCfgMgr.dll		DLL 库文件	HCNetSDKCom 文件夹	
RTSP 通讯库		StreamTransClient.dll	DLL 库文件	HCNetSDKCom 文件夹
转封装库		SystemTransform.dll	DLL 库文件	HCNetSDKCom 文件夹
语音对讲库		AudioIntercom.dll	DLL 库文件	HCNetSDKCom 文件夹
		OpenAL32.dll	DLL 库文件	HCNetSDKCom 文件夹
字符转码库		libiconv2.dll	DLL 库文件	HCNetSDKCom 文件夹
模拟能力集		LocalXml.zip	XML 文件包	
软解码库		plaympeg4.h	头文件	
		PlayCtrl.lib	LIB 库文件	
		PlayCtrl.dll	DLL 库文件	
		AudioRender.dll	DLL 库文件	

		SuperRender.dll	DLL 库文件	
硬解码库		DsSdk.dll	DLL 库文件	HCNetSDKCom 文件夹

表 1.2 Linux SDK 组件

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

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

- **网络通讯库**：设备网络 SDK 的主体，主要用于网络客户端与各类产品之间的通讯交互，负责远程功能调控，远程参数配置及码流数据的获取和处理等。设备网络 SDK V5.0 针对产品应用业务进行细化，对之前版本的 SDK 的功能模块进行组件化，其中外部接口（HCNetSDK.dll）仍然保持和设备网络 SDK V4.x 版本保存一致(向下兼容)，其他单独的业务功能（预览、回放等）可以加载单独的模块组件，多个业务功能也可以组合使用。**更新 SDK 时，HCNetSDK.dll、HCCore.dll 以及 HCNetSDKCom 文件夹下的功能组件库文件都需要更新加载，且 HCNetSDKCom 文件夹名不能修改。**
- **hpr 库**：网络通讯库的依赖库，Linux SDK 使用时和网络通讯库同时加载。
- **RTSP 通讯库**：支持 RTSP 传输协议的网络库。当需要对支持 RTSP 协议的产品进行取流等操作时必须加载该项组件。

- **转封装库：**库的功能可以分为两种：一种是将标准码流转换成采用我们公司封装格式的码流。当用户需要对支持 RTSP 协议的产品捕获采用本公司封装格式的码流数据时（即当设置 NET_DVR_RealPlay_V40 接口中的回调函数捕获数据或者调用 NET_DVR_SetRealDataCallBack 接口捕获数据时）必须加载该组件。另一种功能是将标准码流转换成其他格式的封装，如 3GPP、PS 等。例如，当用户需要对支持 RTSP 协议的产品实时捕获指定封装格式的码流数据（对应的 SDK 接口为 NET_DVR_SaveRealData）时必须加载该项组件。
- **语音对讲库：**用于语音对讲时通过声卡采集数据并按照指定的编码格式编码码流或者解码播放音频码流数据（不带封装格式的码流数据）。V4.2.2.5 及以前版本 SDK 均采用 windows API 实现相关功能。之后版本默认使用语音对讲库的方式，通过接口 [NET_DVR_SetSDKLocalCfg](#) 可以选择之前的 windows API 模式。OpenAL32.dll 为依赖库，语音对讲库模式下必须加载。**Windows64 位或者 Linux 系统下无语音对讲功能。CVR 暂不支持语音对讲功能。**
- **字符转换库：**电脑字符集和设备字符集不一致时，SDK 内部需要进行字符编码转换，SDK 默认使用 libiconv 库进行类型转换。如果用户不想使用 libiconv 编码库，可以调用 [NET_DVR_SetSDKLocalCfg](#) (类型: NET_SDK_LOCAL_CFG_TYPE_BYTE_ENCODE) 设置字符转码回调函数，将用户自己的字符编码接口告知 SDK，然后 SDK 将使用用户提供的字符编码接口进行字符串处理。
- **模拟能力集：**如果需要获取设备能力集（[NET_DVR_GetDeviceAbility](#)），建议调用 [NET_DVR_SetSDKLocalCfg](#) 启用模拟能力集，此时需要加载 LocalXml.zip（要求和网络通讯库放在同一个目录下）。
- **软解码库：**主要用于对实时码流数据进行解码显示（实现预览功能）和对录像文件进行回放解码等。用户如果需要在 SDK 内部进行对实时流和录像码流播放显示时（即 NET_DVR_RealPlay_V40 接口的第二个结构体参数的播放句柄设置成有效句柄时）必须加载该组件，而如果用户仅需要用网络通讯库捕获到数据后再外部自行处理就不需要加载该组件，这种情况下用户在外自行解码将更灵活，可参见软解码库函数说明《播放器 SDK 编程指南》。**Linux 64 位系统不支持软解码功能，预览、回放等窗口句柄传空，仅支持只取流不解码。**
- **硬解码库：**需在配备硬解码卡（MD 卡）的前提下使用，通过解码卡的解码与输出功能实现实时流的解码显示及向监视器上矩阵输出的功能。**Windows64 位或者 Linux 系统下无该硬解码库。对于 LCD 屏，不需要使用该库文件。**

2 SDK 版本更新

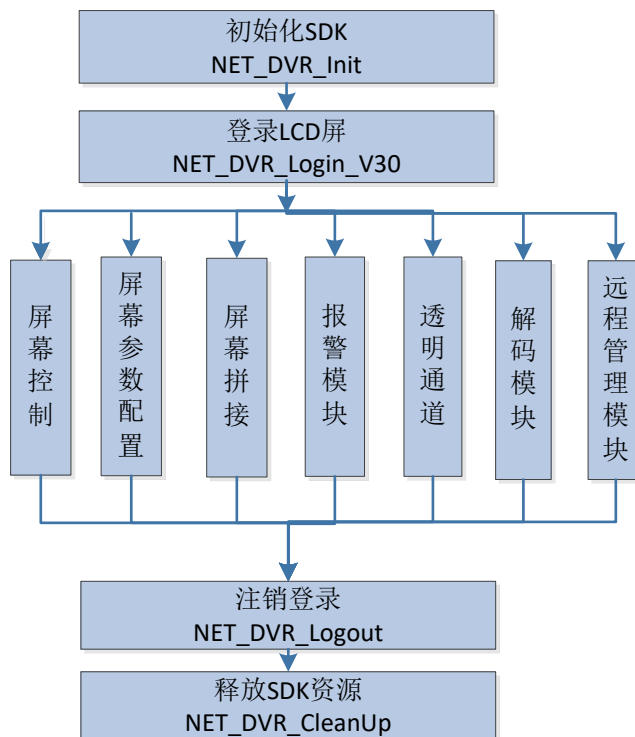
Version 5.1.3.10(build20150720)

- DS-D50XX V1.0
- [NET_DVR_StartRemoteConfig](#)(命令: NET_DVR_SET_SCREEN_PARAM), 可以设置输入源、风扇工作模式、VGA 信号、屏幕菜单、显示效果、色温参数、ADC 参数、屏幕边缘参数、背光参数、画中画参数、去雾参数、屏幕工作状态、开机 logo 参数以及 LCD 音频参数等。
- 设置屏幕控制参数, 通过 [NET_DVR_RemoteControl](#) (命令: NET_DVR_CONTROL_SCREEN), 可以设置屏幕控制各种参数, 如屏幕输入接口、颜色控制、位置控制、模拟遥控案件、屏幕维墙以及区域放大还原参数。
- [NET_DVR_SCREEN_CONTROL_PARAM](#) 中新增 **区域放大还原功能**, 新增 [NET_DVR_RECTCFG](#)、[NET_DVR_AREA_ZOOM_CFG](#) 结构体;
- 屏幕拼接 [NET_DVR_MSC_SPLICE_CFG](#) 新增 **是否使能功能**;
- LONG_CFG_SEND_DATA_TYPE_ENUM 类型中新增 **ENUM_MSC_SEND_DATA** 屏幕控制器数据类型。
- 输入源类型枚举中 INPUT_INTERFACE_TYPE 中新增 **INTERFACE_IP,INTERFACE_USB,INTERFACE_SDI,INTERFACE_DP,INTERFACE_HDBASET** 类型;
- 屏幕工作状态 [NET_DVR_SCREEN_WORK_STATE](#) 结构体中新增增加 **byFanExceptionStatus** 风扇异常状态, **byTempState** 板卡温度状态;
- [NET_DVR_RemoteControl](#) 新增增加 [NET_DVR_SCENE_CONTROL_INFO](#) 场景控制信息结构体;
- 获取外接矩阵输入输出关联关系 [NET_DVR_GetDeviceConfig](#) 接口 [NET_DVR_EXTERNAL_MATRIX_CFG](#) 结构体中 **byMatrixProtocol** 新增增加自定义选项;

3 函数调用顺序

3.1 LCD 屏主要流程

表 3.1 LCD 屏主要流程



LCD 屏功能模块包括屏幕控制、屏幕参数配置功能、屏幕拼接功能、报警功能、透明通道、解码模块、远程管理等。LCD 屏幕服务器可以在安装了服务器的 PC 上，通过串口连接 LCD 屏幕，然后发命令给服务器，服务器转换为串口命令控制 LCD 屏幕。

- 在初始化 SDK 和登录设备之后，可通过接口 [NET_DVR_GetDeviceAbility](#) (能力集类型: LCD_SCREEN_ABILITY) 获取设备能力集，包括软硬件能力集、LCD 屏能力集、解码能力集、网络应用参数能力集等。
- 设置屏幕控制参数，通过 [NET_DVR_RemoteControl](#) (命令: NET_DVR_CONTROL_SCREEN)，可以设置屏幕控制各种参数，如屏幕输入接口、颜色控制、位置控制、模拟遥控案件、屏幕维墙以及区域放大还原参数。(对于 LCD 拼接屏(屏幕服务器)，支持的功能可以通过获取的能力集确认是否支持相应功能)。
- 设置屏幕参数，通过 [NET_DVR_StartRemoteConfig](#) (命令: NET_DVR_SET_SCREEN_PARAM)，可以设置输入源、风扇工作模式、VGA 信号、屏幕菜单、显示效果、色温参数、ADC 参数、屏幕边缘参数、背光参数、画中画参数、去雾参数、屏幕工作状态、开机 logo 参数以及 LCD 音频参数等。
- 设置屏幕拼接，通过 [NET_DVR_StartRemoteConfig](#) (命令: NET_DVR_SCREEN_SPLICE_SET) 设置屏幕拼接；具体实现请参考【[3.2.1 LCD 拼接屏功能接口流程](#)】。
- 带解码板的屏幕相当于单路解码器，可以单独进行登录，可通过 sdk 接口控制屏幕进行屏幕分割，解码上墙。具体实现请参考【[3.2.2 带解码功能的 LCD 屏](#)】
- 报警模块处理从前端服务器的各种上传报警信号，解析后提供给上层的应用。LCD 屏支持布防和监听两

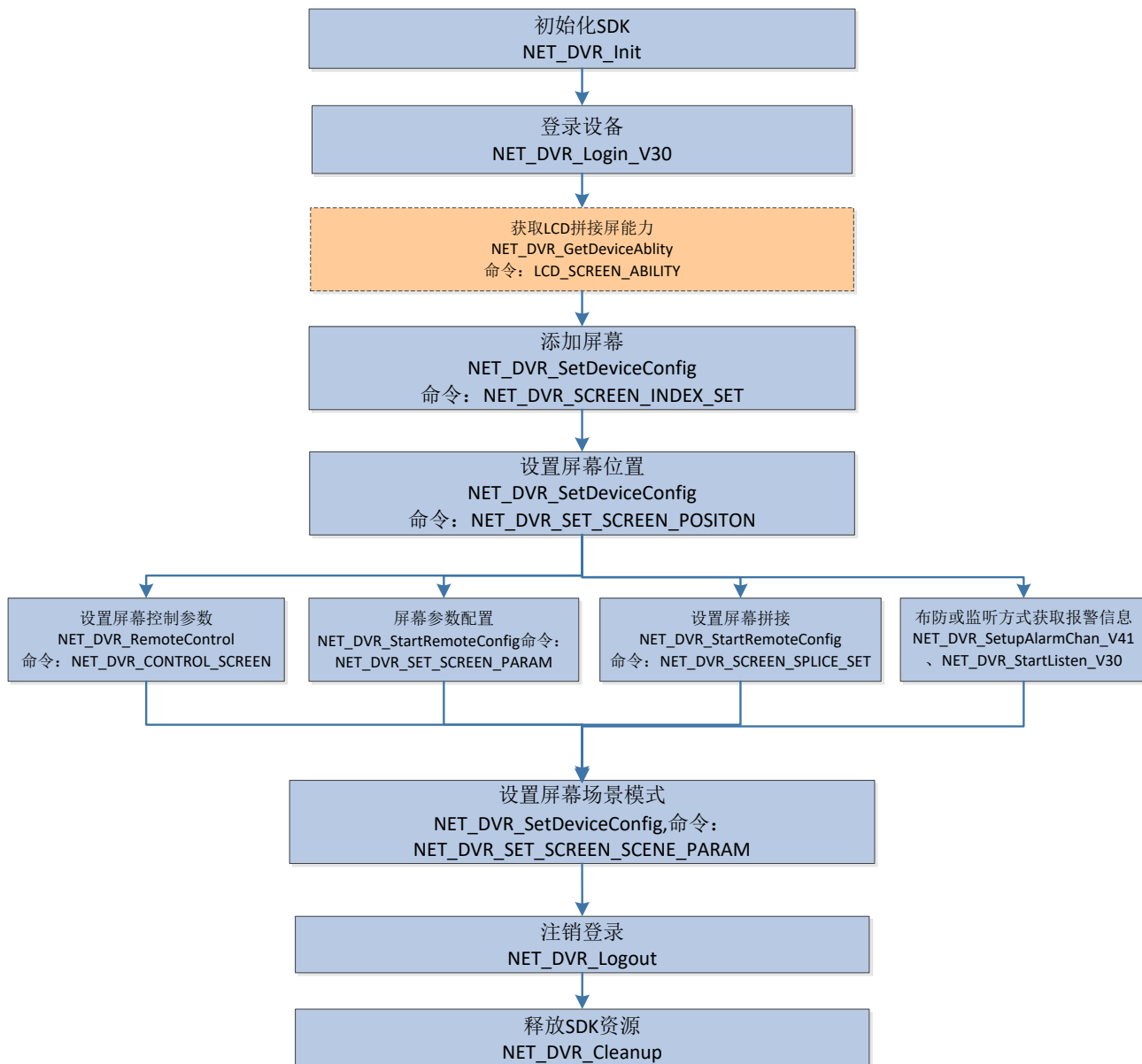
种方式的报警上传方式。通过布防 [NET DVR SetupAlarmChan V41](#) 或者监听 [NET DVR StartListen V30](#) 的方式获取 LCD 报警，报警类型为 COMM_ALARM_LCD；具体实现请参考【[3.3 报警模块流程](#)】。

- 透明通道模块实现 SDK 和设备之间通过串口进行通信，通常实现对设备的控制。具体实现请参考【[3.4 透明通道接口流程](#)】。
- 设备维护模块包括远程升级设备、日志查找、配置文件导入导出、恢复设备默认参数、远程重启相关功能，相关接口参见【[5.15 设备维护](#)】。

3.2 LCD 拼接屏功能接口流程

3.2.1 LCD 屏服务器

图 3.1 LCD 屏服务器接口调用流程



- 登录设备后通过接口 [NET_DVR_GetDeviceAbility](#)（能力集类型：LCD_SCREEN_ABILITY）获取 LCD 拼接屏能力集，包括支持的屏幕/窗口/画面分割个数等信息。
- 通过 [NET_DVR_SetDeviceConfig](#)（命令：NET_DVR_SCREEN_INDEX_SET）添加屏幕。
- 设置屏幕位置，通过 [NET_DVR_SetDeviceConfig](#)（命令：NET_DVR_SET_SCREEN_POSITION）可以设置各 LCD 屏显示位置参数。
- 设置屏幕控制参数，通过 [NET_DVR_RemoteControl](#)（命令：NET_DVR_CONTROL_SCREEN），可以设置屏幕控制各种参数，如屏幕输入接口、颜色控制、位置控制、模拟遥控案件、屏幕维墙以及区域放大还

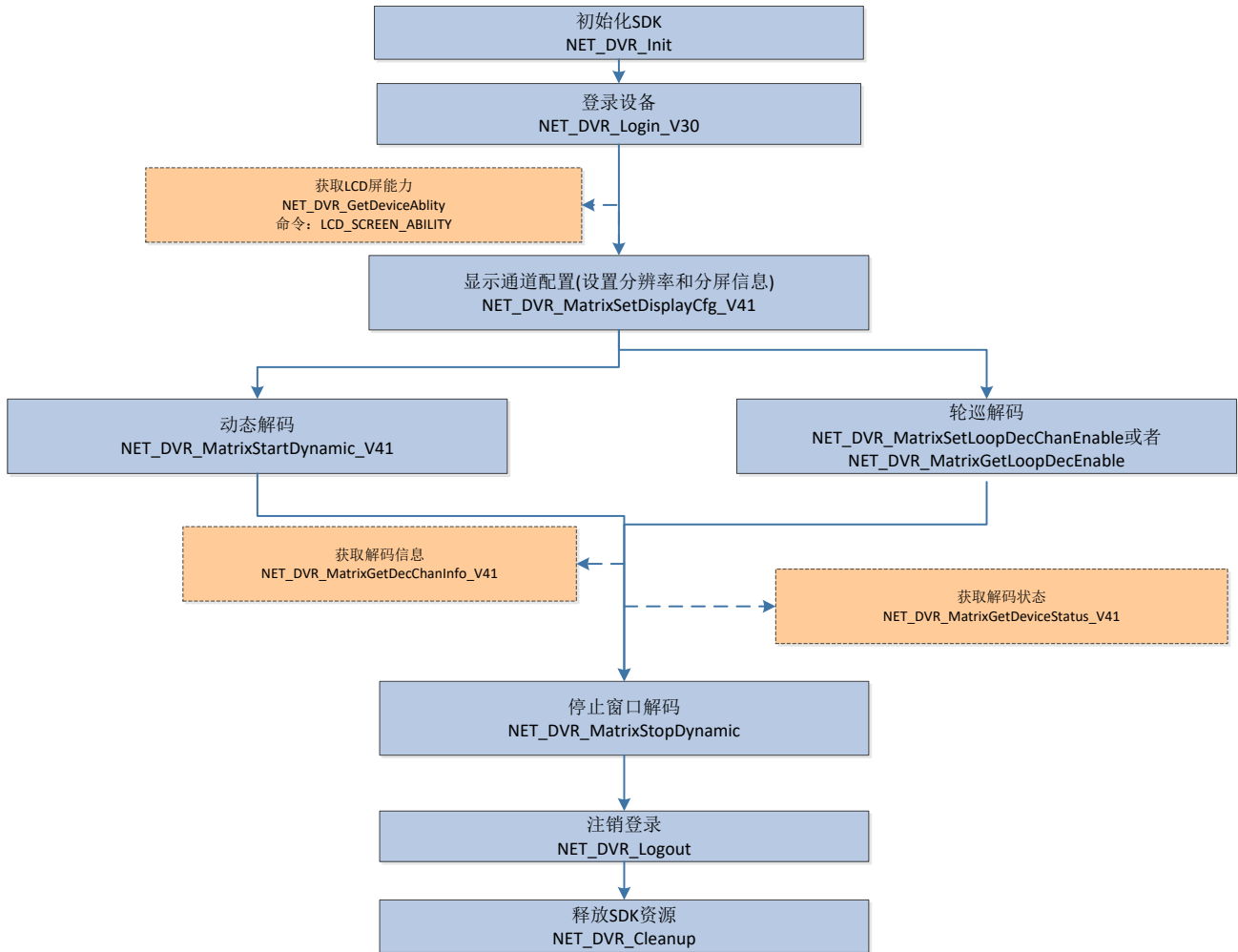
原参数。

- 设置屏幕参数，通过 [NET DVR StartRemoteConfig](#)(命令：NET_DVR_SET_SCREEN_PARAM)，可以设置输入源、风扇工作模式、VGA 信号、屏幕菜单、显示效果、色温参数、ADC 参数、屏幕边缘参数、背光参数、画中画参数、去雾参数、屏幕工作状态、开机 logo 参数以及 LCD 音频参数等。
- 设置屏幕拼接，通过 [NET DVR StartRemoteConfig](#)(命令：NET_DVR_SCREEN_SPLICE_SET)设置屏幕拼接；
- 通过布防 [NET DVR SetupAlarmChan V41](#) 或者监听 [NET DVR StartListen V30](#) 的方式获取 LCD 报警，报警类型为 COMM_ALARM_LCD；
- 屏幕服务器是一个软件，pc 上通过串口环通连接 LCD 屏幕。屏幕服务器将收到的网络命令，转换成串口命令从而对屏幕进行控制。且屏幕服务器只能配置 LCD 屏幕的一些参数，不能控制屏幕输出图像；可以通过 SDK 控制屏幕服务器，通过屏幕服务器转换为串口命令来控制屏幕。
- LCD 屏幕拼接和拼控器里的拼接是有区别的。LCD 屏幕的拼接输入信号是同源的，屏幕根据自身的相对位置，显示了输入信号的一部分，这个是需要屏幕支持的，属于 LCD 屏幕的功能；多屏控制器的拼接是在信号输入时就对信号进行了切分，屏幕完全根据输入信号显示，不做特殊处理，是多屏控制器的功能。

[调用实例代码](#)

3.2.2 带解码功能的 LCD 屏

图 3.2 带解码功能的 LCD 屏接口调用流程

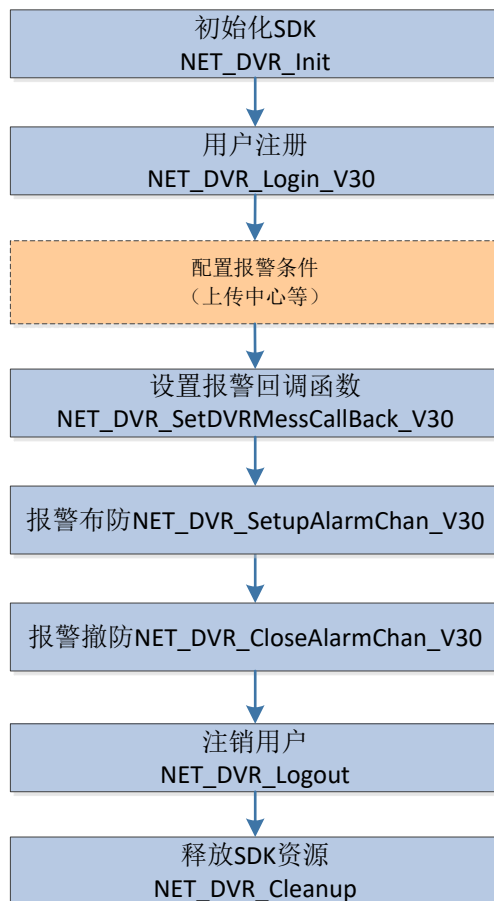


- 带解码板的 LCD 屏相当于一个单路解码器，可以单独进行登录，可通过 sdk 接口控制屏幕进行屏幕分割、解码上墙，具体步骤参考解码器解码部分，参考上述流程图中的步骤；
- 完成参数配置后即可启动窗口上墙显示：
 - 动态解码，相关接口：[NET_DVR_MatrixStartDynamic V41](#)、[NET_DVR_MatrixStopDynamic](#)，切换解码前端源时不需要停止解码，修改动态解码参数（[NET_DVR_PU_STREAM_CFG V41](#)）再次调用动态解码接口即可（多次调用动态解码接口以最后一次的为准）。网络信号源、本地信号源选择均在动态解码接口的参数 [NET_DVR_PU_STREAM_CFG V41](#) 中设置。
- 场景操作：
 - 通过 [NET_DVR_SetDeviceConfig](#)（命令：NET_DVR_SCREEN_SCENE_CONTROL）可以进行场景保存、切换、初始化操作。
 - 保存场景后，可以通过 [NET_DVR_SetDeviceConfig](#)（命令：NET_DVR_WALLSCENEPARAM_SET）设置场景参数，比如修改场景名称等。
 - 通过 [NET_DVR_GetDeviceConfig](#)（命令：NET_DVR_GET_CURRENT_SCENE）可以获取当前正在使用的场景模式。
- 状态获取：通过 [NET_DVR_MatrixGetDeviceStatus V41](#)（命令：NET_DVR_GET_WIN_DEC_INFO）可以获取各窗口当前的解码信息（[NET_DVR_MATRIX_DEC_CHAN_INFO V41](#)），包括窗口关联的信号源、解码状态等。

3.3 报警模块流程

3.3.1 报警布防

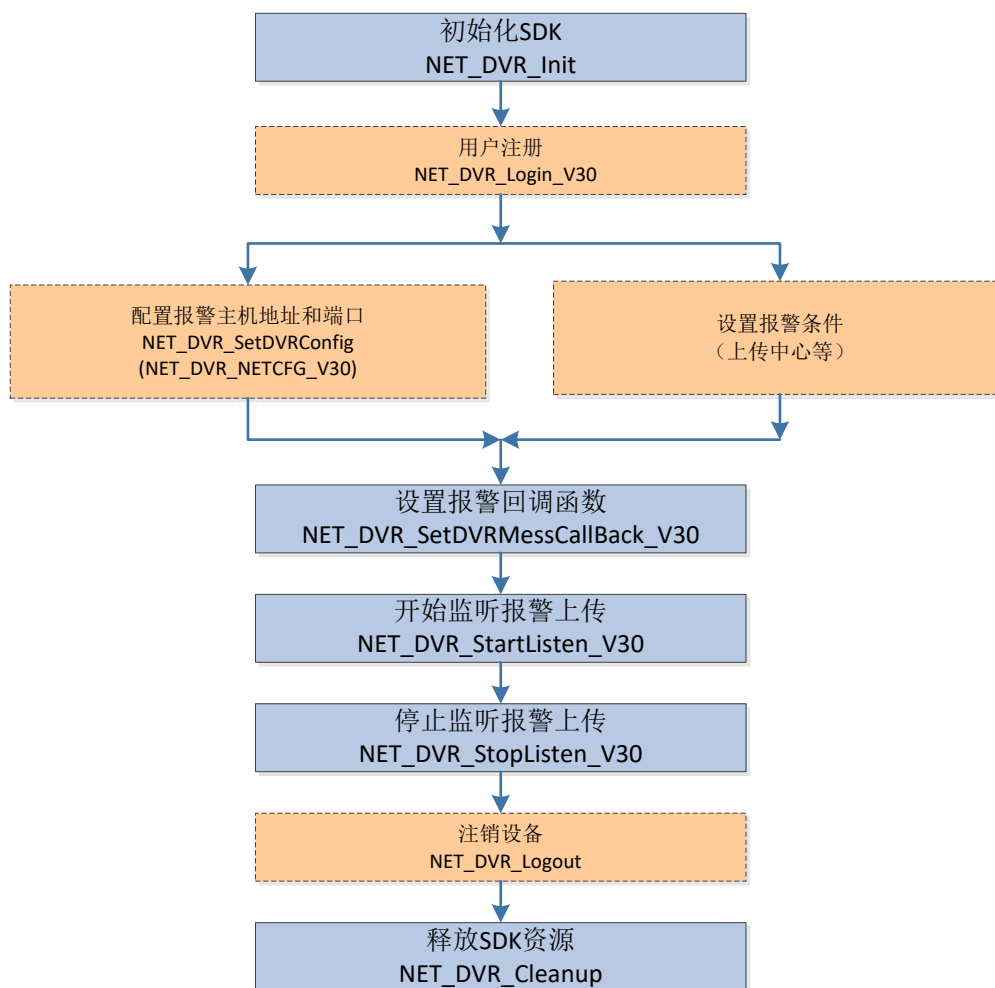
图 3.3 报警布防调用流程



- “布防”报警方式是指 SDK 主动连接设备，并发起报警上传命令，设备发生报警立即发送给 SDK。
- “布防”方式需要先进行用户注册（[NET DVR Login V30](#)），然后需要设置报警回调函数（[NET DVR SetDVRMessageCallBack V30](#)）和对设备进行布防（[NET DVR SetupAlarmChan V41](#)）。报警信息在设置的报警回调函数中获取。整个报警上传过程结束后还需要调用撤防接口等操作，释放相关资源。
- 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（须设置上传中心）的配置。比如，报警输入参数配置，相关的接口：[NET DVR GetDVRConfig](#)（命令：[NET_DVR_GET_ALARMINCFG_V40](#)）、[NET DVR SetDVRConfig](#)（命令：[NET_DVR_SET_ALARMINCFG_V40](#)）。相关参数如果已经配置完成，那么虚线框部分可以省略。

3.3.2 报警监听

图 3.4 报警监听调用流程



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

3.4 透明通道

图 3.5 透明通道调用流程



- 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 RS232 串口参数 [NET_DVR_RS232CFG_V30](#) 中的 dwWorkMode 值为透明通道。如果将 485 串口作为透明通道，则此步骤可以省略。
- 透明通道接口调用步骤如下：调用 [NET_DVR_SerialStart_V40](#) 建立透明通道；调用 [NET_DVR_SerialSend](#) 发送数据；过程结束后断开透明通道连接 [NET_DVR_SerialStop](#)。

4 函数调用实例

4.1 LCD 屏服务器拼接示例代码

[相关模块流程图](#)

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

#define XML_BUF 3*1024*1024
#define SPLICE_NUM 10 //拼接屏总数
    LONG        m_lUserID;
    HANDLE      m_hThread;
    HANDLE      m_lHandle;
    LONG        m_lCfgHandle;
    BOOL        m_bExitThread;
    DWORD      m_dwCurrentScreenNo;
    DWORD      m_dwCfgNum;
    DWORD      m_dwStatusListIndex[SCREEN_NUM]; //索引配置状态列表
    DWORD      m_dwStatusListPos[SCREEN_NUM]; //位置配置状态列表
    DWORD      m_dwIndex[SCREEN_NUM]; //屏幕索引号
    DWORD      m_dwCfgNo;
    int        m_iConfigType;
    int        m_iDeviceIndex;
    int        m_iCurSel;
    DWORD      m_dwSetCount;
    DWORD      m_dwStatus[LCD_SCENE_NUM];
    DWORD      m_dwCopyScene;
    CString    m_csSceneName;
    DWORD      m_dwSceneNum;
    DWORD      m_dwSceneNumOperate;
    DWORD      m_dwWallNo;
    BOOL        m_bEnable;
    DWORD      m_dwWallNoOperate;
    DWORD      m_dwSceneNo;

    NET_DVR_MSC_SCREEN_PARAM_COND m_struCond;
    NET_DVR_MSC_SCREEN_REMOTE_CFG m_struSet;
    LPNET_DVR_MSC_SCREEN_PARAM_CFG m_lpScreenParam;
```

```

NET_DVR_SCREEN_INDEX_CFG          m_struScreenIndex[SCREEN_NUM]; //屏幕索引参数
NET_DVR_VIDEOWALLDISPLAYPOSITION  m_struPosition[SCREEN_NUM]; //屏幕位置参数
NET_DVR_MSC_SPLICE_CFG            m_struSpliceCond[SPLICE_NUM];
NET_DVR_MSC_SPLICE_CFG            m_struSpliceParam[SPLICE_NUM];
NET_DVR_VIDEO_WALL_INFO           m_struSceneNo[LCD_SCENE_NUM];
NET_DVR_WALLSCENECFG             m_struSceneName[LCD_SCENE_NUM];
NET_DVR_SCENE_CONTROL_INFO        m_struSceneControlInfo;

```

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

```

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

```

//拼接状态线程，以 m_struSpliceCond[0]为例

```
UINT LcdSpliceWaitThread(LPVOID pParam)
```

```

{
    CLCDDemoDlg *ptr = (CLCDDemoDlg *)pParam;

    if (ptr->m_iConfigType == 0) //参数配置类型，0-设置，1-获取
    {
        if (!INET_DVR_SendRemoteConfig(ptr->m_ICfgHandle, ENUM_MSC_SEND_DATA, (char *)&ptr->m_struSpliceParam[0],
sizeof(NET_DVR_MSC_SPLICE_CFG)))
        {
            NET_DVR_StopRemoteConfig(ptr->m_ICfgHandle);
            OutputDebugString("ENUM_MSC_SEND_DATA fail,LcdSpliceWaitThread exit 1!");
            ptr->m_IHandle = NULL;
            return 0;
        }
    }
    else//参数配置类型，0-设置，1-获取
    {
        if (!INET_DVR_SendRemoteConfig(ptr->m_ICfgHandle, ENUM_MSC_SEND_DATA, (char *)&ptr->m_struSpliceCond[0],
sizeof(NET_DVR_MSC_SPLICE_CFG)))
        {
            OutputDebugString("ENUM_MSC_SEND_DATA fail,LcdSpliceWaitThread exit 2!");
            NET_DVR_StopRemoteConfig(ptr->m_ICfgHandle);
            ptr->m_IHandle = NULL;

```

```

        return 0;
    }
    Sleep(100);
}

while (!ptr->m_bExitThread)
{
    Sleep(500);
}

if (ptr->m_ICfgHandle != -1)
{
    NET_DVR_StopRemoteConfig(ptr->m_ICfgHandle);
    ptr->m_ICfgHandle = -1;
}

::PostMessage(ptr->m_hWnd, WM_PROC_LCD_SPLICE, NULL, NULL);
OutputDebugString("LcdSpliceWaitThread exit 3!");
return 0;
}

//拼接线程
void CALLBACK LcdSpliceCallBack(DWORD dwType, void* lpBuffer, DWORD dwBufLen, void* pUserData)
{
    CLCDDemoDlg *ptr = (CLCDDemoDlg *)pUserData;
    DWORD dwStatus;
    char szLan[128] = {0};
    NET_DVR_MSC_SPLICE_CFG *lpData = NULL;

    switch (dwType)
    {
    case NET_SDK_CALLBACK_TYPE_STATUS:
        dwStatus = *((DWORD *)lpBuffer);
        if (dwStatus == NET_SDK_CALLBACK_STATUS_SUCCESS)
        {
            sprintf(szLan, "Splice %d success", ptr->m_dwCfgNo);
            OutputDebugString(szLan);
        }
        else if (dwStatus == NET_SDK_CALLBACK_STATUS_PROCESSING)
        {
            printf(szLan, "Splice %d processing", ptr->m_dwCfgNo);
            OutputDebugString(szLan);
        }
        else if (dwStatus == NET_SDK_CALLBACK_STATUS_FAILED)

```

```

    {
        sprintf(szLan, "Splice %d fail", ptr->m_dwCfgNo);
        OutputDebugString(szLan);
        ptr->m_dwCfgNo++;
    }
    else if (dwStatus == NET_SDK_CALLBACK_STATUS_EXCEPTION)
    {
        sprintf(szLan, "Config exception");
        OutputDebugString(szLan);
        ptr->m_bExitThread = TRUE;
    }
    break;
case NET_SDK_CALLBACK_TYPE_DATA:
    lpData = (NET_DVR_MSC_SPLICE_CFG *)lpBuffer;
    memcpy(&ptr->m_struSpliceParam[ptr->m_dwCfgNo - 1], lpData, sizeof(NET_DVR_MSC_SPLICE_CFG));
    if (ptr->m_dwCfgNo == SPLICE_NUM)
    {
        ptr->m_bExitThread = TRUE;
    }
    sprintf(szLan, "Splice %d success", ptr->m_dwCfgNo);
    OutputDebugString(szLan);
    ptr->m_dwCfgNo++;
    break;
default:
    sprintf(szLan, "SPLICE_SET CallBack unknow status %d", dwType);
    OutputDebugString(szLan);
    return;
}

sprintf(szLan, "Splice config over");
OutputDebugString(szLan);
ptr->m_bExitThread = TRUE;
}

void main()
{
    DWORD dwRet = 0;
    m_bExitThread = FALSE;
    m_lUserID = -1;
    m_hThread = NULL;
    m_lCfgHandle = -1;
    m_dwCurrentScreenNo = 0;
    m_dwCfgNum = 0;
    m_BPressGetParamBut = FALSE;
    m_dwCfgNo = 0;
    m_lCfgHandle = -1;
}

```



```
m_IHandle = NULL;
m_dwSceneNum = 0;
m_dwSceneNumOperate = 0;
m_dwWallNo = 1;
m_bEnable = FALSE;
m_dwWallNoOperate = 0;
m_dwSceneNo = 0;
m_lpScreenParam = NULL;
DWORD dwError = 0;
DWORD dwThreadId = 0;

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

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

//-----
// 注册设备
LONG IUserID;
NET_DVR_DEVICEINFO_V30 struDeviceInfo;
IUserID = NET_DVR_Login_V30("172.9.31.205", 8000, "admin", "12345", &struDeviceInfo);
if (IUserID < 0)
{
    printf("NET_DVR_Login_V30 failed, err: %d\n", NET_DVR_GetLastError());
    NET_DVR_Cleanup();
    return;
}

//获取电视墙能力集，包括基准坐标、支持的屏幕/窗口/画面分割个数
char* m_pXMLBuf = NULL;
m_pXMLBuf = new char[XML_BUF];
char szLan[128] = {0};
if (!NET_DVR_GetDeviceAbility(m_IUserID, LCD_SCREEN_ABILITY, NULL, 0, m_pXMLBuf, XML_BUF))
{
    dwError = NET_DVR_GetLastError();
    sprintf(szLan, "LCD_SCREEN_ABILITY FAILED,%d!", dwError);
    AfxMessageBox(szLan);
}
```

```

DWORD dwStatus      = 0;
DWORD dwOutputNum = 0;
memset(m_struScreenIndex, 0, sizeof(m_struScreenIndex));
memset(m_struPosition, 0, sizeof(m_struPosition));
memset(m_dwStatusListIndex, 0, sizeof(m_dwStatusListIndex));
memset(m_dwStatusListPos, 0, sizeof(m_dwStatusListPos));

for (int i = 0; i < SCREEN_NUM; i++)
{
    m_dwIndex[i] = (i + 1) | 0x01000000;
    m_struScreenIndex[i].dwSize = sizeof(m_struScreenIndex[i]);
    m_struPosition[i].dwSize = sizeof(m_struPosition[i]);
}

//获取屏幕
if (!NET_DVR_GetDeviceConfig(m_UserID, NET_DVR_SCREEN_INDEX_GET, SCREEN_NUM, m_dwIndex, sizeof(m_dwIndex),
m_dwStatusListIndex, m_struScreenIndex, sizeof(m_struScreenIndex)))
{
    dwError = NET_DVR_GetLastError();
    sprintf(szLan, "NET_DVR_SCREEN_INDEX_GET FAILED,%d!", dwError);
    AfxMessageBox(szLan);
}

//获取屏幕位置
if (!NET_DVR_GetDeviceConfig(m_UserID, NET_DVR_GET_SCREEN_POSITION, SCREEN_NUM, m_dwIndex, sizeof(m_dwIndex),
m_dwStatusListPos, m_struPosition, sizeof(m_struPosition)))
{
    dwError = NET_DVR_GetLastError();
    sprintf(szLan, "NET_DVR_GET_SCREEN_POSITION FAILED,%d!", dwError);
    AfxMessageBox(szLan);
}

//添加屏幕，以 m_struScreenIndex[0]为例
memset(&m_struScreenIndex[0], 0, sizeof(m_struScreenIndex[0]));
m_struScreenIndex[0].dwSize = sizeof(m_struScreenIndex[0]);
m_struScreenIndex[0].byEnbale = TRUE;
m_struScreenIndex[0].byDevAddType = 1; //按 IP 地址(服务器和屏幕使用网络通信)
if (m_struScreenIndex[0].byDevAddType == 1) //以 IP 地址为例
{
    memcpy(m_struScreenIndex[0].struScreenBaseInfo.struIPInfo.struIPAddr.slIPv4, m_csDeviceIp, min(16,
m_csDeviceIp.GetLength()));
    m_struScreenIndex[0].struScreenBaseInfo.struIPInfo.wPort = m_nDevPort;
    memcpy(m_struScreenIndex[0].struScreenBaseInfo.struIPInfo.byUserName, m_csUser, min(NAME_LEN,
m_csUser.GetLength()));
}

```

```

        memcpy(m_struScreenIndex[0].struScreenBaseInfo.struIPInfo.byPassWord, m_csPWD, min(PASSWD_LEN,
m_csPWD.GetLength()));
    }

    m_struScreenIndex[0].dwSize = sizeof(NET_DVR_SCREEN_INDEX_CFG);
    //设置屏幕索引
    if (!INET_DVR_SetDeviceConfig(m_IUserID, NET_DVR_SCREEN_INDEX_SET, 1, &m_dwIndex[0], sizeof(m_dwIndex[0]), \
        &m_dwStatusListIndex[0], &m_struScreenIndex[0], sizeof(m_struScreenIndex[0])))
    {
        dwError = NET_DVR_GetLastError();
        sprintf(szLan, "NET_DVR_SCREEN_INDEX_SET FAILED,%d!", dwError);
        AfxMessageBox(szLan);
    }

    //设置屏幕位置参数
    m_struPosition[0].byEnable = 1;
    m_struPosition[0].dwVideoWallNo = 1 << 24 | 0;
    m_struPosition[0].struRectCfg.dwXCoordinate = 0;
    m_struPosition[0].struRectCfg.dwYCoordinate = 0;
    m_struPosition[0].dwSize = sizeof(NET_DVR_VIDEOWALLDISPLAYPOSITION);

    if (!INET_DVR_SetDeviceConfig(m_IUserID, NET_DVR_SET_SCREEN_POSITION, 1, &m_dwIndex[0], sizeof(m_dwIndex[0]), \
        &m_dwStatusListPos[0], &m_struPosition[0], sizeof(m_struPosition[0])))
    {
        dwError = NET_DVR_GetLastError();
        sprintf(szLan, "NET_DVR_SET_SCREEN_POSITION FAILED,%d!", dwError);
        AfxMessageBox(szLan);
    }

    //设置屏幕控制参数
    NET_DVR_SCREEN_CONTROL_V41 m_struScreenCtrl;
    memset(&m_struScreenCtrl, 0, sizeof(m_struScreenCtrl));
    m_struScreenCtrl.dwSize = sizeof(m_struScreenCtrl);
    m_struScreenCtrl.byProtocol = 0xff;
    m_struScreenCtrl.byWallNo = 1;
    m_struScreenCtrl.struRect.dwXCoordinate = 0;
    m_struScreenCtrl.struRect.dwYCoordinate = 0;
    m_struScreenCtrl.struRect.dwWidth = 1920;
    m_struScreenCtrl.struRect.dwHeight = 1920;
    m_struScreenCtrl.dwCommand = 1;

    //打开屏幕控制为例
    if (!INET_DVR_RemoteControl(m_IUserID, NET_DVR_CONTROL_SCREEN, &m_struScreenCtrl, sizeof(m_struScreenCtrl)))
    {

```

```

        dwError = NET_DVR_GetLastError();
        sprintf(szLan,"NET_DVR_CONTROL_SCREEN FAILED,%d!",dwError);
        AfxMessageBox(szLan);
    }

    memset(&m_struCond, 0, sizeof(m_struCond));
    memset(&m_struSet, 0, sizeof(m_struSet));
    memset(&m_lpScreenParam, 0, sizeof(m_lpScreenParam));
    memset(&m_struSet, 0, sizeof(m_struSet));
    m_struSet.dwSize = sizeof(m_struSet);
    m_struCond.byParamType = 1;
    m_struCond.byWallNo = 1;
    m_struCond.struRectCfg.dwXCoordinate = 0;
    m_struCond.struRectCfg.dwYCoordinate = 0;
    m_struCond.struRectCfg.dwWidth = 1920;
    m_struCond.struRectCfg.dwHeight = 1920;

    m_struSet.struScreenParam.byParamType = 1;
    m_struSet.byWallNo = 1;
    m_struSet.struRectCfg.dwXCoordinate = 0;
    m_struSet.struRectCfg.dwYCoordinate = 0;
    m_struSet.struRectCfg.dwWidth = 1920;
    m_struSet.struRectCfg.dwHeight = 1920;
    m_struCond.dwSize = sizeof(m_struCond);
    //获取屏幕参数
    m_ICfgHandle = NET_DVR_StartRemoteConfig(m_IUserID, NET_DVR_GET_SCREEN_PARAM, &m_struCond, sizeof(m_struCond),
ScreenConfigCallBack, this);
    if (m_ICfgHandle == -1)
    {
        sprintf(szLan,"NET_DVR_SET_SCREEN_PARAM failed, err: %d\n", NET_DVR_GetLastError());
        OutputDebugString(szLan);
    }
    //输入源参数、VGA、蓝屏
    m_struSet.struScreenParam.dwSize = sizeof(m_struSet.struScreenParam);
    m_struSet.struScreenParam.struParam.struInterfaceCfg.dwSize =
sizeof(m_struSet.struScreenParam.struParam.struInterfaceCfg);
    m_struSet.struScreenParam.struParam.struInterfaceCfg.byInterfaceType = 0;
    m_struSet.struScreenParam.struParam.struInterfaceCfg.byNoSignalPic = 1;
    //设置屏幕参数
    m_ICfgHandle = NET_DVR_StartRemoteConfig(m_IUserID, NET_DVR_SET_SCREEN_PARAM, &m_struSet, sizeof(m_struSet),
ScreenConfigCallBack, this);
    if (m_ICfgHandle == -1)
    {
        sprintf(szLan,"NET_DVR_SET_SCREEN_PARAM failed, err: %d\n", NET_DVR_GetLastError());

```

```

        OutputDebugString(szLan);
    }
    else
    {
        OutputDebugString("NET_DVR_SET_SCREEN_PARAM SUCC");
        DWORD dwCfgNum = (m_struSet.struRectCfg.dwWidth/1920) * (m_struSet.struRectCfg.dwHeight/1920);
        DWORD dwCurrentScreenNo = 1;
        m_hThread = CreateThread(NULL, 0, LPTHREAD_START_ROUTINE((unsigned int) LcdParamCfgWaitThread), this, 0,
&dwThreadId);
        if (m_hThread == NULL)
        {
            OutputDebugString("Create wait thread fail");
            NET_DVR_StopRemoteConfig(m_ICfgHandle);
        }
    }

//设置 LCD 拼接
    m_struSpliceCond[0].dwSize = sizeof(m_struSpliceCond[0]);
    m_struSpliceCond[0].bySpliceNo = 1;
    m_struSpliceCond[0].byWallNo = 1;
    m_struSpliceParam[0].dwSize = sizeof(m_struSpliceParam[0]);
    m_dwCfgNo = 1;
    m_bExitThread = FALSE;
    m_iConfigType = 0;
    m_ICfgHandle = NET_DVR_StartRemoteConfig(m_IUserID, NET_DVR_SCREEN_SPLICE_SET, NULL, 0, LcdSpliceCallBack, this);
    if (m_ICfgHandle == -1)
    {
        dwError = NET_DVR_GetLastError();
        sprintf(szLan, "NET_DVR_SCREEN_SPLICE_SET FAILED,%d!", dwError);
        AfxMessageBox(szLan);
    }
    else
    {
        printf("NET_DVR_SCREEN_SPLICE_SET success");
        m_IHandle = CreateThread(NULL, 0, LPTHREAD_START_ROUTINE(LcdSpliceWaitThread), this, 0, &dwThreadId);
        if (m_IHandle == NULL)
        {
            printf("Create wait thread fail");
            NET_DVR_StopRemoteConfig(m_ICfgHandle);
        }
    }

//关闭长连接
    if (m_ICfgHandle != -1)

```

```
{
    NET_DVR_StopRemoteConfig(m_ICfgHandle);
    m_ICfgHandle = -1;
}
if (NULL != m_lpScreenParam)
{
    delete []m_lpScreenParam;
    m_lpScreenParam = NULL;
}

//销毁 LCD 屏幕拼接
if (m_IHandle != NULL)
{
    m_bExitThread = TRUE;
    WaitForSingleObject(m_IHandle, INFINITE);
    CloseHandle(m_IHandle);
    m_IHandle = NULL;
}

//销毁屏幕配置参数线程
if (m_hThread != NULL)
{
    m_bExitThread = TRUE;
    WaitForSingleObject(m_hThread, INFINITE);
    CloseHandle(m_hThread);
    m_hThread = NULL;
}

//设备登出
if (m_IUserID != -1)
{
    NET_DVR_Logout(m_IUserID);
    m_IUserID = -1;
}

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

return;
}
```

5 函数说明

5.1 SDK 初始化

5.1.1 初始化 SDK `NET_DVR_Init`

函 数: `BOOL NET_DVR_Init()`

参 数: 无

返回值: `TRUE` 表示成功, `FALSE` 表示失败。

说 明: 调用设备网络 SDK 其他函数的前提

[返回目录](#)

5.1.2 释放 SDK 资源 `NET_DVR_Cleanup`

函 数: `BOOL NET_DVR_Cleanup()`

参 数: 无

返回值: `TRUE` 表示成功, `FALSE` 表示失败。

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

[返回目录](#)

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

说 明: **SDK 默认建立连接的超时时间为 3 秒。** SDK4.0 及以后版本中当设置的超时时间超过或低于限制的值时接口不返回失败, 将取最接近的上下限限制值作为实际的超时时间。

[返回目录](#)

5.1.4 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, 参数默认值为 1

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

过错误码判断出错原因。

说明： 该接口可以同时控制预览、透明通道和布防的重连功能。不调用该接口时，SDK 默认启动预览、透明通道和布防的重连功能，重连时间间隔为 5 秒。

[返回目录](#)

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

函数： BOOL NET_DVR_SetRecvTimeOut(DWORD nRecvTimeOut)

参数： [in] nRecvTimeOut 接收超时时间，单位毫秒，默认为 5000，最小为 3000 毫秒

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

说明： 该接口用于设置接收超时时间，例如预览接收实时流数据、回放下载接收录像数据、报警接收报警信息等接收超时时间。

[返回目录](#)

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

函数： BOOL NET_DVR_GetLocalIP(char strIP[16][16], DWORD *pValidNum, BOOL *pEnableBind)

参数： [out] strIP 存放 IP 的缓冲区，不能为空

[out] pValidNum 所有有效 IP 的数量

[out] pEnableBind 是否绑定

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

说明： 该接口获取客户端本地多网卡的所有 IP 地址，可以通过接口 [NET_DVR_SetValidIP](#) 选择要使用的 IP 地址。

[返回目录](#)

5.1.7 设置 IP 绑定 **NET_DVR_SetValidIP**

函数： BOOL NET_DVR_SetValidIP(DWORD dwIPIndex, BOOL bEnableBind)

参数： [in] dwIPIndex 选择使用的 IP 下标，由 [NET_DVR_GetLocalIP](#) 获取

[in] bEnableBind 是否绑定

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

说明：

[返回目录](#)

5.1.8 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 地址或域名
- [in]wServerPort 解析服务器的端口号, IP Server 解析服务器端口号为 7071, HiDDNS 服务器的端口号为 80
- [in]sDVRName 设备名称或者设备域名
- [in]wDVRNameLen 设备名称(域名)的长度
- [in]sDVRSerialNumber 设备的序列号
- [in]wDVRSerialLen 设备序列号的长度
- [out]sGetIP 获取到的设备 IP 地址指针
- [out]dwPort 获取到的设备端口号指针

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

说 明: 该接口中的设备名称和设备序列号不能同时为空。支持的解析服务器有 IP Server 和 hiDDNS。通过设备域名或者序列号解析出设备当前 IP 地址和端口, 然后调用 [NET_DVR_Login_V30](#) 登录设备。

[返回目录](#)

5.2 异常消息回调

5.2.1 注册接收异常、重连等消息回调函数 [NET_DVR_SetExceptionCallback_V30](#)

函 数: **Windows 系统下:**

BOOL NET_DVR_SetExceptionCallback_V30 (UINT nMessage, HWND hWnd, fExceptionCallBack cbExceptionCallBack, void* pUser)

Linux 系统下:

BOOL NET_DVR_SetExceptionCallback_V30(UINT nMessage, void* hWnd, fExceptionCallBack cbExceptionCallBack, void* pUser)

参 数:

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

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

- [out]dwType 异常或重连等消息的类型, 详见表 5.1
- [out]IUserID 登录 ID
- [out]IHandle 出现异常的相应类型的句柄
- [out]pUser 用户数据

表 5.1 异常消息类型

dwType 宏定义	宏定义值	含义
EXCEPTION_EXCHANGE	0x8000	用户交互时异常(注册心跳超时, 心跳间隔为 2 分钟)
EXCEPTION_AUDIOEXCHANGE	0x8001	语音对讲异常
EXCEPTION_ALARM	0x8002	报警异常

EXCEPTION_PREVIEW	0x8003	网络预览异常
EXCEPTION_SERIAL	0x8004	透明通道异常
EXCEPTION_RECONNECT	0x8005	预览时重连
EXCEPTION_ALARMRECONNECT	0x8006	报警时重连
EXCEPTION_SERIALRECONNECT	0x8007	透明通道重连
SERIAL_RECONNECTSUCESS	0x8008	透明通道重连成功
EXCEPTION_PLAYBACK	0x8010	回放异常
EXCEPTION_DISKFORMAT	0x8011	硬盘格式化
EXCEPTION_PASSIVEDECODE	0x8012	被动解码异常
PREVIEW_RECONNECTSUCESS	0x8015	预览时重连成功
ALARM_RECONNECTSUCESS	0x8016	报警时重连成功
RESUME_EXCHANGE	0x8017	用户交互恢复
EXCEPTION_MAX_ALARM_INFO	0x8022	报警信息缓存已达上限
EXCEPTION_LOST_ALARM	0x8023	报警丢失

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

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

如果此结构是以回调方式反馈异常消息, 那么应用程序中的异常回调函数实现如下, 该函数中的参数 dwType 表示异常消息类型 (见表 5.1); 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_EXCHANGE:
            sprintf(tempbuf,"用户交互时异常!!!");
            TRACE("%s",tempbuf);
            break;

        case EXCEPTION_ALARM:
            sprintf(tempbuf,"报警时网络异常!!!");
            TRACE("%s",tempbuf);
            //TODO: 关闭报警上传
    }
}
```

```

        break;
    case EXCEPTION_ALARMRECONNECT:
        sprintf(tempbuf,"报警重连!!!");
        break;
    case ALARM_RECONNECTSUCCESS:
        sprintf(tempbuf,"报警重连成功!!!");
        break;
    default:
        break;
}
}

```

[返回目录](#)

5.3 SDK 日志和信息

5.3.1 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](#)

5.3.2 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，详见：[NET_DVR_SDKSTATE](#)

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

说明:

[返回目录](#)

5.3.3 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，详见：[NET_DVR_SDKABL](#)

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

说明:

[返回目录](#)

5.3.4 获取 SDK 本地参数 `NET_DVR_GetSDKLocalCfg`

函数: `BOOL NET_DVR_GetSDKLocalCfg(NET_SDK_LOCAL_CFG_TYPE enumType, void *lpOutBuff)`
 参数: [in] enumType 配置类型, 不同的取值对应不同的 SDK 参数, 详见表 5.2
 [out] lpOutBuff 输出参数, 不同的配置类型, 输出参数对应不同的结构, 详见表 5.2
 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

表 5.2 本地参数类型

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

[返回目录](#)

5.3.5 设置 SDK 本地参数 `NET_DVR_SetSDKLocalCfg`

函数: `BOOL NET_DVR_SetSDKLocalCfg(NET_SDK_LOCAL_CFG_TYPE enumType, void* const lpInBuff)`
 参数: [in] enumType 配置类型, 不同的取值对应不同的 SDK 参数, 详见表 5.3
 [in] lpInBuff 输入参数, 不同的配置类型, 输出参数对应不同的结构, 详见表 5.3
 返回值: TRUE 表示成功, FALSE 表示失败。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码, 通过错误码判断出错原因。

说明:

表 5.3 本地参数类型

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

NET_SDK_LOCAL_CFG_TYPE_CHAR_ENCODE	13	配置字符编码相关处理 回调	NET_DVR_LOCAL_BYTE_ENCODE_CONVERT
------------------------------------	----	------------------	---

[返回目录](#)

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

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

参数: [in]bLogEnable 日志的等级（默认为 0）：
 0-表示关闭日志
 1-表示只输出 ERROR 错误日志
 2-输出 ERROR 错误信息和 DEBUG 调试信息
 3-输出 ERROR 错误信息、DEBUG 调试信息和 INFO 普通信息等所有信息

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

[in]bAutoDel 是否删除超出的文件数，默认值为 TRUE

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

说明: 日志文件路径必须是绝对路径，且以"\"结尾，例如"C:\\SdkLog\\"，建议用户先手动创建文件。若未指定文件路径，则采用默认路径"C:\\SdkLog\\"。可多次调用该接口创建新的日志文件，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。通过错误码可以获取错误信息，然后判断出错原因。

[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： 通过接口 `NET_DVR_GetErrorMsg` 直接返回错误信息，直接通过错误信息判断错误原因。

[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] Sdvrrip	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，详见： NET_DVR_DEVICEINFO_V30

返回值： -1 表示失败，其他值表示返回的用户 ID 值。该用户 ID 具有唯一性，后续对设备的操作都需要通过此 ID 实现。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说明： SDK 注册设备新增支持静态域名的方式，即可设置 `sDVRIP="test.vicp.net"`。
一台设备同时最多允许 128 个用户注册。
SDK 最多支持 2048 个注册，返回 `UserID` 的取值范围为 0~2047。

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

说明： 建议使用此接口实现注销功能。

[返回目录](#)

5.6 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] IUserID	登录主控板, NET DVR Login V30 的返回值
	[in] dwAbilityType	能力类型, 具体定义见表 5.4
	[in] pInBuf	输入缓冲区指针, 参见表 5.5
	[in] dwInLength	输入缓冲区的长度
	[out] pOutBuf	输出缓冲区指针, 参见表 5.5
	[in] dwOutLength	接收数据的缓冲区的长度

表 5.4 设备能力集类型

dwAbilityType 宏定义	宏定义值	含义
DEVICE_ALL_ABILITY	0x000	设备全部能力 (保留)
DEVICE_SOFTWARE_ABILITY	0x001	设备软硬件能力
DEVICE_NETWORK_ABILITY	0x002	设备无线网络能力
DEVICE_ENCODE_CURRENT	0x004	设备当前编码能力
DEVICE_ENCODE_ALL_ABILITY_V20	0x008	设备所有编码能力
IPC_FRONT_PARAMETER_V20	0x009	设备前端参数
VCA_DEV_ABILITY	0x100	智能设备的能力
VCA_CHAN_ABILITY	0x110	智能通道的能力
MATRIXDECODER_ABILITY	0x200	多路解码器显示、解码能力
DECODER_ABILITY	0x261	解码器 XML 能力集
SNAPCAMERA_ABILITY	0x300	智能交通摄像机能力集
LCD_SCREEN_ABILITY	0x400	LCD 屏幕能力

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

说 明: 获取各种能力类型时, 输入参数 pInBuf 和输出参数 pOutBuf 的格式定义, 如表 5.5 所示。

表 5.5 设备能力集

能力类型宏定义	能力类型说明	pInBuf	pOutBuf
DEVICE_SOFTWARE_ABILITY	获取设备软硬件能力	无	设备软硬件能力 XML 描述 (BasicCapability)
DEVICE_NETWORK_ABILITY	获取无线设备网络能力	无	设备无线网络能力 XML 描述 (NetworkSetting)
DEVICE_ENCODE_CURRENT	获取设备当前编码能力	当前编码能力输入描述	设备当前编码能力 XML 描述
DEVICE_ENCODE_ALL_ABILITY_V20	获取设备所有编码能力	编码能力获取输入描述	设备所有编码能力 XML 描述 (AudioVideoCompressInfo)
IPC_FRONT_PARAMETER_V20	获取设备前端参数	前端参数能力获取输入描述	设备前端参数 XML 描述 (CAMERAPARA)
DEVICE_SERIAL_ABILITY	获取设备 RS232 和 RS485 串口能力	串口能力输入描述	设备串口能力 XML 描述 (SerialAbility)
VCA_DEV_ABILITY	获取智能设备的能力	无	NET_VCA_DEV_ABILITY

VCA_CHAN_ABILITY	获取智能通道能力	NET_VCA_CHAN_IN_PARAM	NET_VCA_BEHAVIOR_ABILITY
MATRIXDECODER_ABILITY	获取多路解码器显示、解码能力	无	NET_DVR_MATRIX_ABILITY
DECODER_ABILITY	获取解码器 XML 能力集	解码器能力集获取输入描述	解码器能力集 XML 描述
SNAPCAMERA_ABILITY	获取智能交通摄像机的能力	无	NET_DVR_SNAP_ABILITY
LCD_SCREEN_ABILITY	获取 LCD 拼接屏能力	无	屏幕能力 XML 描述

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

[返回目录](#)

5.7 布防、撤防

设置报警等信息上传的回调函数

5.7.1 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，详见表 5.6
[out] pAlarmer 报警设备信息，详见：[NET_DVR_ALARMER](#)
[out] pAlarmInfo 报警信息，详见表 5.7
[out] dwBufLen 报警信息缓存大小
[out] pUser 用户数据

表 5.6 报警布防信息类型

ICommand 宏定义	宏定义值	含义
COMM_ALARM_LCD	0x5011	LCD 屏幕报警信息上传

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

说明： 该接口中回调函数的第一个参数（ICommand）和第三个参数（pAlarmInfo）是密切关联的，其关系见表 5.7。

表 5.7 报警信息结构

消息类型（ICommand）	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM_LCD	LCD 屏幕报警信息	NET_DVR_LCD_ALARM

LCD 服务器在空闲时会每隔一定时间上传各屏幕工作状态信息。

[返回目录](#)

布防撤防

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

函 数： LONG NET_DVR_SetupAlarmChan_V41(LONG IUserID, [LPNET_DVR_SETUPALARM_PARAM](#) lpSetupParam)

参 数： [in] IUserID NET_DVR_Login_V30 的返回值
[in] lpSetupParam 报警布防参数，详见：[NET_DVR_SETUPALARM_PARAM](#)

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

说 明： 启动布防前，需要调用注册回调函数的接口（如 [NET_DVR_SetDVRMessageCallBack_V30](#)）才能获取到上传的报警信息。

[返回目录](#)

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

函 数： BOOL NET_DVR_CloseAlarmChan_V30(LONG IAlarmHandle)

参 数： [in] IAlarmHandle NET_DVR_SetupAlarmChan_V41 的返回值

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

说 明：

[返回目录](#)

5.8 监听报警

5.8.1 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，详见表 5.8

[out] pAlarmer 报警设备信息，详见：[NET_DVR_ALARMER](#)

[out] pAlarmInfo 报警信息，详见表 5.9

[out] dwBufLen 报警信息缓存大小

[out] pUser 用户数据

表 5.8 报警监听消息类型

ICommand 宏定义	宏定义值	含义
COMM_ALARM_LCD	0x5011	LCD 屏幕报警信息上传

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

- 说明:**
- 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。**
 - 接口中回调函数的第一个参数 (ICommand) 和第三个参数 (pAlarmInfo) 是密切关联的, 其关系见表 5.9。

表 5.9 报警信息结构

消息类型 (ICommand)	上传内容	pAlarmInfo 对应的结构体
COMM_ALARM_LCD	LCD 屏幕报警信息	NET_DVR_LCD_ALARM

[返回目录](#)

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

函数: BOOL NET_DVR_StopListen_V30(LONG IListenHandle)

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

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

说明:

[返回目录](#)

5.9 通用参数配置

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

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

参数: [in] IUserID 登录主控板, NET_DVR_Login_V30 的返回值

[in] dwCommand 设备配置命令, 详见表 5.10

[in] IChannel 通道号或者其他序号, 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, 详见表 5.10

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

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

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

说明： 不同的获取功能对应不同的结构体和命令号，如表 5.10 所示。

表 5.10 参数获取命令列表

dwCommand 宏定义	dwCommand 含义	IChannel	lpOutBuffer 对应结构体	宏定义值
NET_DVR_GET_DEVICECFG_V40	获取设备参数	无效	NET_DVR_DEVICECFG_V40	1100
NET_DVR_GET_NETCFG_V30	获取网络参数	无效	NET_DVR_NETCFG_V30	1000
NET_DVR_GET_PICCFG	获取图像参数	有效	NET_DVR_PICCFG	104
NET_DVR_GET_COMPRESSCFG_V30	获取压缩参数	有效	NET_DVR_COMPRESSIONCFG_V30	106
NET_DVR_GET_TIMECFG	获取时间参数	无效	NET_DVR_TIME	118
NET_DVR_GET_EXCEPTIONCFG_V30	获取异常参数	无效	NET_DVR_EXCEPTION_V30	126
NET_DVR_GET_RS232CFG_V30	获取 RS232 串口参数	无效	NET_DVR_RS232CFG_V30	1036
NET_DVR_GET_EXCEPTIONCFG_V40	获取异常参数	组号，从 0 开始，每组 32 种异常	NET_DVR_EXCEPTION_V40	6177
NET_DVR_GET_ZONEANDDST	获取时区和夏令制参数	无效	NET_DVR_ZONEANDDST	128
NET_DVR_GET_HOLIDAY_PARAM_CFG	获取节假日参数	无效	NET_DVR_HOLIDAY_PARAM_CFG	1240
NET_DVR_BIGSCREEN_GETSERIAL	获取大屏串口信息	无效	NET_DVR_SERIAL_CONTROL	1614
NET_DVR_GET_VCS_USER_CFG	获取用户信息	用户号，从 1 开始	NET_DVR_VCS_USER_INFO	1623
NET_DVR_GET_USING_SERIALPORT	获取当前使用串口	无效	NET_DVR_USING_SERIALPORT	1742
NET_DVR_GET_POWERON_DELAY_CFG	获取延时开机参数	无效	NET_DVR_DELAY_TIME	9088
NET_DVR_GET_EXTERNAL_MATRIX_RELATION	获取外接矩阵输入输出关联关系	矩阵 ID 号	NET_DVR_IO_RELATION_INFO	9095

Remark

- 用户名密码采用简单加密方式进行加密解密
- 管理员登陆时，普通用户不能操作。管理员没有登陆时，几个普通用户可以同时操作。
- 管理员具有所有操作权限。（包括屏幕和设备）。
- 管理员能够添加、修改、删除普通用户信息。
- 管理员能够给普通用户分配输入设备，每个输入设备都只能被一个普通用户关联。（管理员具有对所有设备操作的权限）。
- 管理员能够给用户划分不同操作屏幕区域，普通用户只能在该区域内进行窗口操作。普通用户只有开窗口和移动窗口操作权限，没有其他权限。
- 屏幕服务器只支持一个管理员用户。
- 当矩阵协议为自定义时，通过透明通道接口发送自定义控制命令。
- 22 寸监视器使用 LCD-D5022 屏幕类型。

[返回目录](#)

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

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

参数: [in] IUserID 登录主控板, NET_DVR_Login_V30 的返回值
[in] dwCommand 设备配置命令, 详见表 5.11
[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, 详见表 5.11
[in] dwInBufferSize 输入数据的缓冲长度(以字节为单位)

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

说明: 不同的获取功能对应不同的结构体和命令号, 如表 5.11 所示。

表 5.11 参数配置命令列表

dwCommand 宏定义	dwCommand 含义	IChannel	lpInBuffer 对应结构体	宏定义值
NET_DVR_SET_DEVICECFG_V40	设置设备参数	无效	NET_DVR_DEVICECFG_V40	1101
NET_DVR_SET_NETCFG_V30	设置网络参数	无效	NET_DVR_NETCFG_V30	1001
NET_DVR_SET_PICCFG	设置图像参数	有效	NET_DVR_PICCFG	105
NET_DVR_SET_COMPRESSCFG	设置压缩参数	有效	NET_DVR_COMPRESSIONCFG_V30	107
NET_DVR_SET_TIMECFG	设置时间参数	无效	NET_DVR_TIME	119
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_RS232CFG_V30	设置 RS232 串口参数	无效	NET_DVR_RS232CFG_V30	1037
NET_DVR_SET_EXCEPTIONCFG_V40	设置异常参数	组号, 从 0 开始, 每组 32 种异常	NET_DVR_EXCEPTION_V40	6178
NET_DVR_SET_VCS_USER_CFG	设置用户信息	用户号, 从 1 开始	NET_DVR_VCS_USER_INFO	1624
NET_DVR_SET_USING_SERIALPORT	设置当前使用串口	无效	NET_DVR_USING_SERIALPORT	1743
NET_DVR_SET_POWERON_DELAY_CFG	设置延时开机参数	无效	NET_DVR_DELAY_TIME	9087

[返回目录](#)

5.10 LCD 屏远程参数配置

5.10.1 启动远程配置（屏幕拼接、配置参数） **NET_DVR_StartRemoteConfig**

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

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

[in] dwCommand	配置命令, 不同的功能对应不同的命令号(dwCommand), lpInBuffer 等参数也对应不同的内容, 详见表 5.12
[in] lpInBuffer	输入参数, 具体内容跟配置命令相关
[in] dwInBufferLen	输入缓冲的大小
[in] cbStateCallback	状态回调函数
[in] pUserData	用户数据

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

[out] dwType	配置状态, 具体定义如下所示, 详见表 5.13
[out] lpBuffer	存放数据的缓冲区指针, 具体内容跟 dwType 相关, 详见列表
[out] dwBufLen	缓冲区大小
[out] pUserData	用户数据

表 5.12 屏幕拼接参数说明

dwCommand 宏定义	宏定义值	含义	lpInBuffer	cbStateCallback
NET_DVR_SCREEN_SPLICE_SET	9081	设置屏幕拼接参数	NULL	返回状态
NET_DVR_SCREEN_SPLICE_GET	9082	获取屏幕拼接参数	NULL	返回状态、信息数据
NET_DVR_SET_SCREEN_PARAM	9083	设置屏幕相关参数	NET_DVR_MSC_SCREEN_REMOTE_CFG	返回状态
NET_DVR_GET_SCREEN_PARAM	9084	获取屏幕相关参数	NET_DVR_MSC_SCREEN_PARAM_COND	返回状态、信息数据

表 5.13 回调状态类型

dwType 宏定义	含义	lpBuffer 对应内容
NET_SDK_CALLBACK_TYPE_STATUS	状态值	回调状态值 typedef enum { NET_SDK_CALLBACK_STATUS_SUCCESS = 1000, //成功 NET_SDK_CALLBACK_STATUS_PROCESSING, //处理中 NET_SDK_CALLBACK_STATUS_FAILED //失败 NET_SDK_CALLBACK_STATUS_EXCEPTION, // 异常 NET_SDK_CALLBACK_STATUS_LANGUAGE_MISMATCH, // (IPC 配置文件导入) 语言不匹配 NET_SDK_CALLBACK_STATUS_DEV_TYPE_MISMATCH // (IPC 配置文件导入) 设备类型不匹配 }NET_SDK_CALLBACK_STATUS_NORMAL;
NET_SDK_CALLBACK_TYPE_PROGRESS	进度值	lpBuffer 的值表示进度值(DWORD)
NET_SDK_CALLBACK_TYPE_DATA	信息数据	获取屏幕拼接参数时对应结构体: NET_DVR_MSC_SPLICE_CFG 获取屏幕相关参数时对应结构体: NET_DVR_MSC_SCREEN_PARAM_CFG

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

说明： 该接口中回调函数的第二个参数（dwCommand）为 NET_DVR_SCREEN_SPLICE_SET 时，则后续需要调用接口 [NET_DVR_SendRemoteConfig](#)，如下表 5.14 说明。

表 5.14 接口调用说明

dwCommand 宏定义	含义	后续接口调用
NET_DVR_SCREEN_SPLICE_SET	设置屏幕拼接参数	NET_DVR_SendRemoteConfig

[返回目录](#)

5.10.2 发送长连接数据（设置屏幕拼接参数）[NET_DVR_SendRemoteConfig](#)

函数： BOOL NET_DVR_SendRemoteConfig(LONG IHandle, DWORD dwDataType, char *pSendBuf, DWORD dwBufSize)

参数： [in] IHandle 句柄，NET_DVR_StartRemoteConfig 的返回值
 [in] dwDataType 数据类型，跟长连接接口 NET_DVR_StartRemoteConfig 的命令参数（dwCommand）有关，详见表 5.15 说明
 [in] pSendBuf 保存发送数据的缓冲区，与 dwDataType 有关，详见“Remarks”说明
 [in] dwBufSize 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： 不同的数据类型(dwDataType)，pSendBuf 对应不同的结构体，如下表所示：

表 5.15 屏幕拼接参数说明

定义	宏定义值	含义
NET_DVR_SCREEN_SPLICE_SET	9081	设置屏幕拼接参数

dwCommand 宏定义	宏定义值	含义	pSendBuf 对应结构体
ENUM_MSC_SEND_DATA	2	屏幕控制器数据类型	NET_DVR_MSC_SPLICE_CFG

回调状态值

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

[返回目录](#)

5.10.3 关闭长连接 **NET_DVR_StopRemoteConfig**

函 数: BOOL NET_DVR_StopRemoteConfig(LONG IHandle)

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

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

说 明:

[返回目录](#)

5.11 解码功能

显示通道配置

5.11.1 获取显示通道配置参数 **NET_DVR_MatrixGetDisplayCfg_V41**

函 数: BOOL NET_DVR_MatrixGetDisplayCfg_V41 (LONG IUserID, DWORD dwDispChanNum, [LPNET_DVR_MATRIX_VOUTCFG](#) lpVoutCfg)

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

[in] dwDispChanNum 显示通道, 从能力集获取

[out] lpVoutCfg 显示通道配置参数, 详见: [NET_DVR_MATRIX_VOUTCFG](#)

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

说 明:

[返回目录](#)

5.11.2 设置显示通道配置参数 **NET_DVR_MatrixSetDisplayCfg_V41**

函 数: BOOL NET_DVR_MatrixGetDisplayCfg_V41 (LONG IUserID, DWORD dwDispChanNum, [LPNET_DVR_MATRIX_VOUTCFG](#) lpDisplayCfg)

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

[in] dwDispChanNum 显示通道, 从能力集获取

[in] lpDisplayCfg 显示通道配置参数, 详见: [NET_DVR_MATRIX_VOUTCFG](#)

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

说 明: 对于 DS6401HD, VGA、BNC、HDMI、DVI 四个显示通道同时只能在一个通道上显示图像, 即最后配置的显示通道有效。

[返回目录](#)

5.12 窗口解码操作

动态解码

5.12.1 启动动态解码 **NET_DVR_MatrixStartDynamic_V41**

函数: BOOL NET_DVR_MatrixStartDynamic_V41(LONG IUserID, DWORD dwDecChanNum, [LPNET_DVR_PU_STREAM_CFG_V41](#) lpDynamicInfo)

参数: [in] IUserID 登录主控板, NET_DVR_Login_V30 返回值
 [in] dwDecChanNum 子窗口号(组合)
 [in] lpDynamicInfo 动态解码参数, 详见: [NET_DVR_PU_STREAM_CFG_V41](#)

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

说明:

[返回目录](#)

5.12.2 停止动态解码 **NET_DVR_MatrixStopDynamic**

函数: BOOL NET_DVR_MatrixStopDynamic(LONG IUserID, DWORD dwDecChanNum)

参数: [in] IUserID 登录主控板, NET_DVR_Login_V30 返回值
 [in] dwDecChanNum 子窗口号(组合)

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

说明:

[返回目录](#)

远程文件回放

5.12.3 远程文件回放解码 **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 返回值
 [in] dwCommand 控制命令, 详见表 5.16
 [in] lpInBuffer 输入参数, 跟控制命令相关, 详见表 5.16
 [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, 详见表 5.16。

表 5.16 远程文件回放控制命令

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer 对应结构体
NET_DVR_DEC_PLAY_REMOTE_FILE	9032	解码播放远程文件	NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX

[返回目录](#)

5.12.4 远程回放文件控制 **NET_DVR_MatrixSetRemotePlayControl**

函数: BOOL NET_DVR_MatrixSetRemotePlayControl(LONG lUserID, DWORD dwDecChanNum, DWORD dwControlCode, DWORD dwInValue, DWORD *lpOutValue)

参数: [in] lUserID 用户 ID 号, NET_DVR_Login_V30 返回值
[in] dwDecChanNum 子窗口号(组合)
[in] dwControlCode 控制命令, 具体定义见详见表 5.17
[in] dwInValue 设置参数
[out] lpOutValue 获取到的参数指针

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: 接口中的 dwInValue 和 lpOutValue 参数根据不同的命令号决定是否输入和输出, 如当进行 NET_DVR_PLAYSETPOS 命令操作时, 需要对 dwInValue 参数进行赋值。

表 5.17 回放控制命令

宏定义	宏定义值	含义
NET_DVR_PLAYSTART	1	开始播放
NET_DVR_PLAYSTOP	2	停止播放
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
NET_DVR_PLAYSTARTAUDIO	9	打开声音
NET_DVR_PLAYSTOPAUDIO	10	关闭声音
NET_DVR_PLAYSETPOS	12	改变文件回放的进度

[返回目录](#)

5.12.5 获取回放状态 **NET_DVR_MatrixGetRemotePlayStatus**

函数: BOOL NET_DVR_MatrixGetRemotePlayStatus(LONG lUserID, DWORD dwDecChanNum, [LPNET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS](#) lpOuter)

参数: [in] lUserID 登录主控板, NET_DVR_Login_V30 返回值
[in] dwDecChanNum 子窗口号(组合)
[out] lpOuter 回放状态, 详见: [NET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS](#)

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

说明： 按时间回放不支持进度控制；由于回放控制命令是转发实现，存在一定的延迟，因此，回放控制命令不宜过于频繁调用，具体视网络状况而定，当把获取的状态作为客户端处理依据时应考虑网络转发的延迟因素；按时间回放时，获取回放状态所得到的文件信息是当前播放的单个片段的信息，并非整个时间范围内全部片段的信息，判断播放是否结束使用 NET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS 结构中的 dwCurDataType 成员。

[返回目录](#)

解码开关控制

5.12.6 获取当前解码通道开关 NET_DVR_MatrixGetDecChanEnable

函数： BOOL NET_DVR_MatrixGetDecChanEnable(LONG IUserID, DWORD dwDecChanNum, LPDWORD lpdwEnable)

参数： [in] IUserID 登录主控板，NET_DVR_Login_V30 返回值
[in] dwDecChanNum 子窗口号(组合)
[out] lpdwEnable 0 表示关闭，1 表示打开

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

说明： 窗口的解码开关，用于控制窗口的解码过程，设置此开关为关时，无论当前窗口是处在动态解码还是轮巡过程，都将停止解码，生效后该通道处于黑屏状态；设置开关为开时，将恢复先前过程。此功能可与轮巡，轮巡开关等配合使用。

[返回目录](#)

5.12.7 设置当前解码通道开关 NET_DVR_MatrixSetDecChanEnable

函数： BOOL NET_DVR_MatrixSetDecChanEnable (LONG IUserID, DWORD dwDecChanNum, DWORD dwEnable)

参数： [in] IUserID 登录主控板，NET_DVR_Login_V30 返回值
[in] dwDecChanNum 子窗口号(组合)
[in] dwEnable 0 表示关闭，1 表示打开

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

说明： 窗口的解码开关，用于控制窗口的解码过程，设置此开关为关时，无论当前窗口是处在动态解码还是轮巡过程，都将停止解码，生效后该通道处于黑屏状态；设置开关为开时，将恢复先前过程。此功能可与轮巡，轮巡开关等配合使用。

[返回目录](#)

轮巡解码

5.12.8 获取轮巡解码通道 **NET_DVR_MatrixGetLoopDecChanInfo_V41**

函数: BOOL NET_DVR_MatrixGetLoopDecChanInfo_V41(LONG IUserID, DWORD dwDecChanNum, [LPNET_DVR_MATRIX_LOOP_DECINFO_V41](#) lpOuter)

参数: [in] IUserID 登录主控板, NET_DVR_Login_V30 返回值
 [in] dwDecChanNum 子窗口号(组合)
 [out] lpOuter 轮巡的解码通道参数, 详见:
[NET_DVR_MATRIX_LOOP_DECINFO_V41](#)

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

说明:

[返回目录](#)

5.12.9 设置轮巡解码通道 **NET_DVR_MatrixSetLoopDecChanInfo_V41**

函数: BOOL NET_DVR_MatrixSetLoopDecChanInfo_V41(LONG IUserID, DWORD dwDecChanNum, [LPNET_DVR_MATRIX_LOOP_DECINFO_V41](#) lpInter)

参数: [in] IUserID 登录主控板, NET_DVR_Login_V30 返回值
 [in] dwDecChanNum 子窗口号(组合)
 [in] lpInter 轮巡的解码通道参数, 详见:
[NET_DVR_MATRIX_LOOP_DECINFO_V41](#)

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

说明:

[返回目录](#)

5.12.10 获取单个解码通道轮巡开关 **NET_DVR_MatrixGetLoopDecChanEnable**

函数: BOOL NET_DVR_MatrixGetLoopDecChanEnable(LONG IUserID, DWORD dwDecChanNum, LPDWORD lpdwEnable)

参数: [in] IUserID 登录主控板, NET_DVR_Login_V30 返回值
 [in] dwDecChanNum 子窗口号(组合)
 [out] lpdwEnable 0 表示关闭, 1 表示打开

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

说明:

[返回目录](#)

5.12.11 设置单个解码通道轮巡开关 **NET_DVR_MatrixSetLoopDecChanEnable**

函 数: BOOL NET_DVR_MatrixSetLoopDecChanEnable(LONG IUserID, DWORD dwDecChanNum, DWORD dwEnable)

参 数: [in] IUserID 登录主控板, NET_DVR_Login_V30 返回值
 [in] dwDecChanNum 子窗口号(组合)
 [in] dwEnable 0 表示关闭, 1 表示打开

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

说 明: 轮巡开关用于控制轮巡过程的启停, 而不是控制解码的启停, 当设置当前正在轮巡解码的窗口的轮巡开关为关时, 该窗口停止循环, 停留在当前所连接的输入源继续解码, 退为动态解码, 当设置轮巡开关为开时, 设备恢复该窗口的循环。

[返回目录](#)

5.12.12 获取所有解码通道轮巡开关 **NET_DVR_MatrixGetLoopDecEnable**

函 数: BOOL NET_DVR_MatrixGetLoopDecEnable (LONG IUserID, LPDWORD lpdwEnable)

参 数: [in] IUserID 登录主控板, NET_DVR_Login_V30 返回值
 [out] lpdwEnable 按位表示, 0 表示关闭, 1 表示打开 0 表示关闭, 1 表示打开

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

说 明: 轮巡开关用于控制轮巡过程的启停, 而不是控制解码的启停, 当设置解码器当前正在轮巡的解码通道的轮巡开关为关时, 该解码通道停止循环, 停留在当前所连接的前端设备通道继续解码, 退为动态解码, 当设置轮巡开关为开时, 解码器恢复该解码通道的循环。

[返回目录](#)

获取解码信息

5.12.13 获取当前解码通道信息 **NET_DVR_MatrixGetDecChanInfo_V41**

函 数: BOOL NET_DVR_MatrixGetDecChanInfo_V41 (LONG IUserID, DWORD dwDecChanNum, LPNET_DVR_MATRIX_DEC_CHAN_INFO_V41 lpOuter)

参 数: [in] IUserID 登录主控板, NET_DVR_Login_V30 返回值
 [in] dwDecChanNum 解码通道
 [out] lpdwEnable 解码通道信息, 详见: [NET_DVR_MATRIX_DEC_CHAN_INFO_V41](#)

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

说 明:

[返回目录](#)

5.12.14 获取当前解码状态 **NET_DVR_MatrixGetDeviceStatus_V41**

函 数： BOOL NET_DVR_MatrixGetDeviceStatus_V41 (LONG IUserID, LPNET_DVR_DECODER_WORK_STATUS_V41 lpDecoderCfg)

参 数： [in] IUserID 登录主控板，NET_DVR_Login_V30 返回值
[out] lpdwEnable 设备状态参数，详见：[NET_DVR_DECODER_WORK_STATUS_V41](#)

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

说 明：

[返回目录](#)

5.13 场景配置、操作

5.13.1 远程控制（场景、屏幕控制以及矩阵输入输出切换）

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 返回值
[in] dwCommand 控制命令，详见表 5.18
[in] lpInBuffer 输入参数，跟控制命令相关，详见列表
[in] dwInBufferSize 输入参数长度

返回值： TRUE 表示成功，FALSE 表示失败。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，如表 5.18 所示。

表 5.18 远程控制命令

dwCommand 宏定义	宏定义值	控制功能	lpInBuffer 对应结构体
NET_DVR_SCREEN_SCENE_CONTROL	9091	屏幕场景控制	NET_DVR_SCENE_CONTROL_INFO
NET_DVR_CONTROL_SCREEN	1625	屏幕控制	NET_DVR_SCREEN_CONTROL_V41
NET_DVR_SWITCH_MATRIX_IO	1334	矩阵输入输出切换	NET_DVR_MATRIX_SWITCH_INFO

Remark:

使用坐标范围进行计算，如同时控制左上角 4 个屏幕，则坐标参数是 (0,0, 1920*2,1920*2)

[返回目录](#)

5.13.2 批量获取信息 **NET_DVR_GetDeviceConfig**

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

参 数:	[in] IUserID	用户 ID 号, NET_DVR_Login_V30 的返回值
	[in] dwCommand	设备配置命令, 详见表 5.19
	[in] dwCount	批量获取个数, 0 和 1 都表示 1 个, 依次递增, 最大 64 个
	[in] lpInBuffer	配置条件缓冲区, 详见表 5.20
	[in] dwInBufferSize	缓冲区长度
	[out] lpStatusList	错误信息列表, 和要查询的监控点一一对应, 例如 lpStatusList[2] 就对应 lpInBuffer[2], 由用户分配内存, 每个错误信息为 4 个字节, 参数值: 0- 成功, 大于 0-失败
	[out] lpOutBuffer	设备返回的参数内容 (详见表 5.20), 和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 0 值, 对应 lpOutBuffer 的内容就是无效的
	[in] dwOutBufferSize	输出缓冲区大小

表 5.19 批量参数获取命令

dwCommand 宏定义	dwCommand 含义	宏定义值
NET_DVR_WALLOUTPUT_GET	电视墙中获取显示输出参数	9011
NET_DVR_SCREEN_INDEX_GET	获取屏幕索引相关参数	9080
NET_DVR_GET_SCREEN_POSITION	获取屏幕位置参数	9090
NET_DVR_GET_CURRENT_SCREEN_SCENE	获取当前屏幕场景号(dwCount 为 1)	9092
NET_DVR_GET_SCREEN_SCENE_PARAM	获取屏幕场景参数	9093
NET_DVR_GET_SWITCH_CFG	获取定时开关机参数(dwCount 为 1)	9086
NET_DVR_GET_EXTERNAL_MATRIX_CFG	获取矩阵信息	1619
NET_DVR_GET_EXTERNAL_MATRIX_RELATION	获取外接矩阵输入输出关联关系	9095

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

说 明: 该接口是带有发送数据的批量获取配置信息的通用接口。lpInBuffer 指定需要获取的 dwCount 个配置信息, lpOutBuffer 保存获取得到的 dwCount 个配置信息。
全部获取时 dwCount 置为 0xffffffff, lpInBuffer 设为 1 个电视墙号(组合), dwInBufferSize 置为 4, lpStatusList 置为 NULL, lpOutBuffer 前 4 个字节为返回的个数。如果设置的 lpOutBuffer 缓冲区不足, 仅返回部分信息, 可以根据返回的个数 (前 4 字节的值) 重新获取。

表 5.20 批量参数获取结构

dwCommand 宏定义	lpInBuffer 对应结构体	lpOutBuffer 对应结构体
NET_DVR_WALLOUTPUT_GET	dwCount 个 4 字节索引号 (1 字节设备号+1 字节保留+2 字节索引号)	dwCount 个 NET_DVR_WALLOUTPUTPARAM
NET_DVR_SCREEN_INDEX_GET	dwCount 个 4 字节索引号	dwCount 个 NET_DVR_SCREEN_INDEX_CFG
NET_DVR_GET_SCREEN_POSITION	dwCount 个 4 字节索引号 (1 字节设备号+1 字节保留+2 字节索引号)	dwCount 个 NET_DVR_VIDEO_WALL_DISPLAY_POSITION
NET_DVR_GET_CURRENT_SCREEN_SCENE	NET_DVR_VIDEO_WALL_INFO	4 字节场景号

EN_SCENE		
NET_DVR_GET_SCREEN_SCENE_PARAM	dwCount 个 NET_DVR_VIDEO_WALL_INFO ，获取所有时传入一个 NET_DVR_VIDEO_WALL_INFO ，指定墙号	dwCount 个 NET_DVR_WALLSCENECFG
NET_DVR_GET_SWITCH_CFG	NET_DVR_VIDEO_WALL_AREA	NET_DVR_TIME_SWITCH_CFG
NET_DVR_GET_EXTERNAL_MATRIX_CFG	dwCount 个 4 字节矩阵 ID	dwCount 个 NET_DVR_EXTERNAL_MATRIX_CFG

注：

- 全部获取时 dwCount 置为 0xffffffff,lpInBuffer 置为 NULL,dwInBufferSize 置为 0, lpOuterBuffer 前 4 字节为个数 N,后面为 N 个 [NET_DVR_VIDEOWALLDISPLAYPOSITION](#) 结构, lpStatusList 置为 NULL。
- 对某个墙获取所有屏幕位置时, dwCount 置为 0xffffffff,lpInBuffer 置为一个 4 字节墙号 (1 字节墙号+3 字节保留), dwInBufferSize 置为 4, lpOuterBuffer 前 4 字节为个数 N,后面为 N 个 [NET_DVR_VIDEOWALLDISPLAYPOSITION](#) 结构, lpStatusList 置为 NULL。
- NET_DVR_GET_SCREEN_SCENE_PARAM,此处为 dwCount 个 [NET_DVR_WALLSCENECFG](#) 结构,获取全部时前 4 字节为个数,一次要设置的监控点个数; 0 和 1,都表示 1 个监控点信息,以此递增,最大 64 个,获取全部时填 0xffffffff
- NET_DVR_GET_SWITCH_CFG 目前屏幕服务器不支持按墙区域设置定时开关机参数。

[返回目录](#)

5.13.3 批量设置参数 [NET_DVR_SetDeviceConfig](#)

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

参 数:

[in] IUserID	用户 ID 号, NET_DVR_Login_V30 的返回值
[in] dwCommand	设备配置命令, 详见表 5.21。
[in] dwCount	批量配置个数, 0 和 1 都表示 1 个, 依次递增, 最大 64 个
[in] pInBuffer	配置条件缓冲区, 详见表 5.22
[in] dwInBufferSize	配置条件缓冲区长度
[out]lpStatusList	错误信息列表, 和要查询的监控点一一对应, 例如 lpStatusList[2] 就对应 lpInBuffer[2], 由用户分配内存, 每个错误信息为 4 个字节 (1 个 32 位无符号整数值), 参数值: 0 或者 1 表示成功, 其他值为失败对应的错误号
[in] lpInParamBuffer	需要设置给设备的参数内容, 和要查询的监控点一一对应。如果某个监控点对应的 lpStatusList 信息为大于 1 的值, 表示对应的 lpInBuffer 设置失败, 为 0 或 1 则表示设置成功, 详见表 5.22
[in]dwInParamBufferSize	设置内容缓冲区大小

表 5.21 批量配置

dwCommand 宏定义

dwCommand 含义

宏定义值

NET_DVR_WALLOUTPUT_SET	电视墙中设置显示输出参数	9012
NET_DVR_SCREEN_INDEX_SET	设置屏幕索引相关参数	9079
NET_DVR_SET_SCREEN_POSITION	设置屏幕位置参数	9089
NET_DVR_SET_SCREEN_SCENE_PARAM	设置屏幕场景参数	9094
NET_DVR_SET_SWITCH_CFG	设置定时开关机参数(dwCount为1)	9085
NET_DVR_SET_EXTERNAL_MATRIX_CFG	设置矩阵信息	1620

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

说明: 该接口是带有发送数据的批量设置监控点配置信息的通用接口扩展接口, 支持设置接收数据超时时间。

表 5.22 批量参数配置结构

dwCommand 宏定义	IpInBuffer 对应结构体	IpInParamBuffer 对应结构体
NET_DVR_WALLOUTPUT_SET	dwCount 个 4 字节索引号	dwCount 个 NET_DVR_WALLOUTPUTPARAM
NET_DVR_SCREEN_INDEX_SET	dwCount 个 4 字节索引号	dwCount 个 NET_DVR_SCREEN_INDEX_CFG
NET_DVR_SET_SCREEN_POSITION	dwCount 个 4 字节索引号 (1 字节设备号+1 字节保留+2 字节索引号)	dwCount 个 NET_DVR_VIDEO_WALLDISPLAYPOSITION
NET_DVR_SET_SCREEN_SCENE_PARAM	dwCount 个 NET_DVR_VIDEO_WALL_INFO	dwCount 个 NET_DVR_WALLSCENECFG
NET_DVR_SET_SWITCH_CFG	NET_DVR_VIDEO_WALL_ARCHIVE	NET_DVR_TIME_SWITCH_CFG
NET_DVR_SET_EXTERNAL_MATRIX_CFG	dwCount 个 4 字节矩阵 ID	dwCount 个 NET_DVR_EXTERNAL_MATRIX_CFG

Remark:

- 设备支持的索引号范围通过屏幕能力返回, 索引号和屏幕使用屏幕添加接口进行关联, 关联完毕后使用此接口对配置各个屏幕的位置。

- 单次设置获取个数最大 256 个
- 场景切换中, 只用到墙号、场景号, 获取当前场景只使用墙号。
- 场景切换时, 设备需保存当前正在解码的信号源信息;
- 场景切换, 保存必须在使能的场景间进行;
- 清空场景后, 场景依旧使能, 为空场景;
- 场景的使能通过场景模式参数配置接口进行修改;
- 4 字节索引号 (1 字节设备号+1 字节保留+2 字节索引号)

[返回目录](#)

5.14 透明通道

5.14.1 建立透明通道 **NET_DVR_SerialStart_V40**

函数: `BOOL NET_DVR_SerialStart_V40 (LONG IUserID, void *lpInBuffer, LONG dwInBufferSize, fSerialDataCallBack cbSerialDataCallBack, void *pUser)`

参数:

[in] IUserID	用户 ID, NET_DVR_Login_V30 的返回值
[in] dwCommand	输入缓冲区, 指向透明通道串口参数, 对应 NET_DVR_SERIALSTART_V40
[in] dwInBufferSize	输入缓冲区大小
[in] cbSerialDataCallBack	串口数据回调函数
[in] pUser	用户数据

```
typedef void(CALLBACK *fSerialDataCallBack)(LONG ISerialHandle, LONG IChannel, char *pRecvDataBuffer, DWORD dwBufSize, void *pUser);
```

[out] ISerialHandle	NET_DVR_SerialStart_V40 的返回值
[out] IChannel	通道号, 透明通道类型为 RS485 通道时有效
[out] pRecvDataBuffer	存放数据的缓冲区指针
[out] dwBufSize	数据大小
[out] pUser	用户数据

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

说明: 如果串口接的外设不支持双工 (全双工或者半双工), 设备不会有数据返回给 SDK。

[返回目录](#)

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

说明:

[返回目录](#)

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

说 明:

[返回目录](#)

5.15 设备维护

远程升级

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

函 数: BOOL NET_DVR_SetNetworkEnvironment(DWORD dwEnvironmentLevel)

参 数: [in]dwEnvironmentLevel 网络环境级别

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

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

说 明: 接口中的网络环境级别参数分为两类:

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

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

在调用远程升级接口之前, 可以通过此接口适应不同的升级环境。

[返回目录](#)

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

说 明:

[返回目录](#)

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

函数: `Int NET_DVR_GetUpgradeProgress(LONG IUpgradeHandle)`

参数: `[in]IUpgradeHandle` NET_DVR_Upgrade 的返回值

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

说明:

[返回目录](#)

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

说明:

[返回目录](#)

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

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

参数: `[in]IUpgradeHandle` NET_DVR_Upgrade 的返回值

`[in]pSubProgress` 升级阶段子进度

返回值: -1 表示失败, 其他值定义如表 5.23 所示。

表 5.23 升级阶段

宏定义	宏定义值	含义
STEP_RECV_DATA	1	接收升级包数据
STEP_UPGRADE	2	升级系统
STEP_BACKUP	3	备份系统
STEP_SEARCH	255	设备正在搜索升级文件

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

说明:

[返回目录](#)

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

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

说明：

[返回目录](#)

日志查找

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

函数： LONG NET_DVR_FindDVRLog_V30(LONG IUserID, LONG ISelectMode, DWORD dwMajorType, DWORD dwMinorType, [LPNET_DVR_TIME](#) IpStartTime, [LPNET_DVR_TIME](#) IpStopTime, BOOL bOnlySmart = FALSE)
参数： [in]IUserID NET_DVR_Login_V30 的返回值
[in]ISelectMode 查询方式：0- 全部，1- 按类型，2- 按时间，3- 按时间和类型
[in]dwMajorType 日志主类型（S.M.A.R.T 搜索时无效），0 表示全部类型，其他类型定义见 [日志主类型列表](#)
[in]dwMinorType 日志次类型（S.M.A.R.T 搜索时无效），0 表示全部类型，根据不同的主类型的次类型定义见 [日志次类型列表](#)
[in]IpStartTime 文件的开始时间，详见：[NET_DVR_TIME](#)
[in]IpStopTime 文件结束时间，详见：[NET_DVR_TIME](#)
[in]bOnlySmart 是否只搜索带 S.M.A.R.T 信息的日志
返回值： -1 表示失败，其他值作为 NET_DVR_FindNextLog_V30 等函数的参数。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。

说明： 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。

[返回目录](#)

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

函数： LONG NET_DVR_FindNextLog_V30(LONG ILogHandle, [LPNET_DVR_LOG_V30](#) IpLogData)
参数： [in]ILogHandle 日志查找句柄，NET_DVR_FindDVRLog_V30 的返回值
[out]IpLogData 保存日志信息的指针，详见：[NET_DVR_LOG_V30](#)
返回值： -1 表示失败，其他值表示当前的获取状态等信息，详见表 5.。接口返回失败请调用 [NET_DVR_GetLastError](#) 获取错误码，通过错误码判断出错原因。

表 5.24 状态信息

宏定义	宏定义值	含义
NET_DVR_FILE_SUCCESS	1000	获取日志信息成功

NET_DVR_FILE_NOFOUND	1001	未查找到日志
NET_DVR_ISFINDING	1002	正在查找请等待
NET_DVR_NOMOREFILE	1003	没有更多的日志，查找结束
NET_DVR_FILE_EXCEPTION	1004	查找日志时异常

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。查找结束后调用 [NET_DVR_FindLogClose_V30](#) 释放资源。

[返回目录](#)

5.15.9 释放查找日志的资源 [NET_DVR_FindLogClose_V30](#)

函数： BOOL NET_DVR_FindLogClose_V30(LONG ILogHandle)

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

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

说明：

[返回目录](#)

远程恢复默认参数、重启

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

说明：

[返回目录](#)

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

说明：

[返回目录](#)

配置文件导入导出

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

说 明:

[返回目录](#)

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

说 明:

[返回目录](#)

6 错误代码及说明

6.1 网络通讯库错误码

错误名称	错误值	说明
NET_DVR_NOERROR	0	没有错误。
NET_DVR_PASSWORD_ERROR	1	用户名密码错误。注册时输入的用户名或者密码错误。
NET_DVR_NOENOUGHPRI	2	权限不足。该注册用户没有权限执行当前对设备的操作，可以与远程用户参数配置做对比。
NET_DVR_NOINIT	3	SDK 未初始化。
NET_DVR_CHANNEL_ERROR	4	通道号错误。设备没有对应的通道号。
NET_DVR_OVER_MAXLINK	5	连接到设备的用户个数超过最大。
NET_DVR_VERSIONNOMATCH	6	版本不匹配。SDK 和设备的版本不匹配。
NET_DVR_NETWORK_FAIL_CONNECT	7	连接设备失败。设备不在线或网络原因引起的连接超时等。
NET_DVR_NETWORK_SEND_ERROR	8	向设备发送失败。
NET_DVR_NETWORK_RECV_ERROR	9	从设备接收数据失败。
NET_DVR_NETWORK_RECV_TIMEOUT	10	从设备接收数据超时。
NET_DVR_NETWORK_ERRORDATA	11	传送的数据有误。发送给设备或者从设备接收到的数据错误，如远程参数配置时输入设备不支持的值。
NET_DVR_ORDER_ERROR	12	调用次序错误。
NET_DVR_OPERNOPERMIT	13	无此权限。
NET_DVR_COMMANDTIMEOUT	14	设备命令执行超时。
NET_DVR_ERRORSERIALPORT	15	串口号错误。指定的设备串口号不存在。
NET_DVR_ERRORALARMPORT	16	报警端口错误。指定的设备报警输出端口不存在。
NET_DVR_PARAMETER_ERROR	17	参数错误。SDK 接口中给入的输入或输出参数为空。
NET_DVR_CHAN_EXCEPTION	18	设备通道处于错误状态
NET_DVR_NODISK	19	设备无硬盘。当设备无硬盘时，对设备的录像文件、硬盘配置等操作失败。
NET_DVR_ERRORDISKNUM	20	硬盘号错误。当对设备进行硬盘管理操作时，指定的硬盘号不存在时返回该错误。
NET_DVR_DISK_FULL	21	设备硬盘满。
NET_DVR_DISK_ERROR	22	设备硬盘出错
NET_DVR_NOSUPPORT	23	设备不支持。
NET_DVR_BUSY	24	设备忙。
NET_DVR_MODIFY_FAIL	25	设备修改不成功。
NET_DVR_PASSWORD_FORMAT_ERROR	26	密码输入格式不正确
NET_DVR_DISK_FORMATING	27	硬盘正在格式化，不能启动操作。
NET_DVR_DVRNORESOURCE	28	设备资源不足。
NET_DVR_DVROPRATEFAILED	29	设备操作失败。
NET_DVR_OPENHOSTSOUND_FAIL	30	语音对讲、语音广播操作中采集本地音频或打开音频输出失败。
NET_DVR_DVRVOICEOPENED	31	设备语音对讲被占用。

NET_DVR_TIMEINPUTERROR	32	时间输入不正确。
NET_DVR_NOSPECFILE	33	回放时设备没有指定的文件。
NET_DVR_CREATEFILE_ERROR	34	创建文件出错。本地录像、保存图片、获取配置文件和远程下载录像时创建文件失败。
NET_DVR_FILEOPENFAIL	35	打开文件出错。设置配置文件、设备升级、上传审讯文件时打开文件失败。
NET_DVR_OPERNOTFINISH	36	上次的操作还没有完成
NET_DVR_GETPLAYTIMEFAIL	37	获取当前播放的时间出错。
NET_DVR_PLAYFAIL	38	播放出错。
NET_DVR_FILEFORMAT_ERROR	39	文件格式不正确。
NET_DVR_DIR_ERROR	40	路径错误
NET_DVR_ALLOC_RESOURCE_ERROR	41	SDK 资源分配错误。
NET_DVR_AUDIO_MODE_ERROR	42	声卡模式错误。当前打开声音播放模式与实际设置的模式不符出错。
NET_DVR_NOENOUGH_BUF	43	缓冲区太小。接收设备数据的缓冲区或存放图片缓冲区不足。
NET_DVR_CREATESOCKET_ERROR	44	创建 SOCKET 出错。
NET_DVR_SETSOCKET_ERROR	45	设置 SOCKET 出错。
NET_DVR_MAX_NUM	46	个数达到最大。分配的注册连接数、预览连接数超过 SDK 支持的最大数。
NET_DVR_USERNOTEXIST	47	用户不存在。注册的用户 ID 已注销或不可用。
NET_DVR_WRITEFLASHERROR	48	写 FLASH 出错。设备升级时写 FLASH 失败。
NET_DVR_UPGRADEFAIL	49	设备升级失败。网络或升级文件语言不匹配等原因升级失败。
NET_DVR_CARDHAVEINIT	50	解码卡已经初始化过。
NET_DVR_PLAYERFAILED	51	调用播放库中某个函数失败。
NET_DVR_MAX_USERNUM	52	登录设备的用户数达到最大。
NET_DVR_GETLOCALIPANDMACFAIL	53	获得本地 PC 的 IP 地址或物理地址失败。
NET_DVR_NOENCODEING	54	设备该通道没有启动编码。
NET_DVR_IPMISMATCH	55	IP 地址不匹配。
NET_DVR_MACMISMATCH	56	MAC 地址不匹配。
NET_DVR_UPGRADELANGMISMATCH	57	升级文件语言不匹配。
NET_DVR_MAX_PLAYERPORT	58	播放器路数达到最大。
NET_DVR_NOSPACEBACKUP	59	备份设备中没有足够空间进行备份。
NET_DVR_NODEVICEBACKUP	60	没有找到指定的备份设备。
NET_DVR_PICTURE_BITS_ERROR	61	图像素位数不符，限 24 色。
NET_DVR_PICTURE_DIMENSION_ERROR	62	图片高*宽超限，限 128*256。
NET_DVR_PICTURE_SIZ_ERROR	63	图片大小超限，限 100K。
NET_DVR_LOADPLAYERSDKFAILED	64	载入当前目录下 Player Sdk 出错。
NET_DVR_LOADPLAYERSDKPROC_ERROR	65	找不到 Player Sdk 中某个函数入口。
NET_DVR_LOADDSSDKFAILED	66	载入当前目录下 DsSdk 出错。
NET_DVR_LOADDSSDKPROC_ERROR	67	找不到 DsSdk 中某个函数入口。
NET_DVR_DSSDK_ERROR	68	调用硬解码库 DsSdk 中某个函数失败。

NET_DVR_VOICEMONOPOLIZE	69	声卡被独占。
NET_DVR_JOINMULTICASTFAILED	70	加入多播组失败。
NET_DVR_CREATEDIR_ERROR	71	建立日志文件目录失败。
NET_DVR_BINDSOCKET_ERROR	72	绑定套接字失败。
NET_DVR_SOCKETCLOSE_ERROR	73	socket 连接中断，此错误通常是由于连接中断或目的地不可达。
NET_DVR_USERID_ISUSING	74	注销时用户 ID 正在进行某操作。
NET_DVR_SOCKETLISTEN_ERROR	75	监听失败。
NET_DVR_PROGRAM_EXCEPTION	76	程序异常。
NET_DVR_WRITEFILE_FAILED	77	写文件失败。本地录像、远程下载录像、下载图片等操作时写文件失败。
NET_DVR_FORMAT_READONLY	78	禁止格式化只读硬盘。
NET_DVR_WITHSAMEUSERNAME	79	远程用户配置结构中存在相同的用户名。
NET_DVR_DEVICETYPE_ERROR	80	导入参数时设备型号不匹配。
NET_DVR_LANGUAGE_ERROR	81	导入参数时语言不匹配。
NET_DVR_PARAVERSION_ERROR	82	导入参数时软件版本不匹配。
NET_DVR_IPCHAN_NOTALIVE	83	预览时外接 IP 通道不在线。
NET_DVR_RTSP_SDK_ERROR	84	加载标准协议通讯库 StreamTransClient 失败。
NET_DVR_CONVERT_SDK_ERROR	85	加载转封装库失败。
NET_DVR_IPC_COUNT_OVERFLOW	86	超出最大的 IP 接入通道数。
NET_DVR_MAX_ADD_NUM	87	添加录像标签或者其他操作超出最多支持的个数。
NET_DVR_PARAMMODE_ERROR	88	图像增强仪，参数模式错误（用于硬件设置时，客户端进行软件设置时错误值）。
NET_DVR_CODESPITTER_OFFLINE	89	码分器不在线。
NET_DVR_BACKUP_COPYING	90	设备正在备份。
NET_DVR_CHAN_NOTSUPPORT	91	通道不支持该操作。
NET_DVR_CALLINEINVALID	92	高度线位置太集中或长度线不够倾斜。
NET_DVR_CALCANCELCONFLICT	93	取消标定冲突，如果设置了规则及全局的实际大小尺寸过滤。
NET_DVR_CALPOINTOUTRANGE	94	标定点超出范围。
NET_DVR_FILTERRECTINVALID	95	尺寸过滤器不符合要求。
NET_DVR_DDNS_DEVOFFLINE	96	设备没有注册到 ddns 上。
NET_DVR_DDNS_INTER_ERROR	97	DDNS 服务器内部错误。
NET_DVR_ALIAS_DUPLICATE	150	别名重复（EasyDDNS 的配置）
NET_ERROR_TRUNK_LINE	711	子系统已被配成干线
NET_ERROR_MIXED_JOINT	712	不能进行混合拼接
NET_ERROR_DISPLAY_SWITCH	713	不能进行显示通道切换
NET_ERROR_USED_BY_BIG_SCREEN	714	解码资源被大屏占用
NET_ERROR_USE_OTHER_DEC_RESOURCE	715	不能使用其他解码子系统资源
NET_ERROR_DISP_MODE_SWITCH	716	显示通道显示状态切换中
NET_ERROR_SCENE_USING	717	场景正在使用
NET_ERR_NO_ENOUGH_DEC_RESOURCE	718	解码资源不足
NET_ERR_NO_ENOUGH_FREE_SHOW_RESOURCE	719	畅显资源不足
NET_ERR_NO_ENOUGH_VIDEO_MEMORY	720	显存资源不足

NET_ERR_MAX_VIDEO_NUM	721	一拖多资源不足
NET_ERR_WIN_COVER_FREE_SHOW_AND_NORMAL	722	窗口跨越了畅显输出口和非畅显输出口
NET_ERR_FREE_SHOW_WIN_SPLIT	723	畅显窗口不支持分屏
NET_ERR_INAPPROPRIATE_WIN_FREE_SHOW	724	不是输出口整数倍的窗口不支持开启畅显
NET_DVR_TRANSPARENT_WIN_NOT_SUPPORT_SPLIT	725	开启透明度的窗口不支持分屏
NET_DVR_SPLIT_WIN_NOT_SUPPORT_TRANSPARENT	726	开启多分屏的窗口不支持透明度设置
NET_ERR_MAX_LOGO_NUM	727	Logo 数达到上限
NET_ERR_MAX_WIN_LOOP_NUM	728	轮巡窗口数达到上限
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
K		
NET_DVR_STATUS_CANT_FORMAT_LITTLE_DISK	802	由于硬盘太小无法格式化
NET_ERR_WINCHAN_IDX	901	开窗通道号错误
NET_ERR_WIN_LAYER	902	窗口层数错误，单个屏幕上最多覆盖的窗口层数
NET_ERR_WIN_BLK_NUM	903	窗口的块数错误，单个窗口可覆盖的屏幕个数
NET_ERR_OUTPUT_RESOLUTION	904	输出分辨率错误
NET_ERR_LAYOUT	905	场景号错误
NET_ERR_INPUT_RESOLUTION	906	输入分辨率不支持
NET_ERR_SUBDEVICE_OFFLINE	907	子设备不在线
NET_ERR_NO_DECODE_CHAN	908	没有空闲解码通道
NET_ERR_MAX_WINDOW_ABILITY	909	开窗能力上限，分布式多屏控制器中解码子设备能力上限或者显示处理器能力上限导致
NET_ERR_ORDER_ERROR	910	调用顺序有误
NET_ERR_PLAYING_PLAN	911	正在执行预案
NET_ERR_DECODER_USED	912	解码板正在使用
NET_ERR_OUTPUT_BOARD_DATA_OVERFLOW	913	输出板数据量超限
NET_ERR_SAME_USER_NAME	914	用户名相同
NET_ERR_INVALID_USER_NAME	915	无效用户名
NET_ERR_MATRIX_USING	916	输入矩阵正在使用
NET_ERR_DIFFERENT_CHAN_TYPE	917	通道类型不同（矩阵输出通道和控制器的输入为不同的类型）
NET_ERR_INPUT_CHAN_BINDED	918	输入通道已经被其他矩阵绑定
NET_ERR_BINDED_OUTPUT_CHAN_OVERFLOW	919	正在使用的矩阵输出通道个数超过矩阵与控制器绑定的通道个数
NET_ERR_MAX_SIGNAL_NUM	920	输入信号源个数达到上限
NET_ERR_INPUT_CHAN_USING	921	输入通道正在使用
NET_ERR_MANAGER_LOGON	922	管理员已经登陆，操作失败
NET_ERR_USERALREADY_LOGON	923	该用户已经登陆，操作失败
NET_ERR_LAYOUT_INIT	924	场景正在初始化，操作失败
NET_ERR_BASEMAP_SIZE_NOT_MATCH	925	底图大小不符
NET_ERR_WINDOW_OPERATING	926	窗口正在执行其他操作，本次操作失败
NET_ERR_SIGNAL_UPLIMIT	927	信号源开窗个数达到上限

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_DESCRIBESENDERROR	412	RTSP DESCRIBE 发送失败
NET_DVR_RTSP_DESCRIBERECVTIMEOUT	413	RTSP DESCRIBE 接收超时
NET_DVR_RTSP_DESCRIBERECVDATAALOST	414	RTSP DESCRIBE 接收数据错误
NET_DVR_RTSP_DESCRIBERECVERROR	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_SETUPSENDTIMEOUT	421	RTSP SETUP 发送超时
NET_DVR_RTSP_SETUPSENDERROR	422	RTSP SETUP 发送错误
NET_DVR_RTSP_SETUPRECVTIMEOUT	423	RTSP SETUP 接收超时
NET_DVR_RTSP_SETUPRECVDATAALOST	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_PLAYSENDTIMEOUT	431	RTSP PLAY 发送超时
NET_DVR_RTSP_PLAYSENDERROR	432	RTSP PLAY 发送错误
NET_DVR_RTSP_PLAYRECVTIMEOUT	433	RTSP PLAT 接收超时
NET_DVR_RTSP_PLAYRECVDATAALOST	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_TEARDOWNSENDTIMEOUT	441	RTSP TEARDOWN 发送超时
NET_DVR_RTSP_TEARDOWNSENDERROR	442	RTSP TEARDOWN 发送错误
NET_DVR_RTSP_TEARDOWNRECVTIMEOUT	443	RTSP TEARDOWN 接收超时
NET_DVR_RTSP_TEARDOWNRECVDATAALOST	444	RTSP TEARDOWN 接收数据错误
NET_DVR_RTSP_TEARDOWNRECVERROR	445	RTSP TEARDOWN 接收失败
NET_DVR_RTSP_TEARDOWNSERVERERR	446	RTSP TEARDOWN 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

7 结构体说明

7.1 宏定义

宏定义	宏定义值	含义
NAME_LEN	32	用户名长度
PASSWD_LEN	16	密码长度
SERIALNO_LEN	48	序列号长度
MACADDR_LEN	6	MAC 地址长度
MAX_NAMELEN	16	DVR 本地登陆名长度
MAX_DOMAIN_NAME	64	最大域名长度
MAX_CYCLE_CHAN_V30	64	最大轮巡通道数（扩展）
MAX_DAYS	7	每周的天数
MAX_UNITEDMATRIX_NUM	8	级联中最多视频综合平台数量
MAX_OPTICALFIBER_NUM	16	最大光纤条数
MAX_ETHERNET	2	设备可配以太网络个数
MAX_DECODECHANNUM	32	最大解码通道数
LOG_INFO_LEN	11840	日志附加信息长度
MAX_ANALOG_ALARMOUT	32	最大模拟报警输出路数
MAX_IP_ALARMOUT	64	允许加入的最多 IP 报警输出路数
MAX_ALARMOUT_V30	96	(MAX_ANALOG_ALARMOUT + MAX_IP_ALARMOUT)
MAX_ALARMOUT_V40	4128	最大报警输出口（模拟通道+IP 通道）个数扩展
MAX_TIMESEGMENT_V30	8	最大时间段数
IPC_PROTOCOL_NUM	50	IPC 协议最大个数
MAX_EXCEPTIONNUM_V30	32	设备最大异常处理数
MAX_USERNUM_V30	32	设备最大用户数
MAX_ANALOG_CHANNUM	32	最大模拟通道个数
MAX_IP_CHANNEL	32	允许加入的最多 IP 通道数
MAX_CHANNUM_V30	64	(MAX_ANALOG_CHANNUM + MAX_IP_CHANNEL)
MAX_CHANNUM_V40	512	最大通道（模拟通道+IP 通道）个数扩展
MAX_RIGHT	32	设备支持的权限（1-12 表示本地权限，13-32 表示远程权限）
MAX_DISKNUM_V30	33	设备支持的最大硬盘数，最多 33 个硬盘(包括 16 个内置 SATA 硬盘、1 个 eSATA 硬盘和 16 个 NFS 盘)
MAX_PRESET_V30	256	设备支持的最大云台预置点数

MAX_TRACK_V30	256	设备支持的最大云台轨迹数
MAX_CRUISE_V30	256	设备支持的最大云台巡航数
STREAM_ID_LEN	32	流 ID 标识长度
MAX_DISPLAY_NUM	512	最大显示输出个数
MAX_NETWORK_CARD	4	设备可配最大网卡数目
MAX_OSD_LEN	64	输出口 OSD 长度
DESC_LEN	16	云台描述字符串长度
MAX_SERIAL_PORT	8	设备支持的最大 232 串口数
MAX_SCREEN_ADDRESS_LEN	16	特征码最大长度
NAME_LEN	32	用户名长度
PASSWD_LEN	16	密码长度
MATRIX_MAX_OUTPUT_NUM	256	矩阵最大输出通道个数
MAX_WINDOWS_V41	16	最大窗口数(v4.1 版本 SDK)
MAX_DISPNUM_V41	32	最大显示通道个数
MAX_ANALOG_ALARMIN	32	最大 32 路模拟报警输入
MAX_VERSION_LENGTH 8	8	最大版本号长度
MAX_CAM_COUNT	24	摄像机个数

7.2 NET_DVR_ADC_CFG: ADC 参数结构体

```

struct{
    BYTE    byGainR;
    BYTE    byGainG;
    BYTE    byGainB;
    BYTE    byOffsetR;
    BYTE    byOffsetG;
    BYTE    byOffsetB;
    BYTE    byRes[6];
}NET_DVR_ADC_CFG,*LPNET_DVR_ADC_CFG;

```

Members

byGainR

增益 R, 取值范围: 0~100

byGainG

增益 G, 取值范围: 0~100

byGainB

增益 B, 取值范围: 0~100

byOffsetR

偏移量 R, 取值范围: 0~100

byOffsetG

偏移量 G，取值范围：0~100

byOffsetB

偏移量 B，取值范围：0~100

byRes

保留，置为 0

See Also

[NET_DVR_SCREEN_DISPLAY_CFG](#)、[NET_DVR_MSC_SCREEN_PARAM](#)

7.3 NET_DVR_ALARMER:报警设备信息

```
struct{
    BYTE    byUserIDValid;
    BYTE    bySerialValid;
    BYTE    byVersionValid;
    BYTE    byDeviceNameValid;
    BYTE    byMacAddrValid;
    BYTE    byLinkPortValid;
    BYTE    byDeviceIPValid;
    BYTE    bySocketIPValid;
    LONG    lUserID;
    BYTE    sSerialNumber[SERIALNO_LEN];
    DWORD   dwDeviceVersion;
    char    sDeviceName[NAME_LEN];
    BYTE    byMacAddr[MACADDR_LEN];
    WORD    wLinkPort;
    char    sDeviceIP[128];
    char    sSocketIP[128];
    BYTE    byIpProtocol;
    BYTE    byRes2[11];
}NET_DVR_ALARMER,*LPNET_DVR_ALARMER;
```

Members

byUserIDValid

userid 是否有效：0- 无效，1- 有效

bySerialValid

序列号是否有效：0- 无效，1- 有效

byVersionValid

版本号是否有效：0- 无效，1- 有效

byDeviceNameValid

设备名字是否有效：0- 无效，1- 有效

byMacAddrValid

MAC 地址是否有效：0- 无效，1- 有效

byLinkPortValid

Login 端口是否有效：0- 无效，1- 有效

byDeviceIPValid

设备 IP 是否有效：0- 无效，1- 有效

bySocketIPValid

Socket IP 是否有效：0-无效，1-有效

lUserID

NET_DVR_Login 或 NET_DVR_Login_V30 返回值，布防时有效

sSerialNumber

序列号

dwDeviceVersion

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

sDeviceName

设备名称

byMacAddr

MAC 地址

wLinkPort

设备通讯端口

sDeviceIP

设备 IP 地址

sSocketIP

报警主动上传时的 Socket IP 地址

byIpProtocol

IP 协议：0- IPV4，1- IPV6

byRes2

保留，置为 0

See Also

[NET_DVR_SetDVRMessageCallBack_V30](#)、[NET_DVR_StartListen_V30](#)

7.4 NET_DVR_AREA_ZOOM_CFG:区域放大还原参数结构体

```
struct{
    BYTE                byCmd;
    BYTE                byRes[3];
    NET\_DVR\_RECTCFG    struArea;
}NET_DVR_AREA_ZOOM_CFG,*LPNET_DVR_AREA_ZOOM_CFG;
```

Members

byCmd

区域放大还原指令：1- 放大，2- 还原

byRes

保留，置为 0

struArea

需要放大的区域，以 1920*1920 作为屏幕总大小，还原时此参数无效

See Also

[NET_DVR_SCREEN_CONTROL_PARAM](#)

7.5 NET_DVR_ASSOCIATE_INPUT_PARAM: 关联的输入设备信息结构体

```
struct{
    BYTE        byEnable;
    BYTE        byRes1[3];
    DWORD       dwDeviceIndex;
    WORD        wInputIndex;
    BYTE        byRes2[18];
}NET_DVR_ASSOCIATE_INPUT_PARAM,*LPNET_DVR_ASSOCIATE_INPUT_PARAM;
```

Members

byEnable

是否启用：0- 否，1- 是

byRes1

保留

dwDeviceIndex

设备序号

wInputIndex

输入信号源

byRes2

保留

Remarks

对于 LCD 拼接屏、SDI 矩阵切换器，dwDeviceIndex、wInputIndex 均无效，设为 0。

See Also

[NET_DVR_VCS_USER_INFO](#)

7.6 NET_DVR_ASSOCIATE_OUTPUT_PARAM: 关联的输出显示屏信息结构体

```
struct{
    BYTE        byEnable;
    BYTE        byWallNo;
    BYTE        byRes1[2];
    DWORD       dwOutputIndex;
    BYTE        byRes2[20];
}NET_DVR_ASSOCIATE_OUTPUT_PARAM,*LPNET_DVR_ASSOCIATE_OUTPUT_PARAM;
```

Members

byEnable

是否启用：0- 否，1- 是

byWallNo

电视墙墙号

byRes1

保留

dwOutputIndex
输出显示屏的序号

byRes2
保留

See Also

[NET_DVR_VCS_USER_INFO](#)

7.7 NET_DVR_BOOT_LOGO_CFG:开机 LOGO 参数结构体

```
struct{
    DWORD    dwSize;
    BYTE     byEnable;
    BYTE     byRes[31];
}NET_DVR_BOOT_LOGO_CFG,*LPNET_DVR_BOOT_LOGO_CFG;
```

Members

dwSize
结构体大小

byEnable
开机是否显示 LOGO: 0- 不显示, 1- 显示

byRes
保留, 置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)

7.8 NET_DVR_COLOR_TEMPERATURE_CFG:色温参数结构体

```
struct{
    BYTE     byRed;
    BYTE     byGreen;
    BYTE     byBlue;
    BYTE     byRedOffset;
    BYTE     byGreenOffset;
    BYTE     byBlueOffset;
    BYTE     byRes[6];
}NET_DVR_COLOR_TEMPERATURE_CFG,*LPNET_DVR_COLOR_TEMPERATURE_CFG;
```

Members

byGainR
红色, 取值范围: 0~100

byGainG
绿色, 取值范围: 0~100

byGainB
蓝色, 取值范围: 0~100

byOffsetR

红色偏移，取值范围：0~100

byOffsetG

绿色偏移，取值范围：0~100

byOffsetB

蓝色偏移，取值范围：0~100

byRes

保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)、[NET_DVR_SCREEN_DISPLAY_CFG](#)

7.9 NET_DVR_COMPRESSIONCFG_V30:通道压缩参数结构体

```
struct{
    DWORD                dwSize;
    NET\_DVR\_COMPRESSION\_INFO\_V30    struNormHighRecordPara;
    NET\_DVR\_COMPRESSION\_INFO\_V30    struRes;
    NET\_DVR\_COMPRESSION\_INFO\_V30    struEventRecordPara;
    NET\_DVR\_COMPRESSION\_INFO\_V30    struNetPara;
}NET_DVR_COMPRESSIONCFG_V30, *LPNET_DVR_COMPRESSIONCFG_V30;
```

Members

dwSize

结构体大小

struRecordPara

录像/事件触发录像的码流压缩参数

struNetPara

网传的码流压缩参数

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.10 NET_DVR_COMPRESSION_INFO_V30:码流压缩参数

```
struct{
    BYTE    byStreamType;
    BYTE    byResolution;
    BYTE    byBitrateType;
    BYTE    byPicQuality;
    DWORD   dwVideoBitrate;
    DWORD   dwVideoFrameRate;
    WORD    wIntervalFrameI;
    BYTE    byIntervalBPFrame;
    BYTE    byres1;
    BYTE    byVideoEncType;
    BYTE    byAudioEncType;
```

BYTE byVideoEncComplexity;

BYTE byres[9];

}NET_DVR_COMPRESSION_INFO_V30, *LPNET_DVR_COMPRESSION_INFO_V30;

Members

byStreamType

码流类型：0-视频流，1-复合流

byResolution

分辨率：0-DCIF(528*384/528*320)，1-CIF(352*288/352*240)，2-QCIF(176*144/176*120)，3-4CIF(704*576/704*480)或 D1(720*576/720*486)，4-2CIF(704*288/704*240)，6-QVGA(320*240)，7-QQVGA(160*120)，12-384*288，13-576*576，16-VGA(640*480)，17-UXGA(1600*1200)，18-SVGA(800*600)，19-HD720P(1280*720)，20-XVGA(1280*960)，21-HD900P(1600*900)，23-1536*1536，24-1920*1920，27-1920*1080p，28-2560*1920，29-1600*304，30-2048*1536，31-2448*2048，32-2448*1200，33-2448*800，34-XGA(1024*768)，35-SXGA(1280*1024)，36-WD1(960*576/960*480)，37-1080i(1920*1080)，38-WXGA(1440*900)，39-HD_F(1920*1080/1280*720)，40-HD_H(1920*540/1280*360)，41-HD_Q(960*540/630*360)，42-2336*1744，43-1920*1456，44-2592*2048，45-3296*2472，46-1376*768，47-1366*768，48-1360*768，49-WSXGA+，50-720*720,51-1280*1280，52-2048*768，53-2048*2048，54-2560*2048，55-3072*2048，56-2304*1296，57-WXGA(1280*800)，58-1600*600，59-1600*900，60-2752*2208，0xff-Auto(使用当前码流分辨率)

byBitrateType

码率类型：0-变码率，1-定码率

byPicQuality

图象质量：0-最好，1-次好，2-较好，3-一般，4-较差，5-差

dwVideoBitrate

码率：0-保留，1-16K(保留)，2-32K，3-48k，4-64K，5-80K，6-96K，7-128K，8-160k，9-192K，10-224K，11-256K，12-320K，13-384K，14-448K，15-512K，16-640K，17-768K，18-896K，19-1024K，20-1280K，21-1536K，22-1792K，23-2048K，24-3072K，25-4096K，26-8192K，27-16384K。

最高位(31位)置成1表示是自定义码流，0~30位表示码流值，最小值16k

dwVideoFrameRate

帧率：0-全部，1-1/16，2-1/8，3-1/4，4-1/2，5-1，6-2，7-4，8-6，9-8，10-10，11-12，12-16，13-20，14-15，15-18，16-22

wIntervalFrameI

I 帧间隔，0xffff-无效

byIntervalBPFrame

帧格式：0-BBP 帧，1-BP 帧，2-单 P 帧，0xff-无效

byres1

保留，置为 0

byVideoEncType

视频编码类型：0-私有 264，1-标准 h264，2-标准 mpeg4，7-M-JPEG，0xff-无效

byAudioEncType

音频编码类型：0-OggVorbis，1-G711_U，2-G711_A，6-G726，0xff-无效

byVideoEncComplexity

视频编码复杂度：0-低，1-中，2-高

byres

保留，置为 0

Remarks

当修改设备视频编码类型时，需要重启设备生效。

See Also

[NET_DVR_COMPRESSIONCFG_V30](#)

7.11 NET_DVR_DAYTIME:时间参数结构体

```
struct{
    BYTE    byHour;
    BYTE    byMinute;
    BYTE    bySecond;
    BYTE    byRes;
    WORD    wMilliSecond;
    BYTE    byRes1[2];
}NET_DVR_DAYTIME, *LPNET_DVR_DAYTIME;
```

Members**byHour**

时，取值范围：0~24

byMinute

分，取值范围：0~60

bySecond

秒，取值范围：0~60

byRes

保留，置为 0

wMilliSecond

毫秒，取值范围：0~1000

byRes1

保留，置为 0

See Also

[NET_DVR_SWITCH_DAY_TIME](#)、

7.12 NET_DVR_DDNS_ADDRESS:设备 DDNS 域名信息

```
struct{
    BYTE    byDevAddress[MAX_DOMAIN_NAME];
    BYTE    byDevDdns[MAX_DOMAIN_NAME];
    BYTE    byDdnsType;
    BYTE    byRes1[3];
    WORD    wDevPort;
    WORD    wDdnsPort;
    BYTE    byres[64];
}NET_DVR_DDNS_ADDRESS, *LPNET_DVR_DDNS_ADDRESS;
```

Members

byDevAddress

设备域名

byDevDdns

DDNS 服务器地址

byDdnsType

域名服务器类型: 0- IPServer, 1- Dyndns, 2- PeanutHull(花生壳), 3- NO-IP, 4- hiDDNS

byRes1

保留

wDevPort

设备端口号

wDdnsPort

DDNS 服务器端口

byres

保留

See Also

[NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX](#)

7.13 NET_DVR_DEC_DDNS_DEV:动态域名取流配置

```
struct{
    NET\_DVR\_DEV\_DDNS\_INFO          struDdnsInfo;
    NET\_DVR\_STREAM\_MEDIA\_SERVER    struMediaServer;
}NET_DVR_DEC_DDNS_DEV,*LPNET_DVR_DEC_DDNS_DEV;
```

Members

struDdnsInfo

动态域名参数配置

struMediaServer

流媒体服务器配置参数

See Also

[NET_DVR_DEC_STREAM_MODE](#)

7.14 NET_DVR_DEC_STREAM_DEV_EX:设备取流配置

```
struct{
    NET\_DVR\_STREAM\_MEDIA\_SERVER    struStreamMediaSvrCfg;
    NET\_DVR\_DEV\_CHAN\_INFO\_EX       struDevChanInfo;
}NET_DVR_DEC_STREAM_DEV_EX,*LPNET_DVR_DEC_STREAM_DEV_EX;
```

Members

struStreamMediaSvrCfg

流媒体服务器配置参数

struDevChanInfo

设备通道配置参数

See Also

[NET_DVR_DEC_STREAM_MODE](#)**7.15 NET_DVR_DEC_STREAM_MODE:取流模式配置**

```
union{
    NET\_DVR\_DEC\_STREAM\_DEV\_EX    struDecStreamDev;
    NET\_DVR\_PU\_STREAM\_URL        struUrlInfo;
    NET\_DVR\_DEC\_DDNS\_DEV        struDdnsDecInfo;
    BYTE                          byRes[300];
}NET_DVR_DEC_STREAM_MODE,*LPNET_DVR_DEC_STREAM_MODE;
```

Members

struDecStreamDev

通过 IP 地址或者域名从设备或者流媒体服务器取流

struUrlInfo

通过 URL 从设备或者流媒体服务器取流

struDdnsDecInfo

通过动态域名解析向设备取流

byRes

保留，置为 0

See Also[NET_DVR_PU_STREAM_CFG_V41](#)、[NET_DVR_MATRIX_CHAN_INFO_V41](#)[NET_DVR_MATRIX_DEC_CHAN_INFO_V41](#)**7.16 NET_DVR_DECODER_WORK_STATUS_V41:解码设备状态参数结构体**

```
struct{
    DWORD                          dwSize;
    NET\_DVR\_MATRIX\_CHAN\_STATUS    struDecChanStatus[MAX_DECODECHANNUM];
    NET\_DVR\_DISP\_CHAN\_STATUS\_V41 struDispChanStatus[MAX_DISPNUM_V41];
    BYTE                           byAlarmInStatus[MAX_ANALOG_ALARMIN];
    BYTE                           byAalarmOutStatus[MAX_ANALOG_ALARMOUT];
    BYTE                           byAudioInChanStatus;
    BYTE                           byRes[127];
}NET_DVR_DECODER_WORK_STATUS_V41,*LPNET_DVR_DECODER_WORK_STATUS_V41;
```

Members

dwSize

结构体大小

struDecChanStatus

解码通道状态

struDispChanStatus

显示通道状态

byAlarmInStatus
报警输入状态

byAlarmOutStatus
报警输出状态

byAudioInChanStatus
语音对讲状态

byRes
保留，置为 0

See Also

[NET_DVR_MatrixGetDeviceStatus_V41](#)

7.17 NET_DVR_DEFOG_LCD:去雾参数结构体

```
struct{
    DWORD    dwSize;
    BYTE     byDefogEnable;
    BYTE     byDefogModel;
    BYTE     byDefogLevel;
    BYTE     byRes[33];
}NET_DVR_DEFOG_LCD,*LPNET_DVR_DEFOG_LCD;
```

Members

dwSize
结构体大小

byDefogEnable
开启去雾开关：0-关闭，1-开启

byDefogModel
去雾模式：0-自动去雾，1-手动去雾

byDefogLevel
去雾强度等级，1~7 个等级，默认等级为 4。等级 1 为 0.5，等级 2 为 0.55，等级 3 为 0.60，等级 4 为 0.65，等级 5 为 0.70，等级 6 为 0.75，等级 7 为 0.80（去雾模式为手动去雾时，可配置去雾强度等级）

byRes
保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)

7.18 NET_DVR_DELAY_TIME:延时开机参数结构体

```
struct{
    DWORD    dwSize;
    DWORD    dwDelayTime;
    BYTE     byRes[32];
}NET_DVR_DELAY_TIME,*LPNET_DVR_DELAY_TIME;
```


Members**dwSize**

结构体大小

dwDelayTime

延时时间，取值范围：0~3000，单位：ms

byRes

保留，置为 0

See Also[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)**7.19 NET_DVR_DEV_CHAN_INFO_EX:前端编码设备信息**

```

struct{
    BYTE        byChanType;
    BYTE        byStreamId[STREAM_ID_LEN];
    BYTE        byRes1[3];
    DWORD       dwChannel;
    BYTE        byRes2[24];
    BYTE        byAddress[MAX_DOMAIN_NAME];
    WORD        wDVRPort;
    BYTE        byChannel;
    BYTE        byTransProtocol;
    BYTE        byTransMode;
    BYTE        byFactoryType;
    BYTE        byDeviceType;
    BYTE        byDispChan;
    BYTE        bySubDispChan;
    BYTE        byResolution;
    BYTE        byRes[2];
    BYTE        sUserName[NAME_LEN];
    BYTE        sPassword[PASSWD_LEN];
}NET_DVR_DEV_CHAN_INFO_EX,*LPNET_DVR_DEV_CHAN_INFO_EX;

```

Members**byChanType**

通道类型：0-普通通道，1-零通道，2-流 ID，3-本地输入源

byStreamId流 ID，通道类型 **byChanType** 为 2 时有效**byRes1**

保留，置为 0

dwChannel通道号，通道类型 **byChanType** 为 0、1、3 时有效**byRes2**

保留，置为 0

byAddress

设备域名

wDVRPort

端口号

byChannel

该参数无效，通道号见 dwChannel

byTransProtocol

传输协议类型：0- TCP，1- UDP

byTransMode

传输码流模式：0- 主码流，1- 子码流

byFactoryType

前端设备厂家类型，通过接口 NET_DVR_GetIPProtoList 获取

byDeviceType

设备类型(视频综合平台智能板使用)：1- 解码器（此时根据视频综合平台能力集中

byVcaSupportChanMode 字段来决定是使用解码通道还是显示通道），2- 编码器，3- X86 高清输入

byDispChan

显示通道号，智能配置使用

bySubDispChan

显示通道子通道号，智能配置时使用

byResolution

1- CIF，2- 4CIF，3- 720P，4- 1080P，5- 500w，大屏控制器使用，根据该参数分配解码资源

byRes

保留，置为 0

sUserName

监控主机登陆帐号

sPassword

监控主机密码

See Also

[NET_DVR_DEC_STREAM_DEV_EX](#)

7.20 NET_DVR_DEV_DDNS_INFO:动态域名参数配置

```
struct{
    BYTE        byDevAddress[MAX_DOMAIN_NAME];
    BYTE        byTransProtocol;
    BYTE        byTransMode;
    BYTE        byDdnsType;
    BYTE        byRes1;
    BYTE        byDdnsAddress[MAX_DOMAIN_NAME];
    WORD        wDdnsPort;
    BYTE        byChanType;
    BYTE        byFactoryType;
    DWORD       dwChannel;
    BYTE        byStreamId[STREAM_ID_LEN];
    BYTE        sUserName[NAME_LEN];
}
```

```

BYTE      sPassword[PASSWD_LEN];
WORD      wDevPort;
BYTE      byRes2[2];
}NET_DVR_DEV_DDNS_INFO,*LPNET_DVR_DEV_DDNS_INFO;

```

Members

byDevAddress

设备域名(IPServer 或 hiDDNS 时可填设备序列号或者别名)

byTransProtocol

传输协议类型: 0- TCP, 1- UDP, 2- 多播

byTransMode

传输码流模式: 0- 主码流, 1- 子码流

byDdnsType

域名服务器类型: 0- IPServer, 1- DynDNS, 2- PeanutHull(花生壳), 3- NO-IP, 4- hiDDNS

byRes1

保留

byDdnsAddress

DDNS 服务器地址

wDdnsPort

DDNS 服务器端口号

byChanType

通道类型: 0-普通通道, 1-零通道, 2-流 ID

byFactoryType

前端设备厂家类型, 通过接口 NET_DVR_GetIPCProtoList 获取

dwChannel

设备通道号

byStreamId

流 ID

sUserName

设备登陆帐号

sPassword

设备密码

wDVRPort

设备端口号

byRes2

保留

See Also

[NET_DVR_DEC_DDNS_DEV](#)

7.21 NET_DVR_DEV_IP_INFO:屏幕 IP 信息结构体

```

struct{
    BYTE      byUserName[NAME_LEN];
    BYTE      byPassWord[PASSWD_LEN];
    NET\_DVR\_IPADDR    strulPAddr;
}

```

```

WORD            wPort;
BYTE            byRes[100];
}NET_DVR_DEV_IP_INFO,*LPNET_DVR_DEV_IP_INFO;

```

Members

```

byUserName      用户名
byPassWord     密码
strulPAddr     IP 地址
wPort          设备端口号
byRes          保留，置为 0

```

See Also

[NET_DVR_SCREEN_BASE_INFO](#)

7.22 NET_DVR_DEVICECFG_V40:设备参数

```

struct{
    DWORD    dwSize;
    BYTE     sDVRName[NAME_LEN];
    DWORD    dwDVRID;
    DWORD    dwRecycleRecord;
    BYTE     sSerialNumber[SERIALNO_LEN];
    DWORD    dwSoftwareVersion;
    DWORD    dwSoftwareBuildDate;
    DWORD    dwDSPSoftwareVersion;
    DWORD    dwDSPSoftwareBuildDate;
    DWORD    dwPanelVersion;
    DWORD    dwHardwareVersion;
    BYTE     byAlarmInPortNum;
    BYTE     byAlarmOutPortNum;
    BYTE     byRS232Num;
    BYTE     byRS485Num;
    BYTE     byNetworkPortNum;
    BYTE     byDiskCtrlNum;
    BYTE     byDiskNum;
    BYTE     byDVRType;
    BYTE     byChanNum;
    BYTE     byStartChan;
    BYTE     byDecordChans;
    BYTE     byVGANum;
    BYTE     byUSBNum;

```

```

BYTE    byAuxoutNum;
BYTE    byAudioNum;
BYTE    byIPChanNum;
BYTE    byZeroChanNum;
BYTE    bySupport;
BYTE    byEsataUseage;
BYTE    byIPCPlug;
BYTE    byStorageMode;
BYTE    bySupport1;
WORD    wDevType;
BYTE    byDevTypeName[24];
BYTE    byRes2[16];
}NET_DVR_DEVICECFG_V40,*LPNET_DVR_DEVICECFG_V40;

```

Members

dwSize

结构体大小

sDVRName

设备名称

dwDVRID

设备 ID 号，用于遥控器，v1.4 的设备号范围为(0-99), v1.5 及以上版本的设备号为(0-255)

dwRecycleRecord

是否循环录像：0—不是；1—是

以下参数不可更改

sSerialNumber

设备序列号

dwSoftwareVersion

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

dwSoftwareBuildDate

软件生成日期，0xYYYYMMDD

dwDSPSoftwareVersion

DSP 软件版本，高 16 位是主版本，低 16 位是次版本

dwDSPSoftwareBuildDate

DSP 软件生成日期，0xYYYYMMDD

dwPanelVersion

前面板版本，高 16 位是主版本，低 16 位是次版本

dwHardwareVersion

硬件版本，高 16 位是主版本，低 16 位是次版本

byAlarmInPortNum

设备模拟报警输入个数

byAlarmOutPortNum

设备模拟报警输出个数

byRS232Num

设备 232 串口个数

- byRS485Num**
设备 485 串口个数
- byNetworkPortNum**
网络口个数
- byDiskCtrlNum**
硬盘控制器个数
- byDiskNum**
硬盘个数
- byDVRType**
设备类型，详见“Remarks”说明
- byChanNum**
设备模拟通道个数
- byStartChan**
模拟通道的起始通道号
- byDecordChans**
设备解码路数
- byVGANum**
VGA 口的个数
- byUSBNum**
USB 口的个数
- byAuxoutNum**
辅口的个数
- byAudioNum**
语音口的个数
- byIPChanNum**
最大数字通道
- byZeroChanNum**
零通道编码个数
- bySupport**
能力，位与结果为 0 表示不支持，1 表示支持
bySupport & 0x1，表示是否支持智能搜索
bySupport & 0x2，表示是否支持备份
bySupport & 0x4，表示是否支持压缩参数能力获取
bySupport & 0x8，表示是否支持双网卡
bySupport & 0x10，表示支持远程 SADP
bySupport & 0x20，表示支持 Raid 卡功能
bySupport & 0x40，表示支持 IPSAN 搜索
bySupport & 0x80，表示支持 rtp over rtsp
- byEsataUseage**
Esata 的默认用途：0-默认备份，1-默认录像
- byIPCPlug**
0-不支持即插即用，1-支持即插即用
- byStorageMode**
存储模式：0-盘组模式，1-磁盘配额，2-抽帧模式

bySupport1

能力集扩充，位与结果为 0 表示不支持，1 表示支持

bySupport1 & 0x1, 表示是否支持 snmp v30

bySupport1 & 0x2, 支持区分回放和下载

wDevType

设备型号，详见“Remarks”说明

byDevTypeName

设备型号名称

byRes2

保留，置为 0

Remarks

如果 byDVRType 是 0, 则接口中解析 wDevType 作为设备型号; 如果 byDVRType 非 0, 则接口中 byDVRType 和 wDevType 值相等, 都是 byDVRType。推荐使用 wDevType 作为设备类型。

See Also

[NET_DVR_LOCAL_BYTE_ENCODE_CONVERT](#)、[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.23 NET_DVR_DEVICEINFO_V30:设备参数结构体

```

struct{
    BYTE        sSerialNumber[SERIALNO_LEN];
    BYTE        byAlarmInPortNum;
    BYTE        byAlarmOutPortNum;
    BYTE        byDiskNum;
    BYTE        byDVRType;
    BYTE        byChanNum;
    BYTE        byStartChan;
    BYTE        byAudioChanNum;
    BYTE        byIPChanNum;
    BYTE        byZeroChanNum;
    BYTE        byMainProto;
    BYTE        bySubProto;
    BYTE        bySupport;
    BYTE        bySupport1;
    BYTE        bySupport2;
    WORD        wDevType;
    BYTE        bySupport3;
    BYTE        byMultiStreamProto;
    BYTE        byStartDChan;
    BYTE        byStartDTalkChan;
    BYTE        byHighDChanNum;
    BYTE        bySupport4;
    BYTE        byLanguageType;
    BYTE        byRes2[9];
}NET_DVR_DEVICEINFO_V30,*LPNET_DVR_DEVICEINFO_V30;

```

Members

sSerialNumber

序列号

byAlarmInPortNum

报警输入个数

byAlarmOutPortNum

报警输出个数

byDiskNum

硬盘个数

byDVRType

设备类型，详见“Remarks”说明

byChanNum

设备模拟通道个数

byStartChan

起始通道号，目前设备通道号从 1 开始

byAudioChanNum

设备语音通道数

byIPChanNum

设备最大数字通道个数

byZeroChanNum

零通道编码个数

byMainProto

主码流传输协议类型：0-private，1-rtsp

bySubProto

子码流传输协议类型：0-private，1-rtsp

bySupport

能力，位与结果为 0 表示不支持，1 表示支持

bySupport & 0x1，表示是否支持智能搜索

bySupport & 0x2，表示是否支持备份

bySupport & 0x4，表示是否支持压缩参数能力获取

bySupport & 0x8，表示是否支持双网卡

bySupport & 0x10，表示支持远程 SADP

bySupport & 0x20，表示支持 Raid 卡功能

bySupport & 0x40，表示支持 IPSAN 目录查找

bySupport & 0x80，表示支持 rtp over rtsp

bySupport1

能力集扩充，位与结果为 0 表示不支持，1 表示支持

bySupport1 & 0x1，表示是否支持 snmp v30

bySupport1 & 0x2，表示是否支持区分回放和下载

bySupport1 & 0x4，表示是否支持布防优先级

bySupport2

能力集扩充，位与结果为 0 表示不支持，1 表示支持

bySupport2 & 0x1，表示解码器是否支持通过 URL 取流解码

bySupport2 & 0x2，表示是否支持 FTPV40

bySupport2 & 0x4, 表示是否支持 ANR(断网录像)

wDevType

设备型号, 详见“Remarks”说明

bySupport3

能力集扩展, 位与结果: 0- 不支持, 1- 支持

bySupport3 & 0x1, 表示是否支持多码流

bySupport3 & 0x4, 表示是否支持按组配置, 具体包含通道图像参数、报警输入参数、IP 报警输入/输出接入参数、用户参数、设备工作状态、JPEG 抓图、定时和时间抓图、硬盘盘组管理等

bySupport3 & 0x20, 表示是否支持通过 DDNS 域名解析取流

byMultiStreamProto

是否支持多码流, 按位表示, 位与结果: 0-不支持, 1-支持

byMultiStreamProto & 0x1, 表示是否支持码流 3

byMultiStreamProto & 0x2, 表示是否支持码流 4

byMultiStreamProto & 0x40,表示是否支持主码流

byMultiStreamProto & 0x80,表示是否支持子码流

byStartDChan

起始数字通道号, 0 表示无数字通道, 比如 DVR 或 IPC

byStartDTalkChan

起始数字对讲通道号, 区别于模拟对讲通道号, 0 表示无数字对讲通道

byHighDChanNum

数字通道个数, 高 8 位

bySupport4

能力集扩展, 按位表示, 位与结果: 0- 不支持, 1- 支持

bySupport4 & 0x4 表示是否支持拼控统一接口

byLanguageType

支持语种能力, 按位表示, 位与结果: 0- 不支持, 1- 支持

byLanguageType ==0, 表示老设备, 不支持该字段

byLanguageType & 0x1, 表示是否支持中文

byLanguageType & 0x2, 表示是否支持英文

byRes2

保留, 置为 0

Remarks

如果 byDVRType 是 0, 则接口中解析 wDevType 作为设备型号; 如果 byDVRType 非 0, 则接口中 byDVRType 和 wDevType 值相等, 都是 byDVRType。推荐使用 wDevType 作为设备类型。

See Also

[NET_DVR_Login_V30](#)、[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.24 NET_DVR_DISP_CHAN_STATUS_V41:显示通道状态结构体

```
struct{
    BYTE    byDispStatus;
    BYTE    byBVGa;
    BYTE    byVideoFormat;
    BYTE    byWindowMode;
```

```

BYTE    byJoinDecChan[MAX_WINDOWS_V41];
BYTE    byFpsDisp[MAX_WINDOWS_V41];
BYTE    byScreenMode;
BYTE    byRes1[3];
DWORD   dwDispChan;
BYTE    byRes2[24];
}NET_DVR_DISP_CHAN_STATUS_V41,*LPNET_DVR_DISP_CHAN_STATUS_V41;

```

Members

byDispStatus

显示状态：0-未显示，1-启动显示

byBVGA

显示接口，0-BNC 输出，1-VGA 输出，2-HDMI 输出，3-DVI 输出，0xff-无效

byVideoFormat

视频制式，具体定义如下：

```

enum{
    VS_NON   = 0,           //无
    VS_NTSC  = 1,           //NTSC 制式
    VS_PAL   = 2           //PAL 制式
}VIDEO_STANDARD

```

byWindowMode

当前画面模式

byJoinDecChan

各个子窗口关联的解码通道

byFpsDisp

每个子画面的显示帧率

byScreenMode

屏幕模式：0- 普通，1- 大屏

byRes1

保留，置为 0

dwDispChan

显示通道号，获取全部显示通道状态时有效，设置时可填 0

byRes2

保留，置为 0

See Also

[NET_DVR_DECODER_WORK_STATUS_V41](#)

7.25 NET_DVR_DISPLAY_COLOR_CTRL:显示单元颜色控制结构体

```

struct{
    BYTE    byColorType;
    BYTE    byScale;
    BYTE    byRes[14];
}NET_DVR_DISPLAY_COLOR_CTRL,*LPNET_DVR_DISPLAY_COLOR_CTRL;

```

Members

byColorType

类型：1- 亮度，2- 对比度，3- 饱和度，4- 清晰度

byScale

控制取值：-1 或者+1

byRes

保留

See Also

[NET_DVR_SCREEN_CONTROL_PARAM](#)

7.26 NET_DVR_DISPLAY_POSITION_CTRL:显示单元位置控制结构体

```
struct{
```

```
    BYTE    byPositionType;
```

```
    BYTE    byScale;
```

```
    BYTE    byRes[14];
```

```
}NET_DVR_DISPLAY_POSITION_CTRL,*LPNET_DVR_DISPLAY_POSITION_CTRL;
```

Members**byPositionType**

类型：1- 水平位置，2- 垂直位置

byScale

控制取值：-1 或者+1

byRes

保留

See Also

[NET_DVR_SCREEN_CONTROL_PARAM](#)

7.27 NET_DVR_ETHERNET_V30:以太网配置参数

```
struct{
```

```
    NET\_DVR\_IPADDR    struDVRIP;
```

```
    NET\_DVR\_IPADDR    struDVRIPMask;
```

```
    DWORD             dwNetInterface;
```

```
    WORD              wDVRPort;
```

```
    WORD              wMTU;
```

```
    BYTE              byMACAddr[MACADDR_LEN];
```

```
    BYTE              byRes[2];
```

```
}NET_DVR_ETHERNET_V30, *LPNET_DVR_ETHERNET_V30;
```

Members**struDVRIP**

设备 IP 地址

struDVRIPMask

设备 IP 地址掩码

dwNetInterface

网络接口: 1-10MBase-T; 2-10MBase-T 全双工; 3-100MBase-TX; 4-100M 全双工; 5-10M/100M/1000M 自适应; 6-1000M 全双工

wDVRPort

设备端口号

wMTU

MTU 设置, 默认 1500

byMACAddr

设备物理地址

byRes

保留

Remarks

MTU 的设置范围为 500-9676, 若 MTU 设置过小客户端将无法注册到设备, 并且客户端预览、回放、配置参数也会失败。

See Also

[NET_DVR_NETCFG_V30](#)、

7.28 NET_DVR_EXCEPTION_V30:异常参数

```
struct{
    DWORD dwSize;
    NET\_DVR\_HANDLEEXCEPTION\_V30 struExceptionHandleType[MAX_EXCEPTIONNUM_V30];
}NET_DVR_EXCEPTION_V30,*LPNET_DVR_EXCEPTION_V30;
```

Members

dwSize

结构体大小

struExceptionHandleType

异常信息处理方式:

数组 0—硬盘满

数组 1—硬盘出错

数组 2—网线断

数组 3—IP 地址冲

数组 4—非法访问

数组 5—输入/输出视频制式不匹配

数组 6—视频信号异常

数组 7—录像异常

数组 8—阵列异常

数组 9—前端/录像分辨率不匹配异常

数组 10—行车超速 (车载专用)

数组 11—热备异常 (N+1 使用)

数组 12—温度异常

数组 13—子系统异常

数组 14—风扇异常

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.29 NET_DVR_EXCEPTION_V40:异常参数配置（扩展）

```
struct{
    DWORD                dwSize;
    DWORD                dwMaxGroupNum;
    NET\_DVR\_HANDLEEXCEPTION\_V41    struExceptionHandle[MAX_EXCEPTIONNUM_V30];
    BYTE                byRes[128];
}NET_DVR_EXCEPTION_V40,*LPNET_DVR_EXCEPTION_V40;
```

Members

dwSize

结构体大小

dwMaxGroupNum

设备支持的异常类型最大组数（只读）

struExceptionHandle

异常信息处理方式，数组的每个元素都表示一种异常类型：

数组 0—硬盘满

数组 1—硬盘出错

数组 2—网线断

数组 3—IP 地址冲突

数组 4—非法访问

数组 5—输入/输出视频制式不匹配

数组 6—视频信号异常

数组 7—录像异常

数组 8—阵列异常

数组 9—前端/录像分辨率不匹配异常

数组 10-行车超速（车载专用）

数组 11-热备异常（N+1 使用）

数组 12-温度异常

数组 13-子系统异常

数组 14-风扇异常

数组 15-POE 供电异常

byRes

保留，置为 0

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.30 NET_DVR_EXTERNAL_MATRIX_CFG: 矩阵信息结构体

```
struct{
    DWORD                dwSize;
    BYTE                byValid;
    BYTE                byRes1[3];
    BYTE                sMatrixName[NAME_LEN];
}
```

```

DWORD          dwMatrixID;
WORD           wMatrixInputChanNum;
WORD           wMatrixOutputChanNum;
WORD           wMatrixOutputChanRef[MAX_CAM_COUNT];
BYTE           byMatrixChanType;
BYTE           byMatrixProtocol;
BYTE           byMatrixType;
BYTE           byRes2;
NET_MATRIX_UNION struMatrixUnion;
BYTE           byRes3[128];

```

```
}NET_DVR_EXTERNAL_MATRIX_CFG,*LPNET_DVR_EXTERNAL_MATRIX_CFG;
```

Members

dwSize

结构体大小

byValid

是否有效：0-无效，1-有效，有效时为设置矩阵，无效时为删除矩阵

byRes1

保留，置为 0

sMatrixName

矩阵的名称

dwMatrixID

矩阵的 ID 号，获取所有矩阵配置时使用该字段，其他接口调用时该值无效

wMatrixInputChanNum

矩阵的输入通道数目

wMatrixOutputChanNum

矩阵的输出通道数目

wMatrixOutputChanRef

设备输入和矩阵输出的绑定关系，数组下标(+1)表示控制器设备输入通道号，元素的值表示矩阵的输出通道号，元素值大于 0 表示绑定矩阵输出通道，元素值为 0 时表示不绑定。对于 LCD 拼接屏（屏幕服务器），表示屏幕和矩阵输出的绑定关系，数组下标(+1)为屏幕索引序号（从 1 开始）。

byMatrixChanType

矩阵的通道类型：1-BNC，2-VGA，3-RGB，4-DVI

byMatrixProtocol

矩阵所用协议：1-ZT1.0，2- ZT2.0，3- Extron，4- Creator，5- 自定义

byMatrixType

矩阵类型：1-模拟矩阵，2-数字矩阵，集中式多屏控制器 V2.1 只支持模拟矩阵

byRes2

保留，置为 0

struMatrixUnion

模拟或者数字矩阵参数联合体

byRes3

保留，置为 0

Remarks

当矩阵协议为自定义时，通过透明通道接口发送自定义控制命令。

See Also

[NET_DVR_GetDeviceConfig](#)、[NET_DVR_SetDeviceConfig](#)

7.31 NET_DVR_HANDLEEXCEPTION:报警和异常处理结构体

```
struct{
    DWORD    dwHandleType;
    BYTE     byRelAlarmOut[MAX_ALARMOUT];
}NET_DVR_HANDLEEXCEPTION, *LPNET_DVR_HANDLEEXCEPTION;
```

Members

dwHandleType

处理方式:

0x00: 无响应

0x01: 监视器上警告

0x02: 声音警告

0x04: 上传中心

0x08: 触发报警输出

0x10: Jpeg 抓图并上传 EMail

byRelAlarmOut

报警触发的输出通道, 0-不触发, 1-触发输出

See Also

[NET_DVR_VILOST](#)、[NET_DVR_MOTION](#)、[NET_DVR_HIDEALARM](#)

7.32 NET_DVR_HANDLEEXCEPTION_V30:报警和异常处理

```
struct{
    DWORD    dwHandleType;
    BYTE     byRelAlarmOut[MAX_ALARMOUT_V30];
}NET_DVR_HANDLEEXCEPTION_V30, *LPNET_DVR_HANDLEEXCEPTION_V30;
```

Members

dwHandleType

处理方式:

0x00: 无响应

0x01: 监视器上警告

0x02: 声音警告

0x04: 上传中心

0x08: 触发报警输出

0x10: Jpeg 抓图并上传 Email

0x20: 无线声光报警器联动

0x40: 联动电子地图(目前仅 PCNVR 支持)

0x200: 抓图并上传 ftp

byRelAlarmOut

报警触发的输出通道, 0-不触发, 1-触发输出

See Also[NET_DVR_EXCEPTION_V30](#)**7.33 NET_DVR_HANDLEEXCEPTION_V41:异常参数信息结构体**

```
struct{
    DWORD    dwHandleType;
    DWORD    dwMaxAlarmOutChannelNum;
    DWORD    dwRelAlarmOut[MAX_ALARMOUT_V40];
    BYTE     byRes[64];
}NET_DVR_HANDLEEXCEPTION_V41,*LPNET_DVR_HANDLEEXCEPTION_V41;
```

Members**dwHandleType**

处理方式，各种异常处理方式的"或"结果，异常处理方式：

0x00: 无响应

0x01: 监视器上警告

0x02: 声音警告

0x04: 上传中心

0x08: 触发报警输出

0x10: Jpeg 抓图并上传 EMail

0x20: 无线声光报警器联动

0x40: 联动电子地图(目前仅 PCNVR 支持)

0x200: 抓图并上传 ftp E.g. dwHandleType==0x01|0x04 表示配置报警发生时联动监视器上警告并且将报警信息上传中心

0x400: 虚交侦测 联动 聚焦模式（提供可配置项，原先设备自动完成）IPC5.1.0

0x800: PTZ 联动跟踪(球机跟踪目标)

dwMaxAlarmOutChannelNum

设备最大支持的触发报警输出通道数（只读）

dwRelAlarmOut

触发报警输出通道，数组元素值表示报警输出号（从 0 开始），例如：dwRelAlarmOut[i]==3 表示触发第 4 个报警输出通道。中间遇到 0xffffffff 则后续无效。

byRes

保留，置为 0

See Also[NET_DVR_EXCEPTION_V40](#)**7.34 NET_DVR_HIDEALARM:遮挡报警参数结构体**

```
struct{
    DWORD    dwEnableHideAlarm;
    WORD     wHideAlarmAreaTopLeftX;
    WORD     wHideAlarmAreaTopLeftY;
    WORD     wHideAlarmAreaWidth;
```



```

WORD                                wHideAlarmAreaHeight;
NET\_DVR\_HANDLEEXCEPTION          strHideAlarmHandleType;
NET\_DVR\_SCHEDTIME                struAlarmTime[MAX_DAYS][MAX_TIMESEGMENT];
}NET_DVR_HIDEALARM, *LPNET_DVR_HIDEALARM;

```

Members

dwEnableHideAlarm

是否启动遮挡报警：0-否，1-低灵敏度，2-中灵敏度，3-高灵敏度

wHideAlarmAreaTopLeftX

遮挡区域的 x 坐标

wHideAlarmAreaTopLeftY

遮挡区域的 y 坐标

wHideAlarmAreaWidth

遮挡区域的宽

wHideAlarmAreaHeight

遮挡区域的高

strHideAlarmHandleType

处理方法

struAlarmTime

布防时间

Remarks

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

See Also[NET_DVR_PICCFG](#)**7.35 NET_DVR_HOLIDAY_PARAM_CFG:假日参数配置结构体**

```

struct{
    DWORD                                dwSize;
    NET\_DVR\_HOLIDAY\_PARAM                struHolidayParam[MAX_HOLIDAY_NUM];
    DWORD                                byRes[40];
}NET_DVR_HOLIDAY_PARAM_CFG, *LPNET_DVR_HOLIDAY_PARAM_CFG;

```

Members

dwSize

结构体大小

struHolidayParam

假日参数

byRes

保留

See Also[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.36 NET_DVR_HOLIDAY_PARAM:假日参数结构体

```
struct{
    BYTE                byEnable;
    BYTE                byDateMode;
    BYTE                byRes1[2];
    NET\_DVR\_HOLIDATE\_UNION uHolidate;
    BYTE                byName[NAME_LEN];
    BYTE                byRes2[20];
}NET_DVR_HOLIDAY_PARAM, *LPNET_DVR_HOLIDAY_PARAM;
```

Members

byEnable
是否启用

byDateMode
日期模式:0-模式 A,1-模式 B,2-模式 C

byRes1
保留

uHolidate
假日日期

byName
假日名称

byRes2
保留

See Also

[NET_DVR_HOLIDAY_PARAM_CFG](#)

7.37 NET_DVR_HOLIDATE_UNION:假日日期结构联合体

```
union{
    DWORD                dwSize[3];
    NET\_DVR\_HOLIDATE\_MODEA struModeA;
    NET\_DVR\_HOLIDATE\_MODEB struModeB;
    NET\_DVR\_HOLIDATE\_MODEC struModeC;
}NET_DVR_HOLIDATE_UNION, *LPNET_DVR_HOLIDATE_UNION;
```

Members

dwSize
联合体大小

struModeA
日期模式 A

struModeB
日期模式 B

struModeC
日期模式 C

See Also[NET_DVR_HOLIDAY_PARAM](#)**7.38 NET_DVR_HOLIDATE_MODEA:假日日期模式 A**

```

struct{
    BYTE    byStartMonth;
    BYTE    byStartDay;
    BYTE    byEndMonth;
    BYTE    byEndDay;
    BYTE    byRes[4];
}NET_DVR_HOLIDATE_MODEA, *LPNET_DVR_HOLIDATE_MODEA;

```

Members**byStartMonth**

开始月，从 1 开始

byStartDay

开始日，从 1 开始

byEndMonth

结束月份

byEndDay

结束日

byRes

保留

See Also[NET_DVR_HOLIDATE_UNION](#)**7.39 NET_DVR_HOLIDATE_MODEB:假日日期模式 B**

```

struct{
    BYTE    byStartMonth;
    BYTE    byStartWeekNum;
    BYTE    byStartWeekday;
    BYTE    byEndMonth;
    BYTE    byEndWeekNum;
    BYTE    byEndWeekday;
    BYTE    byRes[2];
}NET_DVR_HOLIDATE_MODEB, *LPNET_DVR_HOLIDATE_MODEB;

```

Members**byStartMonth**

开始月份，从 1 开始

byStartWeekNum

第几个星期,从 1 开始

byStartWeekday

星期几

byEndMonth

结束月份, 从 1 开始

byEndWeekNum

第几个星期, 从 1 开始

byEndWeekday

星期几

byRes

保留

See Also

[NET_DVR_HOLIDATE_UNION](#)

7.40 NET_DVR_HOLIDATE_MODEC: 假日日期模式 C

```
struct{
    WORD        wStartYear;
    BYTE        byStartMon;
    BYTE        byStartDay;
    WORD        wEndYear;
    BYTE        byEndMon;
    BYTE        byEndDay;
}NET_DVR_HOLIDATE_MODEC, *LPNET_DVR_HOLIDATE_MODEC;
```

Members

wStartYear

开始年份

byStartMon

开始月份, 从 1 开始

byStartDay

开始日

wEndYear

结束年份

byEndMon

结束月份

byEndDay

结束日

See Also

[NET_DVR_HOLIDATE_UNION](#)

7.41 NET_DVR_INPUT_INTERFACE_CTRL: 输入接口控制结构体

```
struct{
    BYTE    byInputSourceType;
    BYTE    byRes[15];
}
```

```
}NET_DVR_INPUT_INTERFACE_CTRL, *LPNET_DVR_INPUT_INTERFACE_CTRL;
```

Members

byInputSourceType

输入源类型，定义如下：

```
enum _INPUT_INTERFACE_TYPE_{
    INTERFACE_VGA    = 0,           //VGA 输入
    INTERFACE_SVIDEO,           //S-Video 输入
    INTERFACE_YPBPR,           //YPbPr 输入
    INTERFACE_DVI,           //DVI 输入
    INTERFACE_BNC,           //BNC 输入
    INTERFACE_DVI_LOOP,           //DVI（环通）输入
    INTERFACE_BNC_LOOP,           //BNC（环通）输入
    INTERFACE_HDMI,           //HDMI 输入
    INTERFACE_IP,           //IP 输入
    INTERFACE_USB,           //USB 输入
    INTERFACE_SDI,           //SDI 输入
    INTERFACE_DP,           //DP 输入
    INTERFACE_HDBASET           //HDBaseT 输入
}INPUT_INTERFACE_TYPE
```

byRes

保留

See Also

[NET_DVR_SCREEN_CONTROL_PARAM](#)

7.42 NET_DVR_IO_RELATION_INFO:矩阵输入输出关联关系信息结构体

```
struct{
    DWORD    dwSize;
    DWORD    dwIORelation[MATRIX_MAX_OUTPUT_NUM];
    BYTE     byRes[256];
}NET_DVR_IO_RELATION_INFO,*LPNET_DVR_IO_RELATION_INFO;
```

Members

dwSize

结构体大小

dwIORelation

输出通道关联的输入通道号，dwIORelation[i] = j 表示输出通道(i+1)和输入通道 j 关联，dwIORelation[i] = 0 表示输出通道 i+1 未关联输入通道。对于 LCD 拼接屏（屏幕服务器），表示屏幕和矩阵输入的关联关系，i+1 为屏幕索引序号（从 1 开始），j 为矩阵输入通道号

byRes

保留，置为 0

See Also

[NET_DVR_GetDVRConfig](#)

7.43 NET_DVR_IP_ADDRESS:设备 IP 地址信息

```
struct{
    BYTE    byDevAddress[MAX_DOMAIN_NAME];
    WORD    wDevPort;
    BYTE    byres[134];
}NET_DVR_IP_ADDRESS, *LPNET_DVR_IP_ADDRESS;
```

Members

byDevAddress
设备 IP 地址

wDevPort
设备端口号

byres
保留

See Also

[NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX](#)

7.44 NET_DVR_IPADDR:IP 地址

```
struct{
    char    slpV4[16];
    BYTE    slpV6[128];
}NET_DVR_IPADDR, *LPNET_DVR_IPADDR;
```

Members

slpV4
设备 IPv4 地址

slpV6
设备 IPv6 地址

See Also

[NET_DVR_PPPCFG_V30](#)、[NET_DVR_ETHERNET_V30](#)、[NET_DVR_NETCFG_V30](#)、[NET_DVR_VCS_USER_INFO](#)、[NET_MATRIX_DIGITALMATRIX](#)、[NET_DVR_LOG_V30](#)、[NET_DVR_MATRIX_PASSIVEMODE](#)、[NET_DVR_USER_INFO_V40](#)

7.45 NET_DVR_LCD_ALARM:LCD 屏幕报警信息结构体

```
struct{
    DWORD    dwSize;
    DWORD    dwScreenID;
    BYTE     byOnOffLine;
    BYTE     byTempState;
    BYTE     byFanState;
    BYTE     byFanException;
```

```

    BYTE    byTemperature;
    BYTE    byRes[27];
}NET_DVR_LCD_ALARM,*LPNET_DVR_LCD_ALARM;

```

Members

dwSize

结构体大小

dwScreenID

屏幕索引号(组合): 1 字节设备号+1 字节保留+2 字节屏幕索引序号

byOnOffLine

屏幕是否在线: 0- 离线, 1- 在线, 当屏幕离线时, 本结构后续参数无效

byTempState

板卡温度状态: 0- 正常, 1- 异常

byFanState

风扇状态: 0- 关, 1- 开

byFanException

风扇异常状态: 0- 不支持, 1- 正常, 2- 异常

byTemperature

屏幕板卡温度, 单位: 摄氏度

byRes

保留, 置为 0

Remarks

LCD 服务器在空闲时会每隔一定时间上传各屏幕工作状态信息。

See Also:[NET_DVR_SetDVRMessageCallBack V31](#)、[NET_DVR_StartListen V30](#)**7.46 NET_DVR_LCD_AUDIO_CFG: LCD 音频参数结构体**

```

struct{
    DWORD    dwSize;
    BYTE    byMute;
    BYTE    byVolume;
    BYTE    byBalance;
    BYTE    byRes[33];
}NET_DVR_LCD_AUDIO_CFG,*LPNET_DVR_LCD_AUDIO_CFG;

```

Members

dwSize

结构体大小

byMute

是否静音: 0- 否, 1- 是

byVolume

音量大小, 取值范围: 0~100

byBalance

声道平衡, 取值范围: -50~+50

byRes

保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)

7.47 NET_DVR_LOCAL_ABILITY_PARSE_CFG:能力集解析库配置

```
struct{
    BYTE    byEnableAbilityParse;
    BYTE    byRes[127];
}NET_DVR_LOCAL_ABILITY_PARSE_CFG,*LPNET_DVR_LOCAL_ABILITY_PARSE_CFG;
```

Members

byEnableAbilityParse

使用能力集解析库：0-不使用，1-使用，默认不使用

byRes

保留，置为 0

Remarks

模拟能力集默认禁用，调用该接口可以启用模拟能力集，支持获取设备各种能力。如果需要获取能力集（NET_DVR_GetDeviceAbility），可以调用此接口来启用模拟能力集，并且需要加载 LocalXml.zip（要求和 SDK 库文件放在同一个目录下）。

See Also

[NET_DVR_GetSDKLocalCfg](#)、[NET_DVR_SetSDKLocalCfg](#)

7.48 NET_DVR_LOCAL_BYTE_ENCODE_CONVERT:字符编码转换参数结构体

```
struct{
    CHAR_ENCODE_CONVERT    fnCharConvertCallBack;
    BYTE                    byRes[256];
}NET_DVR_LOCAL_BYTE_ENCODE_CONVERT,*LPNET_DVR_LOCAL_BYTE_ENCODE_CONVERT;
```

Members

fnCharConvertCallBack

字符编码转换回调函数

byRes

保留，置为 0

Callback Function

```
typedef int(CALLBACK *CHAR_ENCODE_CONVERT)(
    char        *pInput,
    DWORD       dwInputLen,
    DWORD       dwInEncodeType,
    char        *pOutput,
    DWORD       dwOutputLen,
    DWORD       dwOutEncodeType
```


);

Callback Function Parameters

pInput

[in] 输入字符串，内存由 SDK 申请，字符串数据也由 SDK 提供

dwInputLen

[in] 输入字符串缓冲区大小

dwInEncodeType

[in] 输入的字符编码格式：0- 无字符编码信息(老设备)，1- GB2312(简体中文)，2- GBK，3- BIG5(繁体中文)，4- Shift_JIS(日文)，5- EUC-KR(韩文)，6- UTF-8，7- ISO8859-1，8- ISO8859-2，9- ISO8859-3，...，依次类推，21- ISO8859-15(西欧)

pOutput

[out] 输出字符串，内存由 SDK 申请，存放使用用户字符编码接口转换之后的字符串

dwOutputLen

[out] 输出字符串缓冲区大小

dwOutEncodeType

[out] 输出字符编码格式：0- 无字符编码信息(老设备)，1- GB2312(简体中文)，2- GBK，3- BIG5(繁体中文)，4- Shift_JIS(日文)，5- EUC-KR(韩文)，6- UTF-8，7- ISO8859-1，8- ISO8859-2，9- ISO8859-3，...，依次类推，21- ISO8859-15(西欧)

Remarks

- 回调函数的返回值：-1 表示失败，0 表示成功（内存足够存放转换以后的字符串）。
- 设备的字符编码类型在登录接口返回，对应 [NET_DVR_DEVICEINFO_V40](#) 结构体中的参数 byCharEncodeType。SDK 内部需要字符编码转换时，SDK 默认使用 libiconv 库进行类型转换。如果不想使用 libiconv 编码库，可以调用 NET_DVR_SetSDKLocalCfg(类型: NET_SDK_LOCAL_CFG_TYPE_BYTE_ENCODE) 设置字符转码回调函数，告知 SDK 用户自己的字符编码接口，然后 SDK 将使用用户提供的字符编码接口进行字符串处理。

See Also[NET_DVR_SetSDKLocalCfg](#)**7.49 NET_DVR_LOCAL_CHECK_DEV:设备在线巡检参数结构体。**

```
struct{
    DWORD    dwCheckOnlineTimeout;
    DWORD    dwCheckOnlineNetFailMax;
    BYTE     byRes[256];
}NET_DVR_LOCAL_CHECK_DEV, *LPNET_DVR_LOCAL_CHECK_DEV;
```

Members

dwCheckOnlineTimeout

巡检时间间隔，单位：ms，取值范围：30s~120s，0 表示用默认值(120s)，推荐设置 30s

dwCheckOnlineNetFailMax

由于网络原因失败的最大累加次数，达到该次数，SDK 才回调用户异常消息，0 表示使用默认值 1，推荐设置 3 次

byRes

保留，置为 0

Remarks

- SDK 按照该结构体中的时间间隔对设备进行自动巡检，巡检过程中如果连失败或者重连成功在 [NET_DVR_SetExceptionCallBack_V30](#) 设置的异常消息回调函数中返回，对应异常消息类型为：EXCEPTION_EXCHANGE、RESUME_EXCHANGE。
- 推荐设置 30s 时间间隔、3 次，即心跳间隔为 1.5 分钟。

See Also

[NET_DVR_GetSDKLocalCfg](#)、[NET_DVR_SetSDKLocalCfg](#)

7.50 NET_DVR_LOCAL_MEM_POOL_CFG:内存池本地配置

```
struct{
    DWORD    dwAlarmMaxBlockNum;
    DWORD    dwAlarmReleaseInterval;
    BYTE     byRes[60];
}NET_DVR_LOCAL_MEM_POOL_CFG,*LPNET_DVR_LOCAL_MEM_POOL_CFG;
```

Members

dwAlarmMaxBlockNum

报警模块内存池最多向系统申请的内存块（block）个数，每个 block 为 64MB，超过这个上限则不向系统申请，0 表示无上限

dwAlarmReleaseInterval

报警模块空闲内存释放的间隔，单位：秒，为 0 表示不释放空闲的内存

byRes

保留，置为 0

See Also

[NET_DVR_GetSDKLocalCfg](#)、[NET_DVR_SetSDKLocalCfg](#)

7.51 NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG:按模块配置超时时间

```
struct{
    DWORD    dwPreviewTime;
    DWORD    dwAlarmTime;
    DWORD    dwVodTime;
    DWORD    dwElse;
    BYTE     byRes[512];
}NET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG,*LPNET_DVR_LOCAL_MODULE_RECV_TIMEOUT_CFG;
```

Members

dwPreviewTime

预览模块超时时间，单位：毫秒，取值范围：0~3000,000，0 表示恢复默认设置

dwAlarmTime

报警模块超时时间，单位：毫秒，取值范围：0~3000,000，0 表示恢复默认设置

dwVodTime

回放模块超时时间，单位：毫秒，取值范围：0~3000,000，0 表示恢复默认设置

dwElse

其他模块超时时间，单位：毫秒，取值范围：0~3000,000，0 表示恢复默认设置

byRes

保留，置为 0

See Also

[NET_DVR_GetSDKLocalCfg](#)、[NET_DVR_SetSDKLocalCfg](#)

7.52 NET_DVR_LOCAL_PROTECT_KEY_CFG:密钥配置

```
struct{
    BYTE    byProtectKey[128];
    BYTE    byRes[128];
}NET_DVR_LOCAL_PROTECT_KEY_CFG,*LPNET_DVR_LOCAL_PROTECT_KEY_CFG;
```

Members

byProtectKey

密钥，默认设置为 0

byRes

保留，置为 0

7.53 NET_DVR_LOCAL_TALK_MODE_CFG:对讲模式配置

```
struct{
    BYTE    byEnableAbilityParse;
    BYTE    byRes[127];
}NET_DVR_LOCAL_TALK_MODE_CFG,*LPNET_DVR_LOCAL_TALK_MODE_CFG;
```

Members

byTalkMode

对讲模式：0- 使用对讲库（默认），1- 使用 windows api 模式

byRes

保留，置为 0

Remarks

V4.2.2.5 及以前版本 SDK 均采用 windows API 实现相关功能。之后版本默认使用语音对讲库的方式，语音对讲库模式下必须加载 AudioIntercom.dll 和 OpenAL32.dll。

See Also

[NET_DVR_GetSDKLocalCfg](#)、[NET_DVR_SetSDKLocalCfg](#)

7.54 NET_DVR_LOCAL_TCP_PORT_BIND_CFG:本地 TCP 端口绑定配置

```
struct{
    WORD    wLocalBindTcpMinPort;
    WORD    wLocalBindTcpMaxPort;
    BYTE    byRes[60];
}NET_DVR_LOCAL_TCP_PORT_BIND_CFG,*LPNET_DVR_LOCAL_TCP_PORT_BIND_CFG;
```

Members**wLocalBindTcpMinPort**

本地绑定 TCP 最小端口

wLocalBindTcpMaxPort

本地绑定 TCP 最大端口

byRes

保留，置为 0

Remarks

端口绑定的策略是：给一个端口段，可以保证使用的端口都是在这个段里（多播除外），但不能保证每一个段内的端口都用到，因为是循环利用的；端口池中取出的端口会去尝试绑定，如果被占用了，将取下一个，如果段内每一个都绑定不了，则连接操作返回失败。建议最好不要设置系统预留的端口（1-1024），比如 80 等。

设置的最大端口应该大于等于最小端口，[0, 0]表示清除绑定，[0, 非 0]将设置失败，因为 0 不能进行绑定。

See Also

[NET_DVR_GetSDKLocalCfg](#)、[NET_DVR_SetSDKLocalCfg](#)

7.55 NET_DVR_LOCAL_UDP_PORT_BIND_CFG:本地 UDP 端口绑定配置

```
struct{
    WORD    wLocalBindUdpMinPort;
    WORD    wLocalBindUdpMaxPort;
    BYTE    byRes[60];
}NET_DVR_LOCAL_UDP_PORT_BIND_CFG,*LPNET_DVR_LOCAL_UDP_PORT_BIND_CFG;
```

Members**wLocalBindUdpMinPort**

本地绑定 UDP 最小端口

wLocalBindUdpMaxPort

本地绑定 UDP 最大端口

byRes

保留，置为 0

Remarks

端口绑定的策略是：给一个端口段，可以保证使用的端口都是在这个段里（多播除外），但不能保证每一个段内的端口都用到，因为是循环利用的；端口池中取出的端口会去尝试绑定，如果被占用了，将取下一个，如果段内每一个都绑定不了，则连接操作返回失败。建议最好不要设置系统预留的端口（1-1024），比如 80 等。

设置的最大端口应该大于等于最小端口，[0, 0]表示清除绑定，[0, 非 0]将设置失败，因为 0 不能进行绑定。

See Also

[NET_DVR_GetSDKLocalCfg](#)、[NET_DVR_SetSDKLocalCfg](#)

7.56 NET_DVR_LOG_V30:日志信息

```

struct{
    NET\_DVR\_TIME      strLogTime;
    DWORD             dwMajorType;
    DWORD             dwMinorType;
    BYTE              sPanelUser[MAX_NAMELEN];
    BYTE              sNetUser[MAX_NAMELEN];
    NET\_DVR\_IPADDR    struRemoteHostAddr;
    DWORD             dwParaType;
    DWORD             dwChannel;
    DWORD             dwDiskNumber;
    DWORD             dwAlarmInPort;
    DWORD             dwAlarmOutPort;
    DWORD             dwInfoLen;
    char              sInfo[LOG_INFO_LEN];
}NET_DVR_LOG_V30,*LPNET_DVR_LOG_V30;

```

Members

strLogTime

日志时间

dwMajorType

报警主类型，定义请参见[日志主类型列表](#)

dwMinorType

报警次类型，根据不同的主类型的次类型定义请参见[日志次类型列表](#)

sPanelUser

操作面板的用户名

sNetUser

网络操作的用户名

struRemoteHostAddr

远程主机地址

dwParaType

当日志的主类型为 MAJOR_OPERATION=03（操作），且次类型为 MINOR_LOCAL_CFG_PARM=0x52（本地配置参数）或 MINOR_REMOTE_GET_PARM=0x76（远程获得参数）或

MINOR_REMOTE_CFG_PARM=0x77（远程配置参数）时，该参数类型有效，其含义如下：

宏定义	宏定义值	含义
PARA_VIDEOOUT	0x1	视频输出结构配置
PARA_IMAGE	0x2	图像参数结构配置
PARA_ENCODE	0x4	压缩参数结构配置
PARA_NETWORK	0x8	网络参数结构配置
PARA_ALARM	0x10	报警参数结构配置
PARA_EXCEPTION	0x20	异常参数结构配置

PARA_DECODER	0x40	解码器参数结构配置
PARA_RS232	0x80	RS232 参数结构配置
PARA_PREVIEW	0x100	本地预览参数结构配置
PARA_SECURITY	0x200	用户权限参数结构配置
PARA_DATETIME	0x400	本地系统配置
PARA_FRAMETYPE	0x800	帧信息参数结构配置
PARA_CODESPLITTER	0x2000	码分器参数
PARA_RS485	0x2001	RS485 配置信息
PARA_DEVICE	0x2002	设备配置信息
PARA_HARDDISK	0x2003	硬盘配置信息
PARA_AUTOBOOT	0x2004	自动重启配置信息
PARA_HOLIDAY	0x2005	节假日配置信息
PARA_IPC	0x2006	IP 通道配置

dwChannel

通道号

dwDiskNumber

硬盘号

dwAlarmInPort

报警输入端口

dwAlarmOutPort

报警输出端口

dwInfoLen

日志附加信息长度

sInfo

日志附加信息

See Also

[NET_DVR_FindNextLog_V30](#)

7.57 NET_DVR_MATRIX_ABILITY: 解码显示和解码能力输出参数结构体

```
struct{
    DWORD    dwSize;
    BYTE     byDecNums;
    BYTE     byStartChan;
    BYTE     byVGANums;
    BYTE     byBNCNums;
    BYTE     byVGAWindowMode[8][12];
    BYTE     byBNCWindowMode[4];
    BYTE     byDspNums;
    BYTE     byHDMINums;
```

```

BYTE    byDVINums;
BYTE    byRes1[13];
BYTE    bySupportResolution[MAX_RESOLUTIONNUM];
BYTE    byHDMIWindowMode[4][8];
BYTE    byDVIWindowMode[4][8];
BYTE    byRes2[24];
}NET_DVR_MATRIX_ABILITY,*LPNET_DVR_MATRIX_ABILITY;

```

Members

dwSize

结构体大小

byDecNums

解码通道数

byStartChan

起始通道号

byVGANums

VGA 显示通道个数

byBNCNums

BNC 显示通道个数

byVGAWindowMode

VGA 支持的窗口模式，共有 1、2、4、9、16 等几种模式，例如 byVGAWindowMode[0][0]为 1 表示 VGA1 支持 1 画面窗口模式，byVGAWindowMode[0][1]为 2 表示 VGA1 支持 2 画面模式，到 0 为止

byBNCWindowMode

BNC 支持的窗口模式，例如 byBNCWindowMode[0]为 1 表示 BNC 输出支持 1 画面窗口模式，byBNCWindowMode[1]为 2 表示 BNC 输出支持 2 画面模式，到 0 为止

byDspNums

DSP 数目

byHDMINums

HDMI 显示通道个数（从 25 开始）

byDVINums

DVI 显示通道个数（从 29 开始）

byRes1

保留，置为 0

bySupportResolution

按下面的枚举类型定义，一个字节表示一种分辨率是否支持，0-不支持，1-支持，分辨率定义如下：

```

enum {
    VGA_NOT_AVALIABLE,
    VGA_THS8200_MODE_SVGA_60HZ,           //无
    VGA_THS8200_MODE_SVGA_75HZ,           // SVGA:800*600
    VGA_THS8200_MODE_XGA_60HZ,            // SVGA:800*600
    VGA_THS8200_MODE_XGA_70HZ,            // XGA:1024*768
    VGA_THS8200_MODE_SXGA_60HZ,           // SXGA:1280*1024
    VGA_THS8200_MODE_720P_60HZ,           // H720P:1280*720
    VGA_THS8200_MODE_1080I_60HZ,          //1080i:1920*1080
    VGA_THS8200_MODE_1080P_30HZ,          //1080P:1920*1080
}

```

```

VGA_THS8200_MODE_UXGA_30HZ, // UXGA:1600*1200
HDMI_SII9134_MODE_XGA_60HZ, // XGA:1024*768
HDMI_SII9134_MODE_SXGA_60HZ, // SXGA:1280*1024
HDMI_SII9134_MODE_SXGA2_60HZ, // SXGA2:1280*960
HDMI_SII9134_MODE_720P_60HZ, //720P:1280*720
HDMI_SII9134_MODE_720P_50HZ, //720P:1280*720
HDMI_SII9134_MODE_1080I_60HZ, //1080P:1920*1080
HDMI_SII9134_MODE_1080I_50HZ, //1080I:1920*1080
HDMI_SII9134_MODE_1080P_25HZ, //1080P:1920*1080
HDMI_SII9134_MODE_1080P_30HZ, //1080P:1920*1080
HDMI_SII9134_MODE_1080P_50HZ, //1080P:1920*1080
HDMI_SII9134_MODE_1080P_60HZ, //1080P:1920*1080
HDMI_SII9134_MODE_UXGA_60HZ, // UXGA:1600*1200
DVI_SII9134_MODE_XGA_60HZ, // XGA:1024*768
DVI_SII9134_MODE_SXGA_60HZ, // SXGA:1280*1024
DVI_SII9134_MODE_SXGA2_60HZ, // SXGA2:1280*960
DVI_SII9134_MODE_720P_60HZ, //720P:1280*720
DVI_SII9134_MODE_720P_50HZ, //720P:1280*720
DVI_SII9134_MODE_1080I_60HZ, //1080P:1920*1080
DVI_SII9134_MODE_1080I_50HZ, //1080I:1920*1080
DVI_SII9134_MODE_1080P_25HZ, //1080P:1920*1080
DVI_SII9134_MODE_1080P_30HZ, //1080P:1920*1080
DVI_SII9134_MODE_1080P_50HZ, //1080P:1920*1080
DVI_SII9134_MODE_1080P_60HZ, //1080P:1920*1080
DVI_SII9134_MODE_UXGA_60HZ // UXGA:1600*1200
} VGA_MODE

```

byHDMIWindowMode

HDMI 支持的窗口模式

byDVIWindowMode

DVI 支持的窗口模式

byRes2

保留，置为 0

See Also:

[NET_DVR_GetDeviceAbility](#)

7.58 NET_DVR_MATRIX_CHAN_INFO_V41:动态解码轮巡通道信息

```

struct{
    BYTE                byEnable;
    BYTE                byStreamMode;
    BYTE                byRes[2];
    NET\_DVR\_DEC\_STREAM\_MODE uDecStreamMode;
}

```



```
}NET_DVR_MATRIX_CHAN_INFO_V41,*LPNET_DVR_MATRIX_CHAN_INFO_V41;
```

Members

byEnable

是否启用：0- 否，1- 启用

byStreamMode

取流模式：0- 无效，1- 通过 IP 或域名取流，2- 通过 URL 取流，3- 通过动态域名解析向设备取流

byRes

保留，置为 0

uDecStreamMode

取流配置信息

See Also

[NET_DVR_MATRIX_LOOP_DECINFO_V41](#)

7.59 NET_DVR_MATRIX_CHAN_STATUS:解码通道状态结构体

```
struct{
    BYTE    byDecodeStatus;
    BYTE    byStreamType;
    BYTE    byPacketType;
    BYTE    byRecvBufUsage;
    BYTE    byDecBufUsage;
    BYTE    byFpsDecV;
    BYTE    byFpsDecA;
    BYTE    byCpuLoad;
    BYTE    byRes1[4];
    DWORD   dwDecodedV;
    DWORD   dwDecodedA;
    WORD    wImgW;
    WORD    wImgH;
    BYTE    byVideoFormat;
    BYTE    byRes2[3];
    DWORD   dwDecChan;
    BYTE    byRes3[20];
}NET_DVR_MATRIX_CHAN_STATUS,*LPNET_DVR_MATRIX_CHAN_STATUS;
```

Members

byDecodeStatus

当前状态：0-未启动，1-启动解码

byStreamType

码流类型，具体定义见下表：

宏定义	宏定义值	含义
NET_DVR_ENCODER_UNKOWN	0	未知编码格式
NET_DVR_ENCODER_H264	1	private 264
NET_DVR_ENCODER_S264	2	标准 H264

NET_DVR_ENCODER_MPEG4	3	MPEG4
NET_DVR_ORIGINALSTREAM	4	原始流
NET_DVR_PICTURE	5	智能分析联动的报警图片
NET_DVR_ENCODER_MJPEG	6	MJPEG
NET_DVR_ECONDER_MPEG2	7	MPEG2

byPacketType

封装格式，具体定义见下表：

宏定义	宏定义值	含义
NET_DVR_STREAM_TYPE_UNKOWN	0	未知封装格式
NET_DVR_STREAM_TYPE_PRIVT	1	私有自定义封装格式
NET_DVR_STREAM_TYPE_TS	7	TS 封装格式
NET_DVR_STREAM_TYPE_PS	8	PS 封装格式
NET_DVR_STREAM_TYPE_RTP	9	RTP 封装格式
NET_DVR_STREAM_TYPE_ORIGIN	10	未封装

byRecvBufUsage

接收缓冲使用率

byDecBufUsage

解码缓冲使用率

byFpsDecV

视频解码帧率，大于 1 的数值即代表帧率值，对于小于 1 的低帧率部分的定义见下表：

宏定义	宏定义值	含义
LOW_DEC_FPS_1_2	51	1/2 帧
LOW_DEC_FPS_1_4	52	1/4 帧
LOW_DEC_FPS_1_8	53	1/8 帧
LOW_DEC_FPS_1_16	54	1/16 帧

byFpsDecA

音频解码帧率，其值的含义和视频解码帧率相同

byCpuLoad

DSP CPU 利用率

byRes1

保留，置为 0

dwDecodedV

解码的视频帧

dwDecodedA

解码的音频帧

wlmgW

解码器当前的图像的宽

wlmgH

解码器当前的图像的高

byVideoFormat

视频制式，具体定义如下：

```
enum{
    VS_NON   = 0,           //无
    VS_NTSC  = 1,           //NTSC 制式
    VS_PAL   = 2           //PAL 制式
}VIDEO_STANDARD
```

byRes2

保留，置为 0

dwDecChan

解码通道号，获取全部解码通道状态时有效，设置时可填 0

byRes3

保留，置为 0

See Also

[NET_DVR_DECODER_WORK_STATUS_V41](#)

7.60 NET_DVR_MATRIX_DEC_CHAN_INFO_V41:当前窗口解码信息

```
struct{
    DWORD           dwSize;
    BYTE            byStreamMode;
    BYTE            byRes1[3];
    NET\_DVR\_DEC\_STREAM\_MODE uDecStreamMode;
    DWORD           dwPlayMode;
    NET\_DVR\_TIME      StartTime;
    NET\_DVR\_TIME      StopTime;
    char            sFileName[128];
    DWORD           dwGetStreamMode;
    NET\_DVR\_MATRIX\_PASSIVEMODE struPassiveMode;
    BYTE            byRes2[32];
}NET_DVR_MATRIX_DEC_CHAN_INFO_V41,*LPNET_DVR_MATRIX_DEC_CHAN_INFO_V41;
```

Members

dwSize

结构体大小

byStreamMode

取流模式：0- 无效，1- 通过 IP 或域名取流，2- 通过 URL 取流，3- 通过动态域名解析向设备取流

byRes1

保留

uDecStreamMode

取流信息

dwPlayMode

解码状态：0-动态解码，1-循环解码，2-按时间回放，3-按文件回放

StartTime

按时间回放开始时间

StopTime

按时间回放结束时间

sFileName

按文件回放文件名

dwGetStreamMode

取流模式：0-未解码，1-主动，2-被动

struPassiveMode

被动解码信息

byRes2

保留

See Also

[NET_DVR_MatrixGetDecChanInfo_V41](#)

7.61 NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX:远程文件回放解码参数

```

struct{
    DWORD    dwSize;
    DWORD    dwDecChannel;
    BYTE     byAddressType;
    BYTE     byChannelType;
    BYTE     byres[2];
    BYTE     sUserName[NAME_LEN];
    BYTE     sPassword[PASSWD_LEN];
    DWORD    dwChannel;
    BYTE     byStreamId[STREAM_ID_LEN];
    DWORD    dwPlayMode;
    union{
        BYTE                                     byRes[200];
        NET\_DVR\_IP\_ADDRESS                       struIpAddr;
        NET\_DVR\_DDNS\_ADDRESS                       struDdnsAddr;
    }unionAddr;
    union{
        BYTE                                     byRes[128];
        NET\_DVR\_PLAY\_BACK\_BY\_TIME                 struPlayBackByTime;
        char                                     sFileName[128];
    }unionPlayBackInfo;
}NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX, *LPNET_DVR_MATRIX_DEC_REMOTE_PLAY_EX;

```

Members

dwSize

结构体大小

dwDecChannel

子窗口号(组合)

byAddressType

设备地址类型：0- IP， 1- DDNS 域名

byChannelType

通道类型：0- 普通通道，1- 零通道，2- 流 ID

byres

保留

sUserName

用户名

sPassword

密码

dwChannel

设备通道号

byStreamId

流 ID,此参数在通道类型为流 ID 时有效

dwPlayMode

回放模式：0- 按文件，1-按时间

unionAddr 为设备地址联合体

byRes

联合体大小为 200 字节

strulpAddr

设备 IP 地址

struDdnsAddr

设备 DDNS 域名

unionPlayBackInfo 为回放参数联合体

byRes

联合体大小为 128 字节

struPlayBackByTime

按时间回放时文件信息

sFileName

按文件名回放时文件信息

See Also

[NET_DVR_RemoteControl](#)

7.62 NET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS:回放状态信息

```
struct{
    DWORD    dwSize;
    DWORD    dwCurMediaFileLen;
    DWORD    dwCurMediaFilePosition;
    DWORD    dwCurMediaFileDuration;
    DWORD    dwCurPlayTime;
    DWORD    dwCurMediaFileFrames;
    DWORD    dwCurDataType;
    BYTE     res[72];
}NET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS,*LPNET_DVR_MATRIX_DEC_REMOTE_PLAY_STATUS;
```

Members

dwSize

结构体大小

dwCurMediaFileLen

当前播放的媒体文件长度

dwCurMediaFilePosition

当前播放文件的播放位置

dwCurMediaFileDuration

当前播放文件的总时间

dwCurPlayTime

当前已经播放的时间

dwCurMediaFileFrames

当前播放文件的总帧数

dwCurDataType

当前传输的数据类型：19-文件头，20-流数据，21-播放结束标志

res

保留，置为 0

See Also

[NET_DVR_MatrixGetRemotePlayStatus](#)

7.63 NET_DVR_MATRIX_LOOP_DECINFO_V41:动态解码轮巡通道参数

```
struct{
    DWORD                dwSize;
    DWORD                dwPoolTime;
    NET\_DVR\_MATRIX\_CHAN\_INFO\_V41 struchanConInfo[MAX_CYCLE_CHAN_V30];
    BYTE                 byRes[16];
}NET_DVR_MATRIX_LOOP_DECINFO_V41,*LPNET_DVR_MATRIX_LOOP_DECINFO_V41;
```

Members

dwSize

结构体大小

dwPoolTime

轮巡时间，单位：s

struchanConInfo

动态解码通道信息

byRes

保留，置为 0

See Also

[NET_DVR_MatrixGetLoopDecChanInfo_V41](#)、[NET_DVR_MatrixSetLoopDecChanInfo_V41](#)

7.64 NET_DVR_MATRIX_PASSIVEMODE:被动解码参数

```
struct{
    WORD                wTransProtol;
    WORD                wPassivePort;
```

```

NET\_DVR\_IPADDR    struMcastIP;
BYTE                byStreamType;
BYTE                byReconnectFlag;
BYTE                byRequestType;
BYTE                byRes[5];

```

```
}NET_DVR_MATRIX_PASSIVEMODE,*LPNET_DVR_MATRIX_PASSIVEMODE;
```

Members

wTransProtol

传输协议：0-TCP，1-UDP，2-MCAST

wPassivePort

UDP 端口，TCP 时端口默认是 8000，不同窗口的 UDP 端口号需设置为不同的值

struMcastIP

TCP/UDP 时无效，MCAST 时为多播地址(暂时不支持多播，保留)

byStreamType

数据播放模式：1- 实时流，2- 文件流

byReconnectFlag

重连标志：0- 新连接，1- 重连

byRequestType

实时流模式：1- 客户端直接发流，不发送交互头；0- 需要发送交互头

byRes

保留，置为 0

See Also

[NET_DVR_MATRIX_DEC_CHAN_INFO_V41](#)

7.65 NET_DVR_MATRIX_SWITCH_INFO:矩阵切换信息结构体

```

struct{
    DWORD    dwSize;
    DWORD    dwInputChan;
    DWORD    dwOutputChan;
    DWORD    dwMatrixID;
    BYTE     byRes[28];
}NET_DVR_MATRIX_SWITCH_INFO,*LPNET_DVR_MATRIX_SWITCH_INFO;

```

Members

dwSize

结构体大小

dwInputChan

输入通道号，为 0 表示取消和指定输出的关联关系

dwOutputChan

要切换到的输出通道号，0xffffffff 表示切换到所有输出。对于 LCD 拼接屏（屏幕服务器），该参数表示屏幕索引号(组合)：1 字节设备号+1 字节保留+2 字节屏幕索引序号

dwMatrixID

矩阵 ID 号

byRes

保留，置为 0

See Also

[NET_DVR_RemoteControl](#)

7.66 NET_DVR_MATRIX_VOUTCFG:显示通道配置结构体

```
struct{
    DWORD                dwSize;
    BYTE                 byAudio;
    BYTE                 byAudioWindowIdx;
    BYTE                 byRes;
    BYTE                 byVedioFormat;
    DWORD                dwResolution;
    DWORD                dwWindowMode;
    BYTE                 byJoinDecChan[MAX_WINDOWS_V41];
    BYTE                 byEnlargeStatus;
    BYTE                 byEnlargeSubWindowIndex;
    BYTE                 byScale;
    BYTE                 byUnionType;
    NET\_DVR\_VIDEO\_PLATFORM struDiff;
    DWORD                dwDispChanNum;
    BYTE                 byRes2[76];
}NET_DVR_MATRIX_VOUTCFG, *LPNET_DVR_MATRIX_VOUTCFG;
```

Members

dwSize

结构体大小

byAudio

音频是否开启：0- 关闭，1- 开启

byAudioWindowIdx

音频开启子窗口

byRes

保留

byVedioFormat

视频制式：1- NTSC，2- PAL，0- NULL

dwResolution

分辨率

dwWindowMode

画面模式，能力集获取

byJoinDecChan

各个子窗口关联的解码通道

byEnlargeStatus

是否处于放大状态：0- 不放大，1- 放大。获取时有效。

byEnlargeSubWindowIndex

放大的子窗口号（获取时有效）

byScale

显示模式：0- 真实显示，1- 缩放显示(针对 BNC)

byUnionType

区分共用体：0- 视频综合平台内部解码器显示通道配置，1- 其他解码器显示通道配置

struDiff

显示通道配置联合体

dwDispChanNum

显示输出号，此参数在全部获取时有效

byRes2

保留

Remarks

在 V41 以下的版本中，byDecResolution[MAX_WINDOWS_V41]是 byDecResolution，V41 版本中改为数组。在低版本向高版本转换时，根据 NET_DVR_MATRIX_VOUTCFG 中的画面模式 dwWindowMode 对 byDecResolution[MAX_WINDOWS_V41]进行赋值；高版本向低版本转化时用 byDecResolution[0]。

See Also

[NET_DVR_MatrixGetDisplayCfg_V41](#)、[NET_DVR_MatrixSetDisplayCfg_V41](#)、[NET_DVR_GetDeviceAbility](#)

7.67 NET_DVR_MOTION:移动侦测参数结构体

```
struct{
    BYTE                byMotionScope[18][22];
    BYTE                byMotionSensitive;
    BYTE                byEnableHandleMotion;
    BYTE                reservedData[2];
    NET\_DVR\_HANDLEEXCEPTION strMotionHandleType;
    NET\_DVR\_SCHEDTIME struAlarmTime[MAX_DAYS][MAX_TIMESEGMENT];
    BYTE                byRelRecordChan[MAX_CHANNUM];
}NET_DVR_MOTION, *LPNET_DVR_MOTION;
```

Members

byMotionScope

移动侦测区域，共有 22*18 个小宏块，取值为 1 表示该宏块设定为移动侦测区域，0 表示不设定为移动侦测区域

byMotionSensitive

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

byEnableHandleMotion

是否处理移动侦测：0-不处理，1-处理

reservedData

保留，置为 0

strMotionHandleType

处理方式

struAlarmTime

布防时间

byRelRecordChan

报警触发的录象通道，为 1 表示触发该通道

Remarks

整个图像的区域大小为 704*576

See Also

[NET_DVR_PICCFG](#)

7.68 NET_DVR_MSC_SCREEN_BACKLIGHT_CFG: 屏幕背光参数结构体

```
struct{
    DWORD    dwSize;
    BYTE     byBacklight;
    BYTE     byRes[15];
}NET_DVR_MSC_SCREEN_BACKLIGHT_CFG, *LPNET_DVR_MSC_SCREEN_BACKLIGHT_CFG;
```

Members

dwSize

结构体大小

byBacklight

背光参数，取值范围：0~100

byRes

保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)

7.69 NET_DVR_MSC_SCREEN_INTERFACE_CFG: 输入源参数结构体

```
struct{
    DWORD    dwSize;
    BYTE     byInterfaceType;
    BYTE     byNoSignalPic;
    BYTE     byRes[30];
}NET_DVR_MSC_SCREEN_INTERFACE_CFG, *LPNET_DVR_MSC_SCREEN_INTERFACE_CFG;
```

Members

dwSize

结构体大小

byInterfaceType

输入源索引，具体定义如下：

```
enum _INPUT_INTERFACE_TYPE_{
    INTERFACE_VGA    = 0,           // VGA 输入
    INTERFACE_SVIDEO,             // S-Video 输入
    INTERFACE_YPBPR,             // YPbPr 输入
    INTERFACE_DVI,               // DVI 输入
    INTERFACE_BNC,               // BNC 输入
    INTERFACE_DVI_LOOP,          // DVI（环通）输入
    INTERFACE_BNC_LOOP,          // BNC（环通）输入
};
```

```

INTERFACE_HDMI,           // HDMI 输入
INTERFACE_IP,             // IP 输入
INTERFACE_USB,           // USB 输入
INTERFACE_SDI,           // SDI 输入
INTERFACE_DP,            // DP 输入
INTERFACE_HDBASET        // HDBaseT 输入
}INPUT_INTERFACE_TYPE

```

byNoSignalPic

无信号画面模式：1- 蓝屏，2- 黑屏，3- LOGO 画面

byRes

保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)

7.70 NET_DVR_MSC_SCREEN_PIP_CFG: 屏幕画中画参数结构体

```

struct{
    DWORD           dwSize;
    BYTE            byEnable;
    BYTE            bySubWindowSource;
    BYTE            bySubWindowBorderColor;
    BYTE            byRes1;
    NET\_DVR\_RECTCFG\_EX  struPosition;
    BYTE            byRes2[16];
}NET_DVR_MSC_SCREEN_PIP_CFG, *LPNET_DVR_MSC_SCREEN_PIP_CFG;

```

Members

dwSize

结构体大小

byEnable

是否启用画中画：0- 不启用，1- 启用

bySubWindowSource

子画面信号源，具体定义如下：

```

enum _INPUT_INTERFACE_TYPE_{
    INTERFACE_VGA = 0,           // VGA 输入
    INTERFACE_SVIDEO,          // S-Video 输入
    INTERFACE_YPBPR,           // YPbPr 输入
    INTERFACE_DVI,             // DVI 输入
    INTERFACE_BNC,             // BNC 输入
    INTERFACE_DVI_LOOP,        // DVI（环通）输入
    INTERFACE_BNC_LOOP,        // BNC（环通）输入
    INTERFACE_HDMI,           // HDMI 输入
    INTERFACE_IP,             // IP 输入
    INTERFACE_USB,            // USB 输入
    INTERFACE_SDI,            // SDI 输入
}

```

```

INTERFACE_DP, // DP 输入
INTERFACE_HDBASET // HDBaset 输入
}INPUT_INTERFACE_TYPE

```

bySubWindowBorderColor

子画面边框颜色：1- 蓝色，2- 黑色

byRes1

保留，置为 0

struPosition

子画面位置，屏幕总大小为 1920*1920

byRes2

保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)

7.71 NET_DVR_MSC_SCREEN_PARAM: 屏幕参数联合体

union{

NET_DVR_MSC_SCREEN_INTERFACE_CFG	struInterfaceCfg;
NET_DVR_SCREEN_FAN_WORK_MODE_CFG	struFanWorkMode;
NET_DVR_SCREEN_VGA_CFG	struVgaCfg;
NET_DVR_SCREEN_MENU_CFG	struMenuCfg;
NET_DVR_VIDEO_OUT_CFG	struOutEffectCfg;
NET_DVR_COLOR_TEMPERATURE_CFG	struColorTemperatureCfg;
NET_DVR_ADC_CFG	struAdcCfg;
NET_DVR_SCREEN_EDGE_CFG	struScreenEdgeCfg;
NET_DVR_MSC_SCREEN_BACKLIGHT_CFG	struBacklight;
NET_DVR_MSC_SCREEN_PIP_CFG	struPicInPicCfg;
NET_DVR_DEFOG_LCD	struDefog;
NET_DVR_SCREEN_WORK_STATE	struWorkState;
NET_DVR_BOOT_LOGO_CFG	struBootLogoCfg;
NET_DVR_LCD_AUDIO_CFG	struAudioCfg;
BYTE	byRes[256];

}NET_DVR_MSC_SCREEN_PARAM, *LPNET_DVR_MSC_SCREEN_PARAM;

Members

struInterfaceCfg

输入源参数

struFanWorkMode

风扇工作模式参数

struVgaCfg

VGA 信号参数

struMenuCfg

屏幕菜单参数

struOutEffectCfg

显示效果参数

struColorTemperatureCfg
 色温参数
struAdcCfg
 ADC 参数
struScreenEdgeCfg
 屏幕边缘参数
struBacklight
 背光参数
struPicInPicCfg
 画中画参数
struDefog
 去雾参数
struWorkStatw
 屏幕工作状态
struBootLogoCfg
 开始显示 logo 参数
struAudioCfg
 LCD 音频参数
byRes
 联合体大小，256 字节

See Also

[NET_DVR_MSC_SCREEN_PARAM_CFG](#)

7.72 NET_DVR_MSC_SCREEN_PARAM_CFG: 屏幕参数结构体

```

struct{
    DWORD                dwSize;
    BYTE                 byParamType;
    BYTE                 byRes1[3];
    NET\_DVR\_MSC\_SCREEN\_PARAM struParam;
    BYTE                 byRes2[32];
}NET_DVR_MSC_SCREEN_PARAM_CFG, *LPNET_DVR_MSC_SCREEN_PARAM_CFG;
  
```

Members

dwSize

结构体大小

byParamType

参数类型：1-输入源参数，2-风扇工作模式参数，3-VGA 信号参数，4-屏幕菜单参数，5-显示效果参数，6-色温参数，7-ADC 参数，8-屏幕边缘参数，9-背光参数，10-画中画参数，11-透雾参数，12-屏幕工作状态（只能获取），13-开机 LOGO 是否显示参数，14-音频参数

byRes1

保留，置为 0

struParam

屏幕参数

byRes2

保留，置为 0

See Also

[NET_DVR_StartRemoteConfig](#)、[NET_DVR_MSC_SCREEN_REMOTE_CFG](#)

7.73 NET_DVR_MSC_SCREEN_PARAM_COND: 屏幕相关参数配置条件结构体

```
struct{
    DWORD                dwSize;
    BYTE                 byWallNo;
    BYTE                 byParamType;
    BYTE                 byRes1[2];
    NET\_DVR\_RECTCFG\_EX  struRectCfg;
    BYTE                 byRes2[32];
}NET_DVR_MSC_SCREEN_PARAM_COND, *LPNET_DVR_MSC_SCREEN_PARAM_COND;
```

Members

dwSize

结构体大小

byWallNo

电视墙号

byParamType

参数类型：1-输入源参数，2-风扇工作模式参数，3-VGA 信号参数，4-屏幕菜单参数，5-显示效果参数，6-色温参数，7-ADC 参数，8-屏幕边缘参数，9-背光参数，10-画中画参数，11-透雾参数，12-屏幕工作状态（只能获取），13-开机 LOGO 是否显示参数，14-音频参数

byRes1

保留，置为 0

struRectCfg

屏幕区域

byRes2

保留，置为 0

See Also

[NET_DVR_StartRemoteConfig](#)

7.74 NET_DVR_MSC_SCREEN_REMOTE_CFG: 屏幕相关参数配置结构体

```
struct{
    DWORD                dwSize;
    BYTE                 byWallNo;
    BYTE                 byRes1[3];
    NET\_DVR\_RECTCFG\_EX  struRectCfg;
    NET\_DVR\_MSC\_SCREEN\_PARAM\_CFG struScreenParam;
    BYTE                 byRes2[32];
}
```

```
}NET_DVR_MSC_SCREEN_REMOTE_CFG, *LPNET_DVR_MSC_SCREEN_REMOTE_CFG;
```

Members

dwSize

结构体大小

byWallNo

电视墙号

byRes1

保留，置为 0

struRectCfg

屏幕区域

struScreenParam

配置的屏幕参数

byRes2

保留，置为 0

See Also

[NET_DVR_StartRemoteConfig](#)

7.75 NET_DVR_MSC_SPLICE_CFG:屏幕拼接配置结构体

```
struct{
    DWORD                dwSize;
    BYTE                 bySpliceNo;
    BYTE                 byWallNo;
    BYTE                 byEnable;
    BYTE                 byRes1;
    NET\_DVR\_RECTCFG\_EX  struRectCfg;
    BYTE                 byRes2[32];
}NET_DVR_MSC_SPLICE_CFG, *LPNET_DVR_MSC_SPLICE_CFG;
```

Members

dwSize

结构体大小

bySpliceNo

拼接屏号

byWallNo

电视墙号

byEnable

是否使能：0- 否，1- 是

byRes1

保留，置为 0

struRectCfg

需要进行拼接的区域，获取的时候此参数填 0

byRes2

保留，置为 0

See Also

[NET_DVR_StartRemoteConfig](#)、[NET_DVR_SendRemoteConfig](#)

7.76 NET_DVR_NETCFG_V30:网络配置

```

struct{
    DWORD                dwSize;
    NET\_DVR\_ETHERNET\_V30 struEtherNet[MAX_ETHERNET];
    NET\_DVR\_IPADDR       struRes1[2];
    NET\_DVR\_IPADDR       struAlarmHostIpAddr;
    WORD                wRes2[2];
    WORD                wAlarmHostIpPort;
    BYTE                byUseDhcp;
    BYTE                byRes3;
    NET\_DVR\_IPADDR       struDnsServer1IpAddr;
    NET\_DVR\_IPADDR       struDnsServer2IpAddr;
    BYTE                byIpResolver[MAX_DOMAIN_NAME];
    WORD                wIpResolverPort;
    WORD                wHttpPortNo;
    NET\_DVR\_IPADDR       struMulticastIpAddr;
    NET\_DVR\_IPADDR       struGatewayIpAddr;
    NET\_DVR\_PPPOECFG     struPPPoE;
    BYTE                byRes[64];
}NET_DVR_NETCFG_V30,*LPNET_DVR_NETCFG_V30;

```

Members

dwSize

结构体大小

struEtherNet

以太网口

struRes1

保留，置为 0

struAlarmHostIpAddr

报警主机 IP 地址

wRes2

保留，置为 0

wAlarmHostIpPort

报警主机端口号

byUseDhcp

是否启用 DHCP: 0xff-无效; 0-不启用; 1-启用

byRes3

保留，置为 0

struDnsServer1IpAddr

域名服务器 1 的 IP 地址

struDnsServer2IpAddr

域名服务器 2 的 IP 地址

byIpResolver

IP 解析服务器域名或 IP 地址（8000 设备不支持域名）

wIpResolverPort

IP 解析服务器端口号

wHttpPortNo

HTTP 端口号

struMulticastIpAddr

多播组地址

struGatewayIpAddr

网关地址

struPPPoE

PPPoE 参数

byRes

保留，置为 0

Remarks

8000 等 3.0 协议以下的设备，参数 byUseDhcp 为 0xff-无效，将其 IP 地址填成空，设备会自动去获取 DHCP。

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.77 NET_DVR_NTTPARA:网络应用参数(NTP)

```
struct{
    BYTE          sNTPServer[64];
    WORD          wInterval;
    BYTE          byEnableNTP;
    signed char   cTimeDifferenceH;
    signed char   cTimeDifferenceM;
    BYTE          res1;
    WORD          wNtpPort;
    BYTE          res2[8];
}NET_DVR_NTTPARA,*LPNET_DVR_NTTPARA;
```

Members

sNTPServer

NTP 服务器域名或者 IP 地址

wInterval

校时间隔时间，以分钟或小时为单位（通过能力集 [NET_DVR_GetDeviceAbility](#) 获取，对应网络应用参数能力集：DEVICE_NETAPP_ABILITY）

byEnableNTP

NTP 校时是否启用：0—否，1—是

cTimeDifferenceH

与国际标准时间的时差（小时），-12 ... +13

cTimeDifferenceM

与国际标准时间的时差（分钟），0, 30, 45

res1

保留，置为 0

wNtpPort

NTP 服务器端口，设备默认为 123

res2

保留，置为 0

See Also

[NET_DVR_ZONEANDDST](#)

7.78 NET_DVR_PICCFG:通道图像结构体

```
struct{
    DWORD                dwSize;
    BYTE                 sChanName[NAME_LEN];
    DWORD                dwVideoFormat;
    BYTE                 byRes[4];
    DWORD                dwShowChanName;
    WORD                 wShowNameTopLeftX;
    WORD                 wShowNameTopLeftY;
    NET\_DVR\_VILOST      struVILost;
    NET\_DVR\_MOTION     struMotion;
    NET\_DVR\_HIDEALARM  struHideAlarm;
    DWORD                dwEnableHide;
    WORD                 wHideAreaTopLeftX;
    WORD                 wHideAreaTopLeftY;
    WORD                 wHideAreaWidth;
    WORD                 wHideAreaHeight;
    DWORD                dwShowOsd;
    WORD                 wOSDTopLeftX;
    WORD                 wOSDTopLeftY;
    BYTE                 byOSDType;
    BYTE                 byDispWeek;
    BYTE                 byOSDAttrib;
    char                 reservedData2;
}NET_DVR_PICCFG,*LPNET_DVR_PICCFG;
```

Members

dwSize

结构体大小

sChanName

通道名称

dwVideoFormat

视频制式：1-NTSC；2-PAL

byRes

保留

dwShowChanName

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

wShowNameTopLeftX

通道名称显示位置的 x 坐标

wShowNameTopLeftY

通道名称显示位置的 y 坐标

struVILost

信号丢失报警参数

struMotion

移动侦测报警参数

struHideAlarm

遮挡报警参数

dwEnableHide

是否启动隐私遮蔽：0-否，1-是

wHideAreaTopLeftX

隐私遮蔽区域的 x 坐标

wHideAreaTopLeftY

隐私遮蔽区域的 y 坐标

wHideAreaWidth

隐私遮蔽区域的宽

wHideAreaHeight

隐私遮蔽区域的高

dwShowOsd

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

wOSDTopLeftX

OSD 的 x 坐标

wOSDTopLeftY

OSD 的 y 坐标

byOSDType

OSD 类型(年月日格式):

0—XXXX-XX-XX 年月日

1—XX-XX-XXXX 月日年

2—XXXX 年 XX 月 XX 日

3—XX 月 XX 日 XXXX 年

4—XX-XX-XXXX 日月年

5—XX 日 XX 月 XXXX 年

byDispWeek

是否显示星期：0-不显示，1-显示

byOSDAttrib

OSD 属性（透明/闪烁）:

1—透明，闪烁

2—透明，不闪烁

3—闪烁，不透明

4—不透明，不闪烁

reservedData2

保留，置为 0

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.79 NET_DVR_PLAY_BACK_BY_TIME:回放时间信息

```
struct{
    NET\_DVR\_TIME    StartTime;
    NET\_DVR\_TIME    StopTime;
}NET_DVR_PLAY_BACK_BY_TIME, *LPNET_DVR_PLAY_BACK_BY_TIME;
```

Members

StartTime

回放开始时间

StopTime

回放结束时间

See Also

[NET_DVR_MATRIX_DEC_REMOTE_PLAY_EX](#)

7.80 NET_DVR_PPPCFG_V30:PPP 参数结构体

```
struct{
    NET\_DVR\_IPADDR    struRemotelP;
    NET\_DVR\_IPADDR    struLocalIP;
    char              sLocalIPMask[16];
    BYTE              sUsername[NAME_LEN];
    BYTE              sPassword[PASSWD_LEN];
    BYTE              byPPPMoDe;
    BYTE              byRedial;
    BYTE              byRedialMoDe;
    BYTE              byDataEncrypt;
    DWORD             dwMTU;
    char              sTelephoneNumber[PHONENUMBER_LEN];
}NET_DVR_PPPCFG_V30, *LPNET_DVR_PPPCFG_V30;
```

Members

struRemotelP

远端 IP 地址

struLocalIP

本地 IP 地址

sLocalIPMask

本地 IP 地址掩码

sUsername

用户名

sPassword

密码

byPPPMoDe

PPP 模式: 0- 主动, 1- 被动

byRedial

是否回拨: 0-否, 1-是

byRedialMoDe

回拨模式: 0-由拨入者指定, 1-预置回拨号码

byDataEncrypt

数据加密: 0-否, 1-是

dwMTU

MTU

sTelephoneNumber

电话号码

See Also

[NET_DVR_RS232CFG_V30](#)

7.81 NET_DVR_PPPOECFG:PPPoE 配置结构体

```
struct{
    DWORD          dwPPPOE;
    BYTE           sPPPoEUser[NAME_LEN];
    char           sPPPoEPassword[PASSWD_LEN];
    NET\_DVR\_IPADDR struPPPoEIP;
}NET_DVR_PPPOECFG, *LPNET_DVR_PPPOECFG;
```

Members

dwPPPOE

是否启用 PPPoE: 0-不启用, 1-启用

sPPPoEUser

PPPoE 用户名

sPPPoEPassword

PPPoE 密码

struPPPoEIP

PPPoE IP 地址

See Also

[NET_DVR_NETCFG_V30](#)

7.82 NET_DVR_PU_STREAM_CFG_V41:动态解码参数

```
struct{
    DWORD          dwSize;
    BYTE           byStreamMode;
    BYTE           byRes1[3];
    NET\_DVR\_DEC\_STREAM\_MODE uDecStreamMode;
```

```

    BYTE                byRes2[64];
}NET_DVR_PU_STREAM_CFG_V41,*LPNET_DVR_PU_STREAM_CFG_V41;

```

Members

dwSize

结构体大小

byStreamMode

取流模式：0- 无效，1- 通过 IP 或域名取流，2- 通过 URL 取流，3- 通过动态域名解析向设备取流

byRes1

保留，置为 0

uDecStreamMode

取流配置信息

byRes2

保留，置为 0

See Also

[NET_DVR_MatrixStartDynamic_V41](#)

7.83 NET_DVR_PU_STREAM_URL:URL 取流配置

```

struct{
    BYTE    byEnable;
    BYTE    strURL[240];
    BYTE    byTransPortocol;
    WORD    wIPID;
    BYTE    byChannel;
    BYTE    byRes[7];
}NET_DVR_PU_STREAM_URL,*LPNET_DVR_PU_STREAM_URL;

```

Members

byEnable

是否启用：0- 禁用，1- 启用

strURL

取流 URL 路径

byTransPortocol

传输协议类型：0-TCP，1-UDP

wIPID

设备 ID 号，wIPID = iDevInfoIndex + iGroupNO*64 +1

byChannel

设备通道号

byRes

保留，置为 0

See Also

[NET_DVR_DEC_STREAM_MODE](#)

7.84 NET_DVR_RECTCFG:窗口设置结构体

```
struct{
    WORD    wXCoordinate;
    WORD    wYCoordinate;
    WORD    wWidth;
    WORD    wHeight;
}NET_DVR_RECTCFG, *LPNET_DVR_RECTCFG;
```

Members

wXCoordinate
矩形左上角起始点 X 坐标

wYCoordinate
矩形左上角 Y 坐标

wWidth
矩形宽度

wHeight
矩形高度

See Also

[NET_DVR_AREA_ZOOM_CFG](#)

7.85 NET_DVR_RECTCFG_EX:矩形窗口参数

```
struct{
    DWORD    dwXCoordinate;
    DWORD    dwYCoordinate;
    DWORD    dwWidth;
    DWORD    dwHeight;
    BYTE     byRes[4];
}NET_DVR_RECTCFG_EX, *LPNET_DVR_RECTCFG_EX;
```

Members

dwXCoordinate
矩形左上角起始点 X 坐标，对于电视墙窗口该值为基准 X 坐标的倍数

dwYCoordinate
矩形左上角 Y 坐标，对于电视墙窗口该值为基准 Y 坐标的倍数

dwWidth
矩形宽度，对于电视墙窗口该值为基准 X 坐标的倍数

dwHeight
矩形高度，对于电视墙窗口该值为基准 Y 坐标的倍数

byRes
保留，置为 0

See Also

[NET_DVR_VIDEOALLDISPLAYPOSITION](#)、[NET_DVR_MSC_SCREEN_PARAM_COND](#)、[NET_DVR_SCREEN_CONTROL_V41](#)、[NET_DVR_MSC_SCREEN_REMOTE_CFG](#)、[NET_DVR_MSC_SPLICE_CFG](#)、

[NET_DVR_MSC_SCREEN_PIP_CFG](#)、[NET_DVR_VIDEO_WALL_AREA](#)

7.86 [NET_DVR_RS232CFG_V30](#):RS232 串口参数结构体

```
struct{
    DWORD                dwSize;
    NET\_DVR\_SINGLE\_RS232 struRs232[MAX_SERIAL_PORT];
    NET\_DVR\_PPPCFG\_V30   struPPPConfig;
}NET_DVR_RS232CFG_V30, *LPNET_DVR_RS232CFG_V30;
```

Members

dwSize

结构体大小

struRs232

只有一个 232 串口，采用第一个

struPPPConfig

PPP 参数

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.87 [NET_DVR_SCENE_CONTROL_INFO](#):场景控制信息

```
struct{
    DWORD                dwSize;
    NET\_DVR\_VIDEO\_WALL\_INFO struVideoWallInfo;
    DWORD                dwCmd;
    BYTE                 byRes[4];
}NET_DVR_SCENE_CONTROL_INFO, *LPNET_DVR_SCENE_CONTROL_INFO;
```

Members

dwSize

结构体大小

struVideoWallInfo

电视墙信息

dwCmd

场景控制命令，1-场景模式切换（如果要切换的是当前场景，则不进行切换），2-初始化场景（将此场景的配置清空，如果是当前场景，则同时对当前场景进行清屏操作），3-强制切换（无论是否是当前场景，强制切换），4-保存当前模式到某场景

byRes

保留

See Also

[NET_DVR_RemoteControl](#)

7.88 NET_DVR_SCHEDULETIME:起止时间段参数

```
struct{
    BYTE    byStartHour;
    BYTE    byStartMin;
    BYTE    byStopHour;
    BYTE    byStopMin;
}NET_DVR_SCHEDULETIME, *LPNET_DVR_SCHEDULETIME;
```

Members

byStartHour

开始时间：时

byStartMin

开始时间：分

byStopHour

结束时间：时

byStopMin

结束时间：分

See Also

[NET_DVR_VIOLST](#)、[NET_DVR_MOTION](#)、[NET_DVR_HIDEALARM](#)

7.89 NET_DVR_SCREEN_BASE_INFO:屏幕基本信息联合体

```
union{
    BYTE                byScreenAddress[MAX_SCREEN_ADDRESS_LEN];
    NET\_DVR\_DEV\_IP\_INFO struIPInfo;
    BYTE                byRes[100];
}NET_DVR_SCREEN_BASE_INFO, *LPNET_DVR_SCREEN_BASE_INFO;
```

Members

byScreenAddress

屏幕特征码，设备添加方式(byDevAddType)为 0 时有效

struIPInfo

屏幕 IP 地址信息，设备添加方式(byDevAddType)为 1 时有效

byRes

联合体大小，100 字节

See Also

[NET_DVR_SCREEN_INDEX_CFG](#)

7.90 NET_DVR_SCREEN_CONTROL_PARAM:屏幕控制参数联合体

```
union{
    NET\_DVR\_INPUT\_INTERFACE\_CTRL    struInputCtrl;
    NET\_DVR\_DISPLAY\_COLOR\_CTRL     struDisplayCtrl;
```

```

NET\_DVR\_DISPLAY\_POSITION\_CTRL    struPositionCtrl;
NET\_DVR\_SIMULATE\_REMOTE\_CTRL    struSimulateRemoteCtrl;
NET\_DVR\_SCREEN\_WALL\_CTRL       struScreenWallCtrl;
NET\_DVR\_AREA\_ZOOM\_CFG          struZoomArea;
BYTE                                byRes[16];
}NET_DVR_SCREEN_CONTROL_PARAM,*LPNET_DVR_SCREEN_CONTROL_PARAM;

```

Members

struInputCtrl
输入接口控制（dwCommand 为 3 时有效）

struDisplayCtrl
显示单元颜色控制（dwCommand 为 4 时有效）

struPositionCtrl
显示单元位置控制（dwCommand 为 5 时有效）

struSimulateRemoteCtrl
模拟遥控按键（dwCommand 为 6 时有效）

struScreenWallCtrl
屏幕维墙参数（dwCommand 为 9 时有效）

struZoomArea
区域放大还原参数（dwCommand 为 13 时有效）

byRes
保留

See Also

[NET_DVR_SCREEN_CONTROL_V41](#)

7.91 NET_DVR_SCREEN_CONTROL_V41: 屏幕控制结构体

```

struct{
    DWORD                dwSize;
    BYTE                 byRes[3];
    BYTE                 byProtocol;
    DWORD                dwCommand;
NET\_DVR\_SCREEN\_CONTROL\_PARAM struControlParam;
    BYTE                 byWallNo;
    BYTE                 byRes1[3];
NET\_DVR\_RECTCFG\_EX      struRect;
    BYTE                 byRes2[28];
}NET_DVR_SCREEN_CONTROL_V41,*LPNET_DVR_SCREEN_CONTROL_V41;

```

Members

dwSize
结构体大小

byRes
保留，置为 0

byProtocol
串口协议类型：1-LCD-S1, 2-LCD-S2, 3-LCD-L1, 4-LCD-DLP, 5-LCD-S3, 6-LCD-H1, 7-LCD-L2, 8-LCD-Z1,

9-LCD-D20/D, 10-LCD-D20/S, 11-LCD-D20/P, 12-LCD-D20/T, 13-LCD-D20/F, 14-LCD-D20/DF, 15-LCD-D20/FS, 16-LCD-D20/FP, 17-LCD-D20/FT, 18-LCD-D5022, 19-LCD-D5032, 20-LCD-D5042+, 0xff-无效(使用预先绑定的协议类型)

dwCommand

控制方法: 1-开, 2-关, 3-屏幕输入源选择, 4-显示单元颜色控制, 5-显示单元位置控制, 6-模拟遥控按键, 7-显示屏幕状态, 8-显示屏幕特征码 (此命令矩形区域起始坐标设为(0,0), 宽高均设为 0xffffffff), 9-屏幕维墙, 10-恢复屏幕默认参数, 11-显示 IP (此命令矩形区域起始坐标设为(0,0), 宽高均设为 0xffffffff), 12-屏幕 USB 升级, 13-区域放大还原

struControlParam

屏幕控制参数, dwCommand 为 3、4、5、6、9、13 时有效

byWallNo

电视墙号

byRes1

保留, 置为 0

struRect

控制范围: 根据坐标进行计算控制的屏幕, 比如, 基准坐标为(1920,1920), 若同时控制左上角 4 个屏幕, 则坐标参数是(0,0,1920*2,1920*2)

byRes2

保留, 置为 0

Remarks

- 对于 LCD 拼接屏 (屏幕服务器), 支持的功能可以通过 [NET_DVR_GetDeviceAbility](#) (能力集类型: LCD_SCREEN_ABILITY) 获取。
- 对于多屏控制器, 各款屏幕支持的功能如下所示:

协议类型	支持的输入口	颜色调整	位置调整	开、关机	备注
LCD-S1	VGA、YPBPR、DVI、BNC、DVI_LOOP、BNC_LOOP, 不支持 SVIDEO、HDMI	亮度、对比度、饱和度、清晰度	不支持	开机、关机 (区域范围要求选择所有屏幕)	必须先选择输入类型再进行颜色调整或位置调整, 否则可能会失败
LCD-S2	VGA、SVIDEO、YPBPR、DVI、BNC, 不支持 DVI_LOOP、BNC_LOOP、HDMI	亮度、对比度、饱和度和、清晰度, VGA、YPBPR 和 DVI 不支持饱和度和清晰度调整	水平位置、垂直位置, VGA 和 YPBPR 输入不支持位置调整	开机、关机	LCD-S2 屏幕选择输入类型不同进行颜色调整/位置调整协议也不相同, 必须先选择输入类型再进行颜色调整或位置调整, 否则可能会失败
LCD-L1	VGA、SVIDEO、YPBPR、DVI、BNC、HDMI, 不支持 DVI_LOOP、BNC_LOOP	亮度、对比度、饱和度和、清晰度, VGA、DVI 和 HDMI 输入不支持饱和度和清晰度调整	VGA 支持水平位置和垂直位置调整, 其他输入口不支持	开机、关机	必须先选择输入类型再进行颜色调整或位置调整, 否则可能会失败
DLP	VGA、YPBPR、DVI、BNC,	亮度、对比度	水平位置、垂直位	开机、关机	必须先选择输入类

	不支持 DVI_LOOP、BNC_LOOP、SVIDEO、HDMI		置		型再进行颜色调整或位置调整，否则可能会失败
LCD-S3	/	/	/	/	/
LCD-H1	VGA、YPBPR、DVI、BNC、HDMI	亮度、对比度、饱和度和清晰度，VGA 输入不支持饱和度和清晰度调整	只有 VGA 输入支持位置调整，其余输入不支持位置调整	开机、关机	必须先选择输入类型再进行颜色调整或位置调整，否则可能会失败

See Also

[NET_DVR_RemoteControl](#)

7.92 NET_DVR_SCREEN_DISPLAY_CFG:显示参数配置结构体

```

struct{
    DWORD                dwSize;
    BYTE                 byCfgType;
    BYTE                 byBackLight;
    BYTE                 byRes1[2];
    NET\_DVR\_VIDEO\_OUT\_CFG struVideoOutCfg;
    NET\_DVR\_COLOR\_TEMPERATURE\_CFG struColorTempCfg;
    NET\_DVR\_ADC\_CFG struAdcCfg;
    NET\_DVR\_SCREEN\_EDGE\_CFG struScreenEdgeCfg;
    BYTE                 byRes[32];
}NET_DVR_SCREEN_DISPLAY_CFG,*LPNET_DVR_SCREEN_DISPLAY_CFG;

```

Members

dwSize

结构体大小

byCfgType

设置的参数类型（获取时此参数无效）：0-无效，1-背光参数，2-视频参数，3-色温参数，4-ADC 参数，5-屏幕边缘参数

byBackLight

背光参数，取值范围：0~100

byRes1

保留，置为 0

struVideoOutCfg

视频参数

struColorTempCfg

色温参数

struAdcCfg

ADC 参数

struScreenEdgeCfg

屏幕边缘参数

byRes

保留，置为 0

Remarks

显示参数：背光参数、视频参数、色温参数、ADC 参数、屏幕边缘参数，一次只能设置其中一种参数，例如：byCfgType=1 时，只需赋值背光参数(byBackLight)，而其它参数无效。

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.93 NET_DVR_SCREEN_EDGE_CFG:屏幕边缘参数结构体

```
struct{
    BYTE    byEnable;
    BYTE    byLeftEdge;
    BYTE    byRightEdge;
    BYTE    byTopEdge;
    BYTE    byLowerEdge;
    BYTE    byRes[7];
}NET_DVR_SCREEN_EDGE_CFG,*LPNET_DVR_SCREEN_EDGE_CFG;
```

Members

byEnable

是否启用屏幕边缘：0-不启用，1-启用

byLeftEdge

左边缘，取值范围：0~100

byRightEdge

右边缘，取值范围：0~100

byTopEdge

上边缘，取值范围：0~100

byLowerEdge

下边缘，取值范围：0~100

byRes

保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)、[NET_DVR_SCREEN_DISPLAY_CFG](#)

7.94 NET_DVR_SCREEN_FAN_WORK_MODE_CFG:风扇工作方式配置结构体

```
struct{
    DWORD    dwSize;
    BYTE     byWorkMode;
    BYTE     byTemperatureLimitValue;
    BYTE     byRes[14];
}
```

```
}NET_DVR_SCREEN_FAN_WORK_MODE_CFG,*LPNET_DVR_SCREEN_FAN_WORK_MODE_CFG;
```

Members

dwSize

结构体大小

byWorkMode

风扇工作方式：1-常开，2-自动

byTemperatureLimitValue

风扇开启的温度阈值（工作模式为自动时该参数有效），取值范围：0~100，单位：摄氏度

byRes

保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)

7.95 NET_DVR_SCREEN_INDEX_CFG: 屏幕索引相关参数配置结构体

```
struct{
    DWORD                dwSize;
    BYTE                 byEnbale;
    BYTE                 byDevAddType;
    BYTE                 byRes1[2];
    NET\_DVR\_SCREEN\_BASE\_INFO struScreenBaseInfo;
    BYTE                 byRes[32];
}NET_DVR_SCREEN_INDEX_CFG, *LPNET_DVR_SCREEN_INDEX_CFG;
```

Members

dwSize

结构体大小

byEnbale

使能：0- 不使能，1- 使能

byDevAddType

设备添加方式：0- 按特征码(服务器和屏幕使用串口通信)，1- 按 IP 地址(服务器和屏幕使用网络通信)

byRes1

保留，置为 0

struScreenBaseInfo

屏幕设备基本信息

byRes

保留，置为 0

See Also

[NET_DVR_GetDeviceConfig](#)、[NET_DVR_SetDeviceConfig](#)

7.96 NET_DVR_SCREEN_MENU_CFG: 屏幕菜单配置结构体

```
struct{
    DWORD    dwSize;
    BYTE     byMenuLanguage;
    BYTE     byTransparency;
    BYTE     byDuration;
    BYTE     byRes[13];
}NET_DVR_SCREEN_MENU_CFG,*LPNET_DVR_SCREEN_MENU_CFG;
```

Members

dwSize

结构体大小

byMenuLanguage

语言：1-中文，2-英文

byTransparency

透明度：0-不透明，1-低，2-中，3-高

byDuration

持续时间：0-常开，1-5 秒，2-15 秒，3-30 秒

byRes

保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)

7.97 NET_DVR_SCREEN_VGA_CFG: VGA 信号配置结构体

```
struct{
    DWORD    dwSize;
    BYTE     byEnableAutoAdjust;
    BYTE     byHorizontalPosition;
    BYTE     byVerticalPosition;
    BYTE     byClock;
    BYTE     byPhase;
    BYTE     byRes[11];
}NET_DVR_SCREEN_VGA_CFG,*LPNET_DVR_SCREEN_VGA_CFG;
```

Members

dwSize

结构体大小

byEnableAutoAdjust

是否自动调整：0-否，1-是，当自动调整启用时以下参数无效

byHorizontalPosition

水平位置，取值范围：0~100

byVerticalPosition

垂直位置，取值范围：0~100

byClock

时钟，取值范围：0~100

byPhase

相位，取值范围：0~100

byRes

保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)

7.98 NET_DVR_SCREEN_WALL_CTRL: 屏幕维墙参数结构体

```
union{
```

```
    BYTE    byEnable;
```

```
    BYTE    byRes[15];
```

```
}NET_DVR_SCREEN_WALL_CTRL,*LPNET_DVR_SCREEN_WALL_CTRL;
```

Members

byEnable

屏幕维墙状态开关：0- 关闭，1- 开启

byRes

保留，置为 0

See Also

[NET_DVR_SCREEN_CONTROL_PARAM](#)

7.99 NET_DVR_SCREEN_WORK_STATE: 屏幕工作状态参数结构体

```
struct{
```

```
    DWORD    dwSize;
```

```
    BYTE    byFanStatus;
```

```
    BYTE    byFanExceptionStatus;
```

```
    BYTE    byRes1[2];
```

```
    DWORD    dwWorkingHours;
```

```
    BYTE    byVersion[VERSION_LEN];
```

```
    int     iTemperature;
```

```
    BYTE    byRes[32];
```

```
}NET_DVR_SCREEN_WORK_STATE,*LPNET_DVR_SCREEN_WORK_STATE;
```

Members

dwSize

结构体大小

byFanStatus

风扇状态：0- 关闭，1- 开启

byFanExceptionStatus

风扇异常状态：0- 不支持，1- 正常，2- 异常

byRes1

保留，置为 0

dwWorkingHours

累积工作时长，单位：小时

byVersion

软件版本号

iTemperature

板卡温度，精确到个位，单位：摄氏度

byRes

保留，置为 0

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)

7.100 NET_DVR_SDKSTATE:SDK 状态信息

```
struct{
    DWORD    dwTotalLoginNum;
    DWORD    dwTotalRealPlayNum;
    DWORD    dwTotalPlayBackNum;
    DWORD    dwTotalAlarmChanNum;
    DWORD    dwTotalFormatNum;
    DWORD    dwTotalFileSearchNum;
    DWORD    dwTotalLogSearchNum;
    DWORD    dwTotalSerialNum;
    DWORD    dwTotalUpgradeNum;
    DWORD    dwTotalVoiceComNum;
    DWORD    dwTotalBroadCastNum;
    DWORD    dwRes[10];
}NET_DVR_SDKSTATE,*LPNET_DVR_SDKSTATE;
```

Members

dwTotalLoginNum

当前注册的用户数

dwTotalRealPlayNum

当前实时预览的路数

dwTotalPlayBackNum

当前回放或下载的路数

dwTotalAlarmChanNum

当前建立报警通道的路数

dwTotalFormatNum

当前硬盘格式化的路数

dwTotalFileSearchNum

当前文件搜索的路数

dwTotalLogSearchNum

当前日志搜索的路数

dwTotalSerialNum

当前建立透明通道的路数

`dwTotalUpgradeNum`

当前升级的路数

`dwTotalVoiceComNum`

当前语音转发的路数

`dwTotalBroadCastNum`

当前语音广播的路数

`dwRes`

保留，置为 0

See Also

[NET_DVR_GetSDKState](#)

7.101 NET_DVR_SDKABL:SDK 功能信息

```
struct{
    DWORD    dwMaxLoginNum;
    DWORD    dwMaxRealPlayNum;
    DWORD    dwMaxPlayBackNum;
    DWORD    dwMaxAlarmChanNum;
    DWORD    dwMaxFormatNum;
    DWORD    dwMaxFileSearchNum;
    DWORD    dwMaxLogSearchNum;
    DWORD    dwMaxSerialNum;
    DWORD    dwMaxUpgradeNum;
    DWORD    dwMaxVoiceComNum;
    DWORD    dwMaxBroadCastNum;
    DWORD    dwRes[10];
}NET_DVR_SDKABL,*LPNET_DVR_SDKABL;
```

Members

`dwMaxLoginNum`

最大注册用户数

`dwMaxRealPlayNum`

最大实时预览的路数

`dwMaxPlayBackNum`

最大回放或下载的路数

`dwMaxAlarmChanNum`

最大建立报警通道的路数

`dwMaxFormatNum`

最大硬盘格式化的路数

`dwMaxFileSearchNum`

最大文件搜索的路数

`dwMaxLogSearchNum`

最大日志搜索的路数

`dwMaxSerialNum`

最大建立透明通道的路数

dwMaxUpgradeNum

最大升级的路数

dwMaxVoiceComNum

最大语音转发的路数

dwMaxBroadCastNum

最大语音广播的路数

dwRes

保留，置为 0

See Also

[NET_DVR_GetSDKAbility](#)

7.102 NET_DVR_SERIAL_CONTROL:串口信息结构体

```
struct{
    DWORD    dwSize;
    BYTE     bySerialNum;
    BYTE     byRes1[3];
    BYTE     bySerial[32];
    BYTE     byRes2[32];
}NET_DVR_SERIAL_CONTROL,*LPNET_DVR_SERIAL_CONTROL;
```

Members

dwSize

结构体大小

bySerialNum

串口个数

byRes1

保留

bySerial

有效串口序号，数组的前 bySerialNum 个数据有效，bySerial[0]=1 表示 COM1

byRes2

保留

See Also

[NET_DVR_GetDVRConfig](#)

7.103 NET_DVR_SETUPALARM_PARAM:报警布防参数

```
struct{
    DWORD    dwSize;
    BYTE     byLevel;
    BYTE     byAlarmInfoType;
    BYTE     byRetAlarmTypeV40;
    BYTE     byRetDevInfoVersion;
```

```

    BYTE    byRetVQDAlarmType;
    BYTE    byRes[11];
}NET_DVR_SETUPALARM_PARAM, *LPNET_DVR_SETUPALARM_PARAM;

```

Members**dwSize**

结构体大小

byLevel

布防等级，LCD 屏不支持，保留，设为 0

byAlarmInfoType

智能交通报警信息上传类型，LCD 屏不支持，保留，设为 0

byRetAlarmTypeV40

LCD 屏不支持，保留，设为 0

byRetDevInfoVersion

LCD 屏不支持，保留，设为 0

byRetVQDAlarmType

VQD 报警上传类型类型，LCD 屏不支持，保留，设为 0

byRes

保留，置为 0

See Also[NET_DVR_SetupAlarmChan_V41](#)**7.104 NET_DVR_SIMULATE_REMOTE_CTRL:模拟遥控按键参数结构体**

```

union{
    BYTE    byControlType;
    BYTE    byControlParam;
    BYTE    byRes[14];
}NET_DVR_SIMULATE_REMOTE_CTRL,*LPNET_DVR_SIMULATE_REMOTE_CTRL;

```

Members**byControlType**

遥控按键类型：1-电源开，2-电源关，3-信号源，4-菜单，5-确定，6-上，7-下，8-左，9-右，10-INFO，11-返回上一层，12-数字按键

byControlParam

控制参数：操作类型为 12 时表示按键值 0~9，其他操作时此参数无效

byRes

保留

See Also[NET_DVR_SCREEN_CONTROL_PARAM](#)**7.105 NET_DVR_SINGLE_RS232:RS232 串口参数**

```

struct{
    DWORD    dwBaudRate;

```

```

BYTE    byDataBit;
BYTE    byStopBit;
BYTE    byParity;
BYTE    byFlowcontrol;
DWORD   dwWorkMode;

```

```
}NET_DVR_SINGLE_RS232, *LPNET_DVR_SINGLE_RS232;
```

Members

dwBaudRate

波特率(bps), 0- 50, 1- 75, 2- 110, 3- 150, 4- 300, 5- 600, 6- 1200, 7- 2400, 8- 4800, 9- 9600, 10- 19200, 11- 38400, 12- 57600, 13- 76800, 14- 115.2k

byDataBit

数据有几位: 0- 5 位, 1- 6 位, 2- 7 位, 3- 8 位

byStopBit

停止位: 0- 1 位, 1- 2 位

byParity

是否校验: 0- 无校验, 1- 奇校验, 2- 偶校验

byFlowcontrol

是否流控: 0- 无, 1- 软流控, 2- 硬流控

dwWorkMode

工作模式, 0- 窄带传输(232 串口用于 PPP 拨号), 1- 控制台(232 串口用于参数控制), 2- 透明通道, 3- ptz 模式 (审讯温湿度传感器), 4- 报警盒模式

See Also

[NET_DVR_RS232CFG_V30](#) 、 [NET_MATRIX_ANALOGMATRIX](#)

7.106 NET_DVR_SNAP_ABILITY:智能交通摄像机能力集参数结构体

```

struct{
    DWORD    dwSize;
    BYTE     byIoInNum;
    BYTE     byIoOutNum;
    BYTE     bySingleSnapNum;
    BYTE     byLightModeArrayNum;
    BYTE     byMeasureModeArrayNum;
    BYTE     byPlateEnable;
    BYTE     byLensMode;
    BYTE     byPreTriggerSupport;
    DWORD    dwAbilityType;
    BYTE     byIoSpeedGroup;
    BYTE     byIoLightGroup;
    BYTE     byRecogRegionType;
    BYTE     bySupport;
    WORD     wSupportMultiRadar;
    BYTE     byICRPresetNum;
    BYTE     byICRTIMEslot;

```

```

BYTE    bySupportRS485Num;
BYTE    byExpandRs485SupportSensor;
BYTE    byExpandRs485SupportSignalLampDet;
BYTE    byRelayNum;
BYTE    bySupport1;
BYTE    bySupport2;
BYTE    bySupportWhiteBalance;
BYTE    byRes[9];

```

```
}NET_DVR_SNAP_ABILITY, *LPNET_DVR_SNAP_ABILITY;
```

Members

dwSize

结构体大小

byIoInNum

IO 输入口数

byIoOutNum

IO 输出口数

bySingleSnapNum

单 IO 触发组数

byLightModeArrayNum

红绿灯模式组数

byMeasureModeArrayNum

测速模式组数

byPlateEnable

车牌识别能力, 0-没有, 1-有

byLensMode

前端镜头模式: 0-CCD, 1-CMOS

byPreTriggerSupport

是否支持原触发模式: 0- 支持, 1- 不支持

dwAbilityType

支持的触发模式能力, 按位表示, 定义如下所示:

```

enum _ITC_MAINMODE_ABILITY_{
    ITC_MODE_UNKNOW        = 0x0,        //无
    ITC_POST_MODE          = 0x1,        //卡口模式
    ITC_EPOLICE_MODE       = 0x2,        //电警模式
    ITC_POSTEPOLICE_MODE   = 0x4        //卡式电警模式
} ITC_MAINMODE_ABILITY

```

byIoSpeedGroup

支持的 IO 测速组数

byIoLightGroup

支持的 IO 红绿灯组数

byRecogRegionType

牌识区域支持的类型, 定义如下所示:

```
enum _ITC_RECOG_REGION_TYPE_{
```

```

ITC_REGION_RECT          = 0x0,          //矩形
ITC_REGION_POLYGON      = 0x1          //多边形
} ITC_RECOG_REGION_TYPE

```

bySupport

设备能力，按位表示，值：0-不支持，1-支持

bySupport&0x1, 表示是否支持扩展的字符叠加配置

bySupport&0x2, 表示是否支持扩展的校时配置结构

bySupport&0x4, 表示是否支持多网卡(多网隔离)

bySupport&0x8, 表示是否支持网卡的 bonding 功能(网络容错)

bySupport&0x10, 表示是否支持语音对讲

bySupport&0x20, 表示是否支持单 IO 触发界面 IO 使能配置

bySupport&0x40, 表示是否支持扩展的 FTP 接口

bySupport&0x80, 表示是否支持云存储接口

wSupportMultiRadar

设备能力 (RS485 口关联雷达能力)，按位表示，值：0-不支持，1-支持

wSupportMultiRadar&0x1, 表示卡口 RS485 雷达是否支持车道关联雷达处理

wSupportMultiRadar&0x2, 表示卡口虚拟线圈是否支持车道关联雷达处理

wSupportMultiRadar&0x8, 表示视频检测是否支持车道关联雷达处理

byICRPresetNum

表示支持的 ICR 预置点 (滤光片偏移点) 数

byICRTIMEslot

表示支持的 ICR 的时间段数 (1~8)

bySupportRS485Num

表示支持的 RS485 口的数量

byExpandRs485SupportSensor

设备能力 (RS485 口关联车检器能力)，按位表示，值：0-不支持，1-支持

byExpandRs485SupportSensor&0x1, 表示电警车检器是否支持车检器

byExpandRs485SupportSensor&0x2, 表示卡式电警车检器是否支持车检器

byExpandRs485SupportSignalLampDet

设备能力 (RS485 口关联信号灯检测器能力)，按位表示，值：0-不支持，1-支持

byExpandRs485SupportSignalLampDet&0x1, 表示电警车检器是否支持外接信号灯检测器

byExpandRs485SupportSignalLampDet&0x2, 表示卡式电警车检器是否支持外接信号灯检测器

byExpandRs485SupportSignalLampDet&0x4, 表示视频电警是否支持外接信号灯检测器

byRelayNum

表示支持继电器的个数

bySupport1

设备能力 (扩展)，按位表示，值：0-不支持，1-支持

bySupport1&0x1, 表示是否支持自定义校时信息配置

bySupport1&0x2, 表示是否支持视频触发优化方案

bySupport1&0x4, 表示是否支持 DDNS 配置

bySupport1&0x8, 表示是否支持获取设备当前触发模式接口命令

bySupport1&0x10, 表示是否支持出入口配置

bySupport1&0x20, 表示是否支持 GPS 模块参数配置

bySupport1&0x40, 表示是否支持 NTP 校时间隔支持分钟设置

bySupport1&0x80, 表示是否支持 TPS 报警数据上传

bySupport2

设备能力（扩展），按位表示，值：0-不支持，1-支持

bySupport2&0x01, 表示是否支持码流附加信息配置

bySupport2&0x02, 表示是否支持通道图像参数中视频叠加时间毫秒使能参数 (byOSDMilliSecondEnable)

bySupportWhiteBalance

白平衡能力（默认都是支持手动白平衡、自动白平衡、自动白平衡 2 这 3 种模式），按位表示，值：0-不支持，1-支持

bySupportWhiteBalance &0x01, 表示是否支持日光灯模式

bySupportWhiteBalance &0x02, 表示是否支持自然光模式

bySupportWhiteBalance &0x04, 表示是否支持暖光灯模式

bySupportWhiteBalance &0x08, 表示是否支持白炽灯模式

byRes

保留，置为 0

See Also

[NET_DVR_GetDeviceAbility](#)

7.107 NET_DVR_STREAM_MEDIA_SERVER:流媒体服务器参数

```
struct{
    BYTE    byValid;
    BYTE    byRes1[3];
    BYTE    byAddress[MAX_DOMAIN_NAME];
    WORD    wDevPort;
    BYTE    byTransmitType;
    BYTE    byRes2[5];
}NET_DVR_STREAM_MEDIA_SERVER,*LPNET_DVR_STREAM_MEDIA_SERVER;
```

Members

byValid

是否启用：0- 否，1- 是

byRes1

保留，置为 0

byAddress

IP 或者域名

wDevPort

流媒体服务器端口

byTransmitType

传输协议类型：0- TCP，1- UDP

byRes2

保留，置为 0

See Also

[NET_DVR_DEC_STREAM_DEV_EX](#)、[NET_DVR_DEC_DDNS_DEV](#)

7.108 NET_DVR_SWITCH_DAY_TIME: 开关机计划参数结构体

```
struct{
    BYTE                byEnable;
    BYTE                bySwitchType;
    BYTE                byRes1[2];
    NET\_DVR\_DAYTIME    struTimePoint;
    BYTE                byRes2[8];
}NET_DVR_SWITCH_DAY_TIME,*LPNET_DVR_SWITCH_DAY_TIME;
```

Members

byEnable

是否启用该时间点：0- 不启用，1- 启用

bySwitchType

开关机类型：0- 关机，1- 开机

byRes1

保留，置为 0

struTimePoint

开关机计划时间点

byRes2

保留，置为 0

See Also

[NET_DVR_TIME_SWITCH_CFG](#)

7.109 NET_DVR_SWITCH_TIME_POINT: 按时间点开关机参数结构体

```
struct{
    BYTE                byEnable;
    BYTE                bySwitchType;
    BYTE                byRes1[2];
    NET\_DVR\_SYSTEM\_TIME struTimePoint;
    BYTE                byRes2[16];
}NET_DVR_SWITCH_TIME_POINT,*LPNET_DVR_SWITCH_TIME_POINT;
```

Members

byEnable

是否启用该时间点：0- 不启用，1- 启用

bySwitchType

开关机类型：0- 关机，1- 开机

byRes1

保留，置为 0

struTimePoint

开关机时间点

byRes2

保留，置为 0

See Also[NET_DVR_TIME_SWITCH_CFG](#)**7.110 NET_DVR_SYSTEM_TIME:时间信息结构体**

```
struct{  
    WORD    wYear;  
    WORD    wMonth;  
    WORD    wDay;  
    WORD    wHour;  
    WORD    wMinute;  
    WORD    wSecond;  
    WORD    wMilliSec;  
    BYTE    byRes[2];  
}NET_DVR_SYSTEM_TIME,*LPNET_DVR_SYSTEM_TIME;
```

Members

wYear
 年

wMonth
 月

wDay
 日

wHour
 时

wMinute
 分

wSecond
 秒

wMilliSec
 毫秒

byRes
 保留

See Also[NET_DVR_SWITCH_TIME_POINT](#)、[NET_DVR_LOG_V30](#)、[NET_DVR_PLAY_BACK_BY_TIME](#)**7.111 NET_DVR_TIME:时间参数**

```
struct{  
    DWORD    dwYear;  
    DWORD    dwMonth;  
    DWORD    dwDay;  
    DWORD    dwHour;  
    DWORD    dwMinute;
```

```
DWORD    dwSecond;
}NET_DVR_TIME, *LPNET_DVR_TIME;
```

Members

dwYear

年

dwMonth

月

dwDay

日

dwHour

时

dwMinute

分

dwSecond

秒

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)、[NET_DVR_FindDVRLog_V30](#)、[NET_DVR_PLAY_BACK_BY_TIME](#)、[NET_DVR_MATRIX_DEC_CHAN_INFO_V41](#)

7.112 NET_DVR_TIMEPOINT:时间点参数

```
struct{
    DWORD    dwMonth;
    DWORD    dwWeekNo;
    DWORD    dwWeekDate;
    DWORD    dwHour;
    DWORD    dwMin;
}NET_DVR_TIMEPOINT, *LPNET_DVR_TIMEPOINT;
```

Members

dwMonth

月： [0,11]取值分别表示第 1 个月到第 12 个月

dwWeekNo

周： 0- 第 1 周， 1- 第 2 周， 2- 第 3 周， 3- 第 4 周， 4- 最后一周

dwWeekDate

星期： 0- 星期日， 1- 星期一， 2- 星期二， 3- 星期三， 4- 星期四， 5- 星期五， 6- 星期六

dwHour

小时， 开始时间： 0~23， 结束时间： 1~23

dwMin

分： 0~59

See Also

[NET_DVR_ZONEANDDST](#)

7.113 NET_DVR_TIME_SWITCH_CFG:时间点参数

```
struct
{
    DWORD                dwSize;
    NET\_DVR\_SWITCH\_DAY\_TIME    struSwitchDayTime[MAX_DAYS][MAX_DAY_TIME_POINT]; //周开关机
    计划
    NET\_DVR\_SWITCH\_TIME\_POINT    struSwitchTimePoint[MAX_TIME_POINT]; //按时间点开关机参数
    BYTE byRes[64];
}NET_DVR_TIME_SWITCH_CFG,*LPNET_DVR_TIME_SWITCH_CFG;
```

Members

dwSize

结构体大小

struSwitchDayTime

周开关机计划

struSwitchTimePoint

按时间点开关机参数

See Also

[NET_DVR_GetDeviceConfig](#)、[NET_DVR_SetDeviceConfig](#)

7.114 NET_DVR_USING_SERIALPORT: 当前使用串口信息结构体

```
struct{
    DWORD    dwSize;
    DWORD    dwSerialPort;
    BYTE     byProtocolType;
    BYTE     byRes[31];
}NET_DVR_USING_SERIALPORT,*LPNET_DVR_USING_SERIALPORT;
```

Members

dwSize

结构体大小

dwSerialPort

当前使用串口号

byProtocolType

串口协议类型: 1- LCD-S1, 2- LCD-S2, 3- LCD-L1, 4- LCD-DLP, 5- LCD-S3, 6- LCD-D20, 7- LCD-L2, 8- LCD-Z1, 9- LCD-D20/D, 10- LCD-D20/S, 11- LCD-D20/P, 12- LCD-D20/T, 13- LCD-D20/F, 14- LCD-D20/DF, 15- LCD-D20/FS, 16- LCD-D20/FP, 17- LCD-D20/FT, 18- LCD-D5022, 19- LCD-D5032, 20- LCD-D5042+

byRes

保留, 置为 0

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.115 NET_VCA_BEHAVIOR_ABILITY:异常行为识别能力输出参数结构体

```
struct{
    DWORD    dwSize;
    DWORD    dwAbilityType;
    BYTE     byMaxRuleNum;
    BYTE     byMaxTargetNum;
    BYTE     bySupport;
    BYTE     byRes[5];
    DWORD    dwAbilityTypeEx;
}NET_VCA_BEHAVIOR_ABILITY,*LPNET_VCA_BEHAVIOR_ABILITY;
```

Members

dwSize

结构体大小

dwAbilityType

支持的异常行为识别能力类型，按位表示，具体定义如下：

```
enum _VCA_ABILITY_TYPE_{
    TRAVERSE_PLANE_ABILITY    = 0x01,        //穿越警戒面
    ENTER_AREA_ABILITY        = 0x02,        //进入区域
    EXIT_AREA_ABILITY         = 0x04,        //离开区域
    INTRUSION_ABILITY         = 0x08,        //入侵
    LOITER_ABILITY            = 0x10,        //徘徊
    LEFT_TAKE_ABILITY         = 0x20,        //丢包捡包
    PARKING_ABILITY           = 0x40,        //停车
    RUN_ABILITY                = 0x80,        //奔跑
    HIGH_DENSITY_ABILITY      = 0x100,       //区域内人员密度
    LF_TRACK_ABILITY          = 0x200,       //球机跟踪
    VIOLENT_MOTION_ABILITY    = 0x400,       //剧烈运动检测
    REACH_HIGHT_ABILITY       = 0x800,       //攀高检测
    GET_UP_ABILITY            = 0x1000,      //起身检测
    LEFT_ABILITY               = 0x2000,     //丢包检测
    TAKE_ABILITY               = 0x4000,     //捡包检测
    LEAVE_POSITION            = 0x8000,     //离岗
    TRAIL_ABILITY              = 0x10000,    //尾随
    KEY_PERSON_GET_UP_ABILITY = 0x20000,    //重点人员起身检测
    STANDUP_ABILITY           = 0x40000,    //起立
    FALL_DOWN_ABILITY         = 0x80000,    //倒地
    AUDIO_ABNORMAL_ABILITY    = 0x100000,   //声强突变
    ADV_REACH_HEIGHT_ABILITY  = 0x200000,   //折线攀高
    TOILET_TARRY_ABILITY      = 0x400000,   //如厕超时
    YARD_TARRY_ABILITY        = 0x800000,   //放风场滞留
    ADV_TRAVERSE_PLANE_ABILITY = 0x1000000, //折线警戒面
    HUMAN_ENTER_ABILITY       = 0x10000000, //人靠近 ATM，只在 ATM_PANEL 模式下支持
```

```

OVER_TIME_ABILITY      = 0x20000000,      //操作超时, 只在 ATM_PANEL 模式下支持
STICK_UP_ABILITY      = 0x40000000,      //贴纸条, 支持区域规则
INSTALL_SCANNER_ABILITY = 0x80000000      //安装读卡器,支持区域规则
} VCA_ABILITY_TYPE

```

byMaxRuleNum

最大规则数

byMaxTargetNum

最大目标数

bySupport

支持的功能类型, 按位表示。bySupport & 0x01 支持标定功能 byBFullChanNum

byRes

保留, 置为 0

dwAbilityTypeEx

支持的能力类型, 按位表示, 具体定义如下:

```

enum _VCA_ABILITY_TYPE_EX_{
    PEOPLENUM_CHANGE_ABILITY = 0x00000002,      //人数变化检测
    SPACING_CHANGE_ABILITY   = 0x00000004,      //间距变化检测
    EVENT_COMBINED_ABILITY   = 0x00000008,      //组合规则事件
    EVENT_SIT_QUIETLY        = 0x00000010      //一动不动事件
} VCA_ABILITY_TYPE

```

See Also

[NET_DVR_GetDeviceAbility](#)

7.116 NET_VCA_DEV_ABILITY:智能设备能力集结构体

struct{

```

DWORD    dwSize;
BYTE     byVCAChanNum;
BYTE     byPlateChanNum;
BYTE     byBBaseChanNum;
BYTE     byBAdvanceChanNum;
BYTE     byBFullChanNum;
BYTE     byATMChanNum;
BYTE     byPDCCChanNum;
BYTE     byITSChanNum;
BYTE     byBPrisonChanNum;
BYTE     byFSnapChanNum;
BYTE     byFSnapRecogChanNum;
BYTE     byFRetrievalChanNum;
BYTE     bySupport;
BYTE     byFR recogChanNum;
BYTE     byBPPerimeterChanNum;

```

```

BYTE    byTPSChanNum;
BYTE    byTFSCChanNum;
BYTE    byFSnapBFullChanNum;
BYTE    byHeatMapChanNum;
BYTE    bySmartVehicleNum;
BYTE    bySmartHVTNum;
BYTE    bySmartNum;
BYTE    byVehicleNum;
BYTE    bySmartRoadDetectionNum;
BYTE    bySmartFaceDetectionNum;
BYTE    bySmartHeatMapNum;
BYTE    byRes[14];
}NET_VCA_DEV_ABILITY,*LPNET_VCA_DEV_ABILITY;

```

Members

dwSize

结构体大小

byVCAChanNum

智能通道个数

byPlateChanNum

车牌通道个数

byBBaseChanNum

行为基本版个数

byBAdvanceChanNum

行为高级版个数

byBFullChanNum

行为完整版个数

byATMChanNum;

智能 ATM 通道个数

byPDCChanNum

人数统计通道个数

byITSChanNum

交通事件通道个数

byBPrisonChanNum

行为监狱版通道个数

byFSnapChanNum

人脸抓拍通道个数

byFSnapRecogChanNum

人脸抓拍和识别通道个数

byFRetrievalChanNum

人脸后检索个数

bySupport

能力，位与结果为 0 表示不支持，1 表示支持

bySupport&0x1，表示是否支持智能跟踪

bySupport&0x2，表示是否支持 128 路取流扩展

byFRecogChanNum
人脸识别通道个数

byBPPerimeterChanNum
行为监狱版(周界)通道个数

byTPSChanNum
交通诱导通道个数

byTFChanNum
道路违章取证（违章事件）通道个数

byFSnapBFullChanNum
人脸抓拍和异常行为识别通道个数

byHeatMapChanNum
热度图通道个数

bySmartVehicleNum
SMART 事件+车辆检测通道个数

bySmartHVTNum
SMART 事件+混行检测通道个数

bySmartNum
SMART 事件个数

byVehicleNum
车辆检测通道个数

bySmartRoadDetectionNum
SMART 事件+道路监控通道个数

bySmartFaceDetectionNum
SMART 事件+人脸侦测通道个数

bySmartHeatMapNum
SMART 事件+热度图通道个数

byRes[14]
保留，置为 0

See Also[NET_DVR_GetDeviceAbility](#)**7.117 NET_DVR_VIHOST:信号丢失报警参数结构体**

```
struct{
    BYTE                byEnableHandleVILost;
    NET\_DVR\_HANDLEEXCEPTION strVILostHandleType;
    NET\_DVR\_SCHEDTIME      struAlarmTime[MAX_DAYS][MAX_TIMESEGMENT];
}NET_DVR_VIHOST, *LPNET_DVR_VIHOST;
```

Members

byEnableHandleVILost
是否处理信号丢失报警：0-不处理，1-处理

strVILostHandleType
处理方式参数

struAlarmTime

布防时间参数

See Also

[NET_DVR_PICCFG](#)

7.118 NET_DVR_VCS_USER_INFO:用户信息结构体

```
struct{
    DWORD                dwSize;
    BYTE                 sUserName[NAME_LEN];
    BYTE                 sPassWord[PASSWD_LEN];
    NET\_DVR\_IPADDR      struUserIP;
    BYTE                 byMacAddr[MACADDR_LEN];
    BYTE                 byPriority;
    BYTE                 byRes1;
    DWORD                dwRight;
    NET\_DVR\_ASSOCIATE\_INPUT\_PARAM struInputParam[MAX_INPUTNUMS];
    NET\_DVR\_ASSOCIATE\_OUTPUT\_PARAM struOutputParam[MAX_OUTPUTNUMS];
    BYTE                 byRes2[128];
}NET_DVR_VCS_USER_INFO,*LPNET_DVR_VCS_USER_INFO;
```

Members

dwSize

结构体大小

sUserName

用户名，最大 16 字节

sPassWord

密码

struUserIP

绑定用户 IP 地址(为 0 时表示允许任何地址)

byMacAddr

物理地址绑定(为 0 时表示允许任何 MAC 地址)

byPriority

优先级：1-管理员，2-操作员，管理员具有任何权限，操作员没有"管理用户"的权限

byRes1

保留，置为 0

dwRight

远程操作权限，按位表示：0-不支持，1-支持，各 32 位代表的权限类型如下所示：

bit0-获取设置基本参数，bit1-显示屏管理，bit2-窗口管理，bit3-信号源管理，
bit4-布局管理，bit5-预案管理，bit6-底图管理，bit7-OSD 管理，bit8-回显权限，
bit9-报警布防，bit10-远程升级，bit11-恢复默认设置，bit12-导入导出配置，
bit13-日志查询，bit14-远程重启，bit15-屏幕区域管理

struInputParam

关联的输入设备

struOutputParam

关联的输出显示屏

byRes2

保留，置为 0

Remarks

- 多屏控制器具有一个 ADMIN 管理员用户和 7 个普通用户：管理员登陆时，普通用户不能操作；管理员没有登陆时，几个普通用户可以同时操作。管理员具有所有操作权限（包括屏幕和设备），能够添加、修改、删除普通用户信息。
- 参数 struUserIP、byMacAddr、struInputParam、struOutputParam 在集中式多屏控制器 V2.1 中无效。
- 设置操作员用户参数时，用户名（sUserName）为空表示删除相应用户。

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.119 NET_DVR_VIDEO_OUT_CFG: 视频参数结构体

```
struct{
    BYTE    byDisplayMode;
    BYTE    byBrightnessLevel;
    BYTE    byContrastLevel;
    BYTE    bySharpnessLevel;
    BYTE    bySaturationLevel;
    BYTE    byHueLevel;
    BYTE    byImageMode;
    BYTE    byRes[5];
}NET_DVR_VIDEO_OUT_CFG,*LPNET_DVR_VIDEO_OUT_CFG;
```

Members

byDisplayMode

显示模式：1-标准，2-冷色，3-暖色，0xff-自定义

byBrightnessLevel

亮度值，取值范围：0~100

byContrastLevel

对比度，取值范围：0~100

bySharpnessLevel

锐度，取值范围：0~100

bySaturationLevel

饱和度，取值范围：0~100

byHueLevel

色调，取值范围：0~100

byImageMode

图像模式：0- 正常模式，1- 夜模式 1，2- 夜模式 2，3- 夜模式 3，4- 日照模式 1，5- 日照模式 2，6- 日照模式 3

byRes

保留，置为 0

Remarks

- byDisplayMode 取值为“1-标准”、“2-冷色”或者“3-暖色”时，亮度、对比度、锐度、饱和度、色调这些参数自动调整，不支持设置。

- 当屏幕信号源为 VGA 时只配置对比度和亮度两个值。

See Also

[NET_DVR_MSC_SCREEN_PARAM](#)、[NET_DVR_SCREEN_DISPLAY_CFG](#)

7.120 NET_DVR_VIDEO_PLATFORM: 显示通道配置联合体

```
union{
    BYTE                byRes[160];
    struct{
        BYTE            byJoinDecoderId[MAX_WINDOWS_V41];
        BYTE            byDecResolution[MAX_WINDOWS_V41];
        BYTE            byRes[88];
    }struVideoPlatform;
    struct{
        BYTE            byRes[160];
    }struNotVideoPlatform;
}NET_DVR_VIDEO_PLATFORM, *LPNET_DVR_VIDEO_PLATFORM;
```

Members

byRes

保留

byJoinDecoderId

各个子窗口对应解码通道所对应的解码子系统的槽位号(对于视频综合平台中解码子系统有效)

byDecResolution

显示窗口所解视频分辨率：1- D1，2- 720P，3- 1080P，设备端需要根据此分辨率进行解码通道的分配。如 1 分屏配置成 1080P，则设备会把 4 个解码通道都分配给此显示窗口。

Remarks

在 V41 以下的版本中，byDecResolution[MAX_WINDOWS_V41]是 byDecResolution，V41 版本中改为数组。在低版本向高版本转换时，根据 NET_DVR_MATRIX_VOUTCFG 中的画面模式 dwWindowMode 对 byDecResolution[MAX_WINDOWS_V41]进行赋值；高版本向低版本转化时用 byDecResolution[0]。

See Also

[NET_DVR_MATRIX_VOUTCFG](#)

7.121 NET_DVR_VIDEO_WALL_AREA: 电视墙区域信息结构体

```
struct{
    DWORD                dwSize;
    DWORD                byWallNo;
    BYTE                byRes1[3];
    NET\_DVR\_RECTCFG\_EX  struRect;
    BYTE                byRes2[32];
}NET_DVR_VIDEO_WALL_AREA,*LPNET_DVR_VIDEO_WALL_AREA;
```

Members

dwSize

结构体大小

byWallNo

电视墙号

byRes1

保留，置为 0

struRect

屏幕区域范围

byRes2

保留，置为 0

Remarks

目前不支持按墙区域设置定时开关机参数，即 struRect 无效。

See Also

[NET_DVR_GetDeviceConfig](#)、[NET_DVR_SetDeviceConfig](#)

7.122 NET_DVR_VIDEO_WALL_INFO:电视墙信息

```
struct{
    DWORD    dwSize;
    DWORD    dwWindowNo;
    DWORD    dwSceneNo;
    BYTE     byRes[20];
}NET_DVR_VIDEO_WALL_INFO,*LPNET_DVR_VIDEO_WALL_INFO;
```

Members

dwSize

结构体大小

dwWindowNo

窗口号：1 字节墙号+3 字节保留

dwSceneNo

场景号

byRes

保留，置为 0

See Also

[NET_DVR_SCENE_CONTROL_INFO](#)、[NET_DVR_GetDeviceConfig](#)、[NET_DVR_SetDVRConfig](#)

7.123 NET_DVR_VIDEOEFFECT:视频参数

```
struct{
    BYTE    byBrightnessLevel;
    BYTE    byContrastLevel;
    BYTE    bySharpnessLevel;
    BYTE    bySaturationLevel;
    BYTE    byHueLevel;
    BYTE    byEnableFunc;
```

```

    BYTE    byLightInhibitLevel;
    BYTE    byGrayLevel;
}NET_DVR_VIDEOEFFECT, *LPNET_DVR_VIDEOEFFECT;

```

Members

byBrightnessLevel

亮度，取值范围：[0,100]

byContrastLevel

对比度，取值范围：[0,100]

bySharpnessLevel

锐度，取值范围：[0,100]

bySaturationLevel

饱和度，取值范围：[0,100]

byHueLevel

色度，取值范围：[0,100]，保留

byEnableFunc

使能，按位表示。bit0-SMART IR(防过曝)，bit1-低照度，bit2-强光抑制使能，值：0-否，1-是，例如 byEnableFunc&0x2==1 表示使能低照度功能； bit3-锐度类型，值：0-自动，1-手动。

byLightInhibitLevel

强光抑制等级，取值范围：[1,3]

byGrayLevel

灰度值域:0-[0,255]，1-[16,235]

See Also

[NET_DVR_WALLOUTPUTPARAM](#)

7.124 NET_DVR_VIDEOWALLDISPLAYPOSITION:电视墙输出位置配置

```

struct{
    DWORD          dwSize;
    BYTE           byEnable;
    BYTE           byRes1[3];
    DWORD          dwVideoWallNo;
    DWORD          dwDisplayNo;
    NET\_DVR\_RECTCFG\_EX struRectCfg;
    BYTE           byRes2[64];
}NET_DVR_VIDEOWALLDISPLAYPOSITION,*LPNET_DVR_VIDEOWALLDISPLAYPOSITION;

```

Members

dwSize

结构体大小

byEnable

使能：0- 禁用，1- 启用

byRes1

保留，置为 0

dwVideoWallNo

电视墙号(组合)

dwDisplayNo

显示输出号(组合), 批量获取全部时有效

struRectCfg

位置坐标, 须为基准坐标(通过能力集获取)的整数倍, 宽度和高度值不用设置, 即为基准值

byRes2

保留, 置为 0

See Also

[NET_DVR_GetDeviceConfig](#)、[NET_DVR_SetDeviceConfig](#)

7.125 NET_DVR_VIDEOWALLWINDOWPOSITION:电视墙窗口信息

```
struct{
    DWORD          dwSize;
    BYTE           byEnable;
    BYTE           byRes1[7];
    DWORD          dwWindowNo;
    DWORD          dwLayerIndex;
    NET\_DVR\_RECTCFG\_EX struRect;
    BYTE           byRes2[64];
}NET_DVR_VIDEOWALLWINDOWPOSITION, *LPNET_DVR_VIDEOWALLWINDOWPOSITION;
```

Members

dwSize

结构体大小

byEnable

窗口使能: 0- 禁用, 1- 启用

byRes1

保留, 置为 0

dwWindowNo

窗口号(组合), 批量获取全部时有效

dwLayerIndex

窗口相对应的图层号, 只能获取, 不能设置, 图层号由设备关联

struRect

目的窗口坐标区域(相对显示墙)

byRes2

保留, 置为 0

Remarks

电视墙中开窗口默认置顶, 窗口置顶置底操作通过 [NET_DVR_RemoteControl](#) 实现。

See Also

[NET_DVR_GetDeviceConfig](#)、[NET_DVR_SetDeviceConfig](#)

7.126 NET_DVR_WALLOUTPUTPARAM:电视墙显示输出参数

struct

```

{
    DWORD          dwSize;
    DWORD          dwResolution;
    NET\_DVR\_VIDEOEFFECT struRes;
    BYTE           byVideoFormat;
    BYTE           byDisplayMode;
    BYTE           byBackgroundColor;
    BYTE           byRes[61];
}NET_DVR_WALLOUTPUTPARAM, *LPNET_DVR_WALLOUTPUTPARAM;

```

Members**dwSize**

结构体大小

dwResolution

输出分辨率

struRes

视频参数，保留

byVideoFormat

视频制式：0-无效，1-N 制，2-P 制

byDisplayMode

输出连接模式（只能获取，不能设置）：1- BNC，2- VGA，3- HDMI，4- DVI，0xff- 无效

byBackgroundColor

背景色：0- 不支持背景色，1- 红，2- 绿，3- 蓝，4- 黄，5- 紫，6- 青，7- 黑，8- 白

byRes

保留，置为 0

See Also[NET_DVR_GetDeviceConfig](#)、[NET_DVR_SetDeviceConfig](#)**7.127 NET_DVR_WALLSCENECFG:电视墙场景信息**

```

struct
{
    DWORD    dwSize;
    BYTE     sSceneName[NAME_LEN];
    BYTE     byEnable;
    BYTE     bySceneIndex;
    BYTE     byRes [78];
}NET_DVR_WALLSCENECFG, *LPNET_DVR_WALLSCENECFG;

```

Members**dwSize**

结构体大小

sSceneName

场景名称

byEnable

场景是否有效：0-无效，1-有效

bySceneIndex

场景号，只能获取。获取所有场景时使用该参数。

byRes

保留，置为 0

See Also

[NET_DVR_GetDeviceConfig](#)、[NET_DVR_SetDeviceConfig](#)

7.128 NET_DVR_ZONEANDDST:夏令时参数

```
struct{
    DWORD                dwSize;
    DWORD                dwZoneIndex;
    BYTE                 byRes1[12];
    DWORD                dwEnableDST;
    BYTE                 byDSTBias;
    BYTE                 byRes2[3];
    NET\_DVR\_TIMEPOINT    struBeginPoint;
    NET\_DVR\_TIMEPOINT    struEndPoint;
}NET_DVR_ZONEANDDST,*LPNET_DVR_ZONEANDDST;
```

Members

dwSize

结构体大小

dwZoneIndex

该参数请赋值为 [NET_DVR_GetDVRConfig](#) 获取的值，设置时区请通过设置 [NET_DVR_NTTPARA](#) 中的 cTimeDifferenceH 或 cTimeDifferenceM 来实现

byRes1

保留，置为 0

dwEnableDST

是否启用夏时制：0- 不启用，1- 启用

byDSTBias

夏令时偏移值（以分钟计）：30min，60min，90min，120min

byRes2

保留，置为 0

struBeginPoint

夏时制开始时间

struEndPoint

夏时制停止时间

See Also

[NET_DVR_GetDVRConfig](#)、[NET_DVR_SetDVRConfig](#)

7.129 NET_MATRIX_ANALOGMATRIX:模拟矩阵参数结构体

```
struct{
```



```

BYTE                bySerPortNum;
BYTE                byMatrixSerPortType;
BYTE                byRes1[2];
NET\_DVR\_SINGLE\_RS232 struRS232;
BYTE                byRes2[200];
}NET_MATRIX_ANALOGMATRIX, *LPNET_MATRIX_ANALOGMATRIX;

```

Members**bySerPortNum**

连接的串口号，0xff 表示环通接入

byMatrixSerPortType

该参数对于多屏控制器无效，置为 0

byRes1

保留，置为 0

struRS232

RS232 串口参数，环通接入时此参数无效

byRes2

保留，置为 0

See Also[NET_MATRIX_UNION](#)**7.130 NET_MATRIX_DIGITALMATRIX:数字矩阵参数结构体**

```

struct{
NET\_DVR\_IPADDR    struAddress;
WORD                wPort;
BYTE                byNicNum;
BYTE                byRes[69];
}NET_MATRIX_DIGITALMATRIX, *LPNET_MATRIX_DIGITALMATRIX;

```

Members**struAddress**

设备为数字设备时的 IP 信息

wPort

端口号

byNicNum

绑定的网口：0 - eth0，1 - eth1，考虑双网口如何通信加入绑定的网口

byRes

保留，置为 0

See Also[NET_MATRIX_UNION](#)**7.131 NET_MATRIX_UNION:矩阵参数联合体**

```

union{

```

```

NET\_MATRIX\_DIGITALMATRIX    struDigitalMatrix;
NET\_MATRIX\_ANALOGMATRIX    struAnalogMatrix;
}NET_MATRIX_UNION,*LPNET_MATRIX_UNION;

```

Members

```

struDigitalMatrix
    数字矩阵信息
struAnalogMatrix
    模拟矩阵信息

```

See Also

[NET_DVR_EXTERNAL_MATRIX_CFG](#)

7.132 LCD 屏幕能力 XML 描述

```

<!-- req, 屏幕能力集描述 -->
<ScreenAbility version="2.0">
  <SimulateRemoteControl>
    <controlType opt="1,2,3,4,5,6,7,8,9,10,11" />
    <!-- req, 控制类型: 1-电源开, 2-电源关, 3-信号源, 4-菜单, 5-确定, 6-上, 7-下, 8-左, 9-右, 10-INFO,
    11-返回上一层 -->
  </SimulateRemoteControl>

  <ScreenSignalCFG>
    <signalSourceType opt="1,2,3,4,5,6,7,8,9,10" />
    <!-- req, 信号源类型: 1-AV, 2-VGA, 3-HDMI, 4-DVI, 5-YpbPr, 6-USB, 7-IP, 8-SDI, 9-DP, 10-HDBASET
    -->
  </ScreenSignalCFG>

  <ScreenBaseCFG>
    <noSignalPic opt="1,2,3" />
    <!-- req, 无信号画面模式: 1-蓝屏, 2-黑屏, 3-LOGO 画面-->
  </ScreenBaseCFG>

  <ScreenSenceCfg>
    <senceNo min="1" max="16" />
    <!-- req, 场景号, 1~16 -->
  </ScreenSenceCfg>

  <ScreenFanWorkMode>
    <workMode opt="1,2" />
    <!-- req, 风扇工作模式: 1-常开, 2-自动 -->
    <temperatureLimitValue min="0" max="100" />
    <!-- req, 风扇开启的温度阈值, 0~100 摄氏度 -->
  </ScreenFanWorkMode>

```

```

<ScreenVGACFG>
  <horizontalPosition min="0" max="100" />
  <!-- req,水平位置, 0~100 -->
  <verticalPosition min="0" max="100" />
  <!--req,垂直位置, 0~100 -->
  <clock min="0" max="100" />
  <!--req,时钟, 0~100-->
  <phase min="0" max="100" />
  <!-- req,相位, 0~100 -->
</ScreenVGACFG>

<ScreenMenuCFG>
  <menuLanguage opt="1,2" />
  <!-- req, 菜单语言, 1-中文, 2-英文 -->
  <transparency opt="0,1,2,3" />
  <!-- req, 菜单透明度,0-不透明, 1-低, 2-中, 3-高-->
  <durationopt opt="0,1,2,3" />
  <!-- req, 持续时间, 0-常开, 1-5 秒, 2-15 秒, 3-30 秒-->
</ScreenMenuCFG>

<ScreenDisplayCFG>
  <brightnessLevel min="0" max="100" />
  <!-- req, 亮度值, 0~100 -->
  <contrastLevel min="0" max="100" />
  <!-- req, 对比值, 0~100 -->
  <sharpnessLevel min="0" max="100" />
  <!-- req, 锐度, 0~100-->
  <saturationLevel min="0" max="100" />
  <!--req, 饱和度, 0~100 -->
  <hueLevel min="0" max="100" />
  <!-- req, 色调, 0~100 -->
  <displayMode
    opt="selfdefine,standard,cold,warm,nightmode1,nightmode2,nightmode3,daymode1,daymode2,daymode3" />
  <!--
    req, 1-标准, 2-冷色, 3-暖色, 4-夜模式 1, 5-夜模式 2, 6-夜模式 3, 7-日照模式 1, 8-日照模式 2,
    9-日照模式 3, 0xff-自定义 -->
  <!-- req, 支持的显示模式, 不返回此节点默认支持 selfdefine,standard,cold,warm-->
</ScreenDisplayCFG>

<ScreenBacklightCFG>
  <backLight min="0" max="100" />
  <!-- req,背光, 0~100 -->
</ScreenBacklightCFG>

```

```
<ColorTemperatureCFG>
  <red min="0" max="100" />
  <!-- req, 色温配置, 红色值; 0~100 -->
  <green min="0" max="100" />
  <!--req, 色温配置, 绿色值; 0~100 -->
  <blue min="0" max="100" />
  <!-- req, 色温配置, 蓝色值; 0~100 -->
  <redOffset min="0" max="100" />
  <!-- req, 色温配置, 红色偏移; 0~100 -->
  <greenOffset min="0" max="100" />
  <!-- req, 色温配置, 绿色偏移; 0~100 -->
  <blueOffset min="0" max="100" />
  <!-- req, 色温配置, 蓝色偏移; 0~100 -->
</ColorTemperatureCFG>
```

```
<ADCCCFG>
  <gainR min="0" max="100" />
  <!-- req, 增益 R; 0~100 -->
  <gainG min="0" max="100" />
  <!-- req, 增益 G; 0~100 -->
  <gainB min="0" max="100" />
  <!-- req, 增益 B; 0~100 -->
  <offsetR min="0" max="100" />
  <!-- req, 偏移量 R; 0~100 -->
  <offsetG min="0" max="100" />
  <!-- req, 偏移量 G; 0~100 -->
  <offsetB min="0" max="100" />
  <!-- req, 偏移量 B; 0~100 -->
```

```
</ADCCCFG>
```

```
<ScreenEdgeCFG>
  <enable opt="0,1" />
  <!-- req, 是否启用屏幕边缘 0-不启用 1-启用-->
  <leftEdge min="0" max="100" />
  <!-- req, 左边缘,0-100 -->
  <rightEdge min="0" max="100" />
  <!-- req, 右边缘,0-100-->
  <topEdge min="0" max="100" />
  <!-- req, 上边缘,0-100 -->
  <lowerEdge min="0" max="100" />
  <!-- req, 下边缘,0-100 -->
```

```
</ScreenEdgeCFG>
```

```
<PIPCFG>
```

```

<enable opt="0,1" />
<!-- req, 是否启用画中画 0-不启用 1-启用 -->
<subWindowSource opt="1,2,3,4,5,6,7" />
<!-- req, 子画面信号源, 1-AV, 2-VGA, 3-HDMI, 4-DVI, 5-YpbPr, 6-USB, 7-IP -->
<subWindowBorderColor opt="0,1" />
<!-- req, 子画面边框颜色, 1-蓝色, 2-黑色-->
<subWindowSize min="0" max="100" />
<!-- req, 子画面大小,0-100 -->
<subWindowXCoordinate min="0" max="100" />
<!-- req, 子画面位置, X坐标 0-100 -->
<subWindowYCoordinate min="0" max="100" />
<!-- req, 子画面位置, Y坐标 0-100 -->
</PIPCFG>

<DefogLCD>
  <defogEnable opt="0,1" />
  <!-- req, 是否开启去雾开关, 0~关闭, 1~开启-->
  <defogModel opt="0,1" />
  <!--req, 去雾模式, 0~自动去雾, 1~手动去雾 -->
  <defogLevel opt="1,2,3,4,5,6,7" />
  <!--
  req, 去雾强度等级, 1-7个等级, 默认等级为4。等级1为0.5; 等级2为0.55; 等级3为0.60;
  等级4为0.65; 等级5为0.70; 等级6为0.75; 等级7为0.80。(去雾模式为手动去雾时,
  可配置去雾强度等级)
  -->
</DefogLCD>

<ScreenWorkStatus>
  <fanStatus opt="off,on" />
  <!-- req,风扇状态 -->
  <fanExceptionStatus>true</fanExceptionStatus>
  <!--req,风扇异常状态 -->
  <workingHours>true</workingHours>
  <!-- req, 累积工作时长 -->
  <versionLen min="" max="" />
  <!-- req, 软件版本号长度 -->
  <temperature min="" max="" />
  <!-- req, 板卡温度, 单位: 摄氏度 -->
</ScreenWorkStatus>

<bootLogoCfg opt="on,off" />
<!-- req, 开机 logo 是否显示 -->
<delayBoot min="" max="" />
<!-- req, 延时开机时间范围, 单位: ms -->

```

```

<TimingSwitch>
  <!-- req,支持定时开关机 -->
  <weekTimingSwitch>
    <!--req,周开关机计划 -->
    <maxPowerOnTimePoint>2</maxPowerOnTimePoint>
    <!-- req, 开机时间点个数 -->
    <maxPowerOffTimePoint>2</maxPowerOffTimePoint>
    <!-- req, 关机时间点个数 -->
    <TimeAccuracy>
      <!-- req,时间精度 -->
      <hour>true</hour>
      <minute>true</minute>
      <second>true</second>
    </TimeAccuracy>
  </weekTimingSwitch>

  <specialTimingSwitch>
    <maxPowerOnTimePoint>8</maxPowerOnTimePoint>
    <!-- req, 开机时间点个数 -->
    <maxPowerOffTimePoint>8</maxPowerOffTimePoint>
    <!-- req, 关机时间点个数 -->
    <TimeAccuracy>
      <!--req,时间精度 -->
      <hour>true</hour>
      <minute>true</minute>
      <second>true</second>
    </TimeAccuracy>
  </specialTimingSwitch>
</TimingSwitch>

<WallCfg>
  <wallNo min="1" max="1" />
  <!-- req, 支持的墙号范围 -->
  <WallScale>
    <!-- req, 墙规模大小 -->
    <length min="" max="" />
    <!-- req,墙宽 -->
    <length min="" max="" />
    <!-- req,墙高 -->
  </WallScale>
</WallCfg>

<ExternalMatrix>

```

```

<!-- req,外接矩阵能力 -->
<maxMatrixNum>4</maxMatrixNum>
<!-- req,最大外接矩阵台数 -->
<matrixNameLen min="" max="" />
<!-- req,外接矩阵名称长度 -->
<maxMatrixInputNum>256</maxMatrixInputNum>
<!-- req,支持外接矩阵最大输入数 -->
<maxMatrixOutputNum>256</maxMatrixOutputNum>
<!-- req,支持外接矩阵最大输出数 -->
<matrixChanType opt="BNC,VGA,RGB,DVI" />
<!-- req,支持的矩阵输出通道类型 -->
<matrixProtocol opt="ZT1.0,ZT1.0,Extron,Creator,SelfDefine" />
<!-- req,支持的矩阵协议 -->
<matrixType opt="analogmatrix,digitalmatrix" />
<!-- req,支持的矩阵类型 -->
<DigitalMatrix>
  <!-- req,数字矩阵能力 -->
  <nicNum min="" max="" />
  <!-- req,支持的网口编号 -->
</DigitalMatrix>
<AnalogMatrix>
  <serPortNum min="" max="" />
  <!-- req,控制器串口号范围,串口参数范围根据串口能力集 -->
</AnalogMatrix>
</ExternalMatrix>

<serialProtocol
  opt="LCD-S1,LCD-S2,LCD-L1,LCD-DLP,LCD-S3,LCD-D20,LCD-L2,LCD-Z1,LCD-D20/D,
  LCD-D20/S,LCD-D20/P,LCD-D20/T,LCD-D20/F,LCD-D20/DF,LCD-D20/FS,LCD-D20/
  FP, LCD-D20/FT,LCD-D5022,LCD-D5032,LCD-D5042+,LCD-D20/Z" />
<!-- req, 支持的屏幕控制协议 -->
<controlType opt="serial,network" />
<!-- req,支持的控制方式 -->

<AudioCfg>
  <!-- req,音频参数能力 -->
  <mute>enable</mute>
  <!-- req,支持静音 -->
  <volume min="" max="" />
  <!-- req,音量大小 -->
  <balance min="" max="" />
  <!-- req,声道平衡 -->
</AudioCfg>

```

```

<AreaZoom>
  <!-- req,是否支持屏幕区域放大, 不支持不返回 -->
  <enabled>true</enabled>
</AreaZoom>

<LCDAlarm>
  <OnOffLine>true</OnOffLine>
  <!-- req,支持在线情况上报 -->
  <FanState>true</FanState>
  <!-- req,支持风扇状态上报 -->
  <FanException>true</FanException>
  <!-- req,支持风扇异常状态上报 -->
  <Temperature>true</Temperature>
  <!-- req,支持屏幕板卡温度上报 -->
  <TemperatureState>true</TemperatureState>
  <!-- req,支持板卡温度状态上报 -->
</LCDAlarm>
</ScreenAbility>

```

7.133 日志类型

dwMajorType

日志主类型

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

dwMinorType

日志次类型

主类型的宏定义	宏定义值	含义
MAJOR_ALARM	0x1	报警
次类型的宏定义	宏定义值	含义
MINOR_ALARM_IN	0x1	报警输入
MINOR_ALARM_OUT	0x2	报警输出
MINOR_MOTDET_START	0x3	移动侦测报警开始
MINOR_MOTDET_STOP	0x4	移动侦测报警结束
MINOR_HIDE_ALARM_START	0x5	遮挡报警开始

MINOR_HIDE_ALARM_STOP	0x6	遮挡报警结束
MINOR_VCA_ALARM_START	0x7	智能报警开始
MINOR_VCA_ALARM_STOP	0x8	智能报警结束
MINOR_NETALARM_START	0x0b	网络报警开始
MINOR_NETALARM_STOP	0x0c	网络报警结束

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_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

主类型的宏定义	宏定义值	含义
MAJOR_OPERATION	0x3	操作
次类型的宏定义	宏定义值	含义
MINOR_START_DVR	0x41	开机
MINOR_STOP_DVR	0x42	关机
MINOR_STOP_ABNORMAL	0x43	异常关机
MINOR_REBOOT_DVR	0x44	本地重启设备
MINOR_LOCAL_LOGIN	0x50	本地登陆
MINOR_LOCAL_LOGOUT	0x51	本地注销登陆
MINOR_LOCAL_CFG_PARM	0x52	本地配置参数
MINOR_LOCAL_PLAYBYFILE	0x53	本地按文件回放或下载
MINOR_LOCAL_PLAYBYTIME	0x54	本地按时间回放或下载

MINOR_LOCAL_START_REC	0x55	本地开始录像
MINOR_LOCAL_STOP_REC	0x56	本地停止录像
MINOR_LOCAL_PTZCTRL	0x57	本地云台控制
MINOR_LOCAL_PREVIEW	0x58	本地预览(保留不使用)
MINOR_LOCAL_MODIFY_TIME	0x59	本地修改时间(保留不使用)
MINOR_LOCAL_UPGRADE	0x5a	本地升级
MINOR_LOCAL_RECFILE_OUTPUT	0x5b	本地备份录像文件
MINOR_LOCAL_FORMAT_HDD	0x5c	本地初始化硬盘
MINOR_LOCAL_CFGFILE_OUTPUT	0x5d	导出本地配置文件
MINOR_LOCAL_CFGFILE_INPUT	0x5e	导入本地配置文件
MINOR_LOCAL_COPYFILE	0x5f	本地备份文件
MINOR_LOCAL_LOCKFILE	0x60	本地锁定录像文件
MINOR_LOCAL_UNLOCKFILE	0x61	本地解锁录像文件
MINOR_LOCAL_DVR_ALARM	0x62	本地手动清除和触发报警
MINOR_IPC_ADD	0x63	本地添加 IPC
MINOR_IPC_DEL	0x64	本地删除 IPC
MINOR_IPC_SET	0x65	本地设置 IPC
MINOR_LOCAL_START_BACKUP	0x66	本地开始备份
MINOR_LOCAL_STOP_BACKUP	0x67	本地停止备份
MINOR_LOCAL_COPYFILE_START_TIME	0x68	本地备份开始时间
MINOR_LOCAL_COPYFILE_END_TIME	0x69	本地备份结束时间
MINOR_LOCAL_ADD_NAS	0x6a	本地添加网络硬盘
MINOR_LOCAL_DEL_NAS	0x6b	本地删除 NAS 盘
MINOR_LOCAL_SET_NAS	0x6c	本地设置 NAS 盘
MINOR_REMOTE_LOGIN	0x70	远程登录
MINOR_REMOTE_LOGOUT	0x71	远程注销登陆
MINOR_REMOTE_START_REC	0x72	远程开始录像
MINOR_REMOTE_STOP_REC	0x73	远程停止录像
MINOR_START_TRANS_CHAN	0x74	开始透明传输
MINOR_STOP_TRANS_CHAN	0x75	停止透明传输
MINOR_REMOTE_GET_PARM	0x76	远程获取参数
MINOR_REMOTE_CFG_PARM	0x77	远程配置参数
MINOR_REMOTE_GET_STATUS	0x78	远程获取状态
MINOR_REMOTE_ARM	0x79	远程布防
MINOR_REMOTE_DISARM	0x7a	远程撤防

MINOR_REMOTE_REBOOT	0x7b	远程重启
MINOR_START_VT	0x7c	开始语音对讲
MINOR_STOP_VT	0x7d	停止语音对讲
MINOR_REMOTE_UPGRADE	0x7e	远程升级
MINOR_REMOTE_PLAYBYFILE	0x7f	远程按文件回放
MINOR_REMOTE_PLAYBYTIME	0x80	远程按时间回放
MINOR_REMOTE_PTZCTRL	0x81	远程云台控制
MINOR_REMOTE_FORMAT_HDD	0x82	远程格式化硬盘
MINOR_REMOTE_STOP	0x83	远程关机
MINOR_REMOTE_LOCKFILE	0x84	远程锁定文件
MINOR_REMOTE_UNLOCKFILE	0x85	远程解锁文件
MINOR_REMOTE_CFGFILE_OUTPUT	0x86	远程导出配置文件
MINOR_REMOTE_CFGFILE_INPUT	0x87	远程导入配置文件
MINOR_REMOTE_RECFILE_OUTPUT	0x88	远程导出录象文件
MINOR_REMOTE_DVR_ALARM	0x89	远程手动清除和触发报警
MINOR_REMOTE_IPC_ADD	0x8a	远程添加 IPC
MINOR_REMOTE_IPC_DEL	0x8b	远程删除 IPC
MINOR_REMOTE_IPC_SET	0x8c	远程设置 IPC
MINOR_REBOOT_VCA_LIB	0x8d	重启智能库
MINOR_REMOTE_ADD_NAS	0x8e	远程添加 NAS 盘
MINOR_REMOTE_DEL_NAS	0x8f	远程删除 NAS 盘
MINOR_REMOTE_SET_NAS	0x90	远程设置 NAS 盘
MINOR_START_DYNAMIC_DECODE	0xb0	开始动态解码
MINOR_STOP_DYNAMIC_DECODE	0xb1	停止动态解码
MINOR_GET_CYC_CFG	0xb2	获取解码器通道轮询配置
MINOR_SET_CYC_CFG	0xb3	设置解码通道轮询配置
MINOR_START_CYC_DECODE	0xb4	开始轮询解码
MINOR_STOP_CYC_DECODE	0xb5	停止轮询解码
MINOR_GET_DECCHAN_STATUS	0xb6	获取解码通道状态
MINOR_GET_DECCHAN_INFO	0xb7	获取解码通道当前信息
MINOR_START_PASSIVE_DEC	0xb8	开始被动解码
MINOR_STOP_PASSIVE_DEC	0xb9	停止被动解码
MINOR_CTRL_PASSIVE_DEC	0xba	控制被动解码
MINOR_RECON_PASSIVE_DEC	0xbb	被动解码重连
MINOR_GET_DEC_CHAN_SW	0xbc	获取解码通道总开关

MINOR_SET_DEC_CHAN_SW	0xbd	设置解码通道总开关
MINOR_CTRL_DEC_CHAN_SCALE	0xbe	解码通道缩放控制
MINOR_SET_REMOTE_REPLAY	0xbf	设置远程回放
MINOR_GET_REMOTE_REPLAY	0xc0	获取远程回放状态
MINOR_CTRL_REMOTE_REPLAY	0xc1	远程回放控制
MINOR_SET_DISP_CFG	0xc2	设置显示通道
MINOR_GET_DISP_CFG	0xc3	获取显示通道设置
MINOR_SET_PLANTABLE	0xc4	设置计划轮询表
MINOR_GET_PLANTABLE	0xc5	获取计划轮询表
MINOR_START_PPPOE	0xc6	开始 PPPoE 连接
MINOR_STOP_PPPOE	0xc7	结束 PPPoE 连接
MINOR_UPLOAD_LOGO	0xc8	上传 LOGO
MINOR_DISPLAY_LOGO	0x205	显示 LOGO
MINOR_HIDE_LOGO	0x206	隐藏 LOGO
MINOR_SET_DEC_DELAY_LEVEL	0x207	解码通道延时级别设置
MINOR_SET_BIGSCREEN_DIPLAY_AREA	0x208	设置大屏显示区域
MINOR_CUT_VIDEO_SOURCE	0x209	大屏视频源切割设置
MINOR_SET_BASEMAP_AREA	0x210	大屏底图区域设置
MINOR_DOWNLOAD_BASEMAP	0x211	下载大屏底图
MINOR_CUT_BASEMAP	0x212	底图切割配置
MINOR_CONTROL_ELEC_ENLARGE	0x213	电子放大操作（放大或还原）
MINOR_SET_OUTPUT_RESOLUTION	0x214	显示输出分辨率设置
MINOR_SET_TRANSPARENCY	0x215	图层透明度设置
MINOR_SET_OSD	0x216	显示 OSD 设置
MINOR_RESTORE_DEC_STATUS	0x217	恢复初始状态（场景切换时，解码恢复初始状态）
MINOR_SCREEN_SET_INPUT	0x251	修改输入源
MINOR_SCREEN_SET_DISPLAY	0x252	修改输出通道
MINOR_SCREEN_SET_OSD	0x253	修改虚拟 LED
MINOR_SCREEN_SET_LOGO	0x254	修改 LOGO
MINOR_SCREEN_SET_LAYOUT	0x255	设置布局
MINOR_SCREEN_PICTUREPREVIEW	0x256	回显操作
MINOR_SCREEN_GET_OSD	0x257	获取虚拟 LED
MINOR_SCREEN_GET_LAYOUT	0x258	获取场景
MINOR_SCREEN_LAYOUT_CTRL	0x259	场景控制
MINOR_GET_ALL_VALID_WND	0x260	获取所有有效窗口

MINOR_GET_SIGNAL_WND	0x261	获取单个窗口信息
MINOR_WINDOW_CTRL	0x262	窗口控制
MINOR_GET_LAYOUT_LIST	0x263	获取场景列表
MINOR_LAYOUT_CTRL	0x264	场景控制
MINOR_SET_LAYOUT	0x265	设置单个场景
MINOR_GET_SIGNAL_LIST	0x266	获取输入信号源列表
MINOR_GET_PLAN_LIST	0x267	获取预案列表
MINOR_SET_PLAN	0x268	修改预案
MINOR_CTRL_PLAN	0x269	控制预案
MINOR_CTRL_SCREEN	0x270	屏幕控制
MINOR_ADD_NETSIG	0x271	添加信号源
MINOR_SET_NETSIG	0x272	修改信号源
MINOR_SET_DECBDCFG	0x273	设置解码板参数
MINOR_GET_DECBDCFG	0x274	获取解码板参数
MINOR_GET_DEVICE_STATUS	0x275	获取设备信息
MINOR_UPLOAD_PICTURE	0x276	底图上传
MINOR_SET_USERPWD	0x277	设置用户密码
MINOR_ADD_LAYOUT	0x278	添加场景
MINOR_DEL_LAYOUT	0x279	删除场景
MINOR_DEL_NETSIG	0x280	删除信号源
MINOR_ADD_PLAN	0x281	添加预案
MINOR_DEL_PLAN	0x282	删除预案
MINOR_GET_EXTERNAL_MATRIX_CFG	0x283	获取外接矩阵配置
MINOR_SET_EXTERNAL_MATRIX_CFG	0x284	设置外接矩阵配置
MINOR_GET_USER_CFG	0x285	获取用户配置
MINOR_SET_USER_CFG	0x286	设置用户配置
MINOR_GET_DISPLAY_PANEL_LINK_CFG	0x287	获取显示墙连接配置
MINOR_SET_DISPLAY_PANEL_LINK_CFG	0x288	设置显示墙连接配置
MINOR_GET_WALLSCENE_PARAM	0x289	获取电视墙场景
MINOR_SET_WALLSCENE_PARAM	0x28a	设置电视墙场景
MINOR_GET_CURRENT_WALLSCENE	0x28b	获取当前使用场景
MINOR_SWITCH_WALLSCENE	0x28c	场景切换

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