

**IPCAM、视频服务器**

客户端软件开发包编程手册

V1.3

2018-9-6

第一章 简介 .....	3
<b>1.1 组成</b> .....	3
<b>1.2 注意事项</b> .....	3
<b>1.2.1 基本注意事项</b> .....	3
<b>1.2.2 软件环境要求</b> .....	3
<b>1.2.3 支持的硬件</b> .....	3
第二章 编程指南 .....	4
<b>2.1 编程说明</b> .....	4
<b>2.1.1 初始化</b> .....	4
<b>2.1.2 网络延时</b> .....	4
<b>2.1.3 透明通道</b> .....	4
<b>2.1.4 服务器搜索</b> .....	4
<b>2.1.5 录像说明</b> .....	4
<b>2.2 函数调用顺序</b> .....	5
<b>2.2.1 服务器操作函数</b> .....	5
<b>2.2.2 本地播放函数调用</b> .....	6
第三章 OCX 使用说明 .....	7
<b>3.1 OCX 接口调用</b> .....	7
<b>3.2 设置显示使用详解</b> .....	7
第四章 数据结构 .....	8
<b>4.1 错误码定义</b> .....	8
<b>4.2 结构定义</b> .....	9
<b>4.2.1 服务器基础信息</b> .....	9
<b>4.2.2 服务器消息结构</b> .....	10
<b>4.2.3 服务器流信息</b> .....	11
<b>4.2.4 服务器探头信息</b> .....	11

<b>4.2.5</b> 服务器设置结构.....	12
<b>4.3 函数定义.....</b>	19
<b>4.3.1</b> 服务器操作函数.....	19
<b>4.3.2</b> 通道操作函数.....	20
<b>4.3.3</b> 对讲操作函数.....	22
<b>4.3.4</b> 云台操作函数.....	23
<b>4.3.5</b> 主机搜索函数.....	24
<b>4.3.6</b> 其他辅助函数.....	25
<b>4.4 服务器配置结构 .....</b>	25
<b>4.4.1</b> 设置服务器信息.....	25
<b>4.4.2</b> 人脸库管理命令.....	28
<b>4.4.3</b> 获取服务器信息.....	28

# 第一章 简介

## 1.1 组成

客户端软件开发包是网络产品系列的配套产品，本手册详细介绍了此软件开发包中各个函数所实现的功能以及使用方法、各个函数之间的调用关系。

客户端软件开发包主要由客户端操作和设备端操作两大部分组成：

客户端操作：图像预览、文件回放、云台控制、语音对讲、日志管理等功能。

设备端操作：参数配置（系统配置、通道配置、串口配置、报警配置、用户配置）服务器重启动，服务器升级，服务器录像配置等功能。

本开发包包括以下文件： JBNVSDK.dll, JBNVOCX.ocx, JBNVSDK.lib, JBNVSDK.h,。

其中 JBNVSDK.dll, JBNVSDK.lib, JBNVSDK.h 客户端的操作函数库，JBNVOCX.OCX 是封装好的单路服务器操作控件，并具有配置服务器端功能。

## 1.2 注意事项

### 1.2.1 基本注意事项

⌘ 支持的网络协议有 TCP, UDP, MULTICAST。

⌘ 网络预览时，每个设备最多支持 5 个客户同时连接，当码流很大时，这个数值会有一定的下降。

### **1.2.2 软件环境要求**

此软件需要使用 DirectX 7.0 支持。CPU 需要 I3 以上，如果要解码多路图像，建议配置较高频率的 CPU 和独立显卡。

### **1.2.3 支持的硬件**

支持本公司的网络产品系列网络摄像机 (IPC)、网络硬盘录像机 (NVR)

## 第二章 编程指南

### 2.1 编程说明

#### 2.1.1 初始化

本系统不需要进行任何初始化操作就可以使用。

#### 2.1.2 网络延时

本系统使用 JBNV\_SetStreamBufferSize 设置本地解码缓冲数据包，如果最小值为 0，系统延时情况为 200ms 以内，如果网络情况良好，延时通常为 80-100 毫秒。

#### 2.1.3 透明通道

使用设置命令直接发送 CMD\_SEND\_COMDATA 命令可完成数据发送，当服务器收到 485 或 232 数据时，会通过消息通知客户端。

#### 2.1.4 服务器搜索

本系统使用更先进的方式进行局域网内服务器搜索，能够返回位于各网段的本公司的在线视频服务器。同时返回的还有各服务器的当前报警和视频状态等信息。

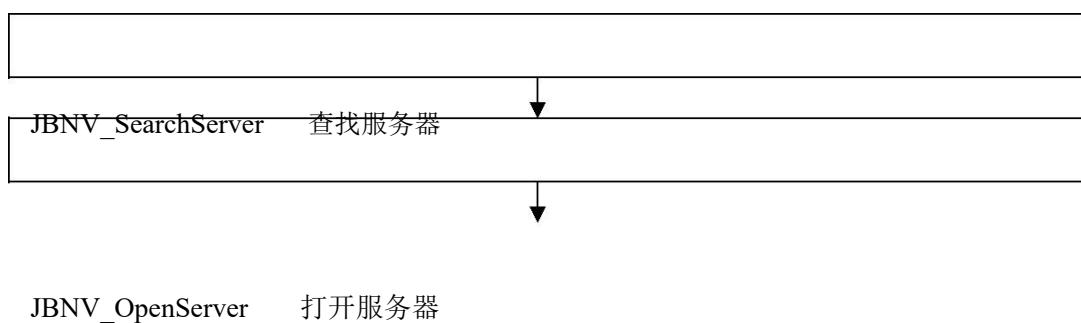
#### 2.1.5 录象说明

本系统码流符合 MPEG4, HEVC 的国际标准要求，可以使用 MediaPlayer10 或 RealPlayer 播放，同时也支持 RTSP 数据广播功能，使用手机可以直接连接观看。录像采用文件系统为微软标准 ASF 格式，能够在任何地方进行播放。

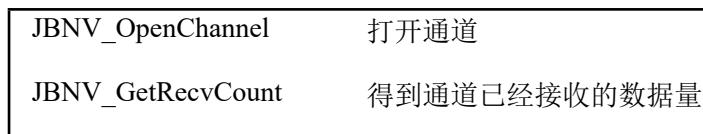
## 2.2 函数调用顺序

### 2.2.1 服务器操作函数

JBNV_SearchServer	搜索服务器
CheckIpIsInternet	检查输入 IP 是否为局域网
JBNV_GetErrorMessage	获取错误描述
GetDataCRC	获取指定数据的 CRC32 值
CheckWaterMark	检查文件水印
MakeWaterMark	为文件加上水印
GetWindowsMacInfo	获得本机的 MAC 地址（支持 WIN2000 以上系统）



JBNV_GetServerInfo	获取服务器信息
JBNV_SetMessage	设置服务器消息通知
JBNV_SetStreamCallBack	设置数据流回调



JBNV_SetStreamBufferSize	设置视频流缓冲
JBNV_SetDisplayRect	设置显示区域（可用于局部放大）
JBNV_SetSecondSurface	设置第二块显示屏(支持最多 4 个)
JBNV_Mute	音频操作（静音与否）
JBNV_SetOSDCALLBACK	OSD 回调设置 ↓
JBNV_SetDisplayhWnd	显示窗口更换
JBNV_SetVideoCallback	设置视频解码后的 YUV422 数据回调
JBNV_StartRecord	开始录像

CJBNVOCX.ShowSetup(hServer)进行服务器设置操作



JBNV_StopRecord	停止录像
JBNV_CloseChannel	关闭通道 ↓
JBNV_CloseServer	关闭服务器连接

## 2.2.2 本地播放函数调用

JBPlay_OpenFile	打开本地文件
JBPlay_SetMessage	设置通知消息
JBPlay_PlayCommand	发送播放命令

JBPlay_GetPlayInfo	获取播放进度
JBPlay_SetAudio	设置声音是否播放
JBPlay_SeekToSecond	支持拖拉
JBPlay_UpdateBound	更新图像
JBPlay_SetDisplayRect	设置显示区域 (可以用于局部放大)
JBPlay_SetOSDCallback	设置 OSD 叠加
JBPlay_SetPlayLoop	设置为循环播放
JBPlay_CapturePicture	本地捕捉图象, 存为 BMP
JBPlay_SetVolume	设置音量
JBPlay_SaveStreamData	保存部分视频数据为一个文件

JBPlay_Release	停止播放, 释放播放句柄
----------------	--------------

# 第三章 OCX 使用说明

## 3.1 OCX 接口调用

OpenServer                      CloseServer

OpenChannel

DefMessageProc

SendPTZCommand

ShowSetup

VideoRecord

isConnect

Mute

SnapShot

TalkBack

GetChannelCount

SetSelectChannel

SetShowWindows

CloseChannel

打开服务器	抓图（服务器进行抓图操作，完成后 OCX 自动按照要求保存）
打开通道	对讲操作
默认消息处理	得到当前连接的服务器的通道总数
发送云台操作命令	设置当前选择的通道（如果 OCX 可显示的话，会自动切换选择的焦点）
显示设置	设置显示的窗口数（1 或 4）
进行视频录像	关闭通道
服务器是否连接	关闭服务器
静音操作	

### 3.2 设置显示使用详解

**VC++6.0 代码:**

```
CJBNOCX *m_pJBnv = new CJBNOCX;

m_pJBnv->Create(NULL, WS_CHILD|WS_VISIBLE, rc, this, 12532);
if(m_pJBnv->GetSafeHwnd() == NULL)

    AfxMessageBox("OCX 没有注册，请重新运行安装程序进行注册");

else

    m_pJBnv->>ShowWindow(SW_HIDE);

if(m_pJBnv && m_pJBnv->GetSafeHwnd() && hServer)

{
```

```
m_pJBnv->ShowSetup(TRUE,(long) hServer,NULL);
```

```
}
```

程序结束后进行销毁

```
if(m_pJBnv)
```

```
{
```

```
    m_pJBnv->DestroyWindow();
```

```
    delete m_pJBnv;
```

```
    m_pJBnv = 0;
```

```
}
```

# 第四章 数据结构

## 4.1 错误码定义

本系统的任何 API 函数调用后均会返回一个 DWORD 返回值，其含义如下：

```
#define JB_SUCCESS          0x00000000

#define JBERR_BASE           0x10000000

#define JBERR_PASSWORD_UNMATCH (JBERR_BASE|0x01) //密码不匹配

#define JBERR_TIME_OVER       (JBERR_BASE|0x02) //操作超时

#define JBERR_INVALID_PARAM    (JBERR_BASE|0x03) //无效参数

#define JBERR_MAX_LINK         (JBERR_BASE|0x04) //超过最大连接数

#define JBERR_INVALID_HANDLE    (JBERR_BASE|0x05) //句柄非法或错误

#define JBERR_INVALID_COMMAND   (JBERR_BASE|0x06) //不被接受的命令

#define JBERR_SENDCMD_FAILD    (JBERR_BASE|0x07) //发送命令数据失败

#define JBERR_GETCONFIG_FAILD   (JBERR_BASE|0x08) //取得命令数据失败

#define JBERR_NO_LOGON          (JBERR_BASE|0x09) //没有登录

#define JBERR_ALLOC_FAILD        (JBERR_BASE|0x0a) //分配内存失败

#define JBERR_INVALID_NETADDRESS (JBERR_BASE|0x0b) //无效的或无法解析的网络地址

#define JBERR_FILE_CRC32         (JBERR_BASE|0x0c) //文件校验错

#define JBERR_SOFTVER_ERR        (JBERR_BASE|0x0d) //程序版本过低，无法校验升级文件
```

#define JBERR_CPUTYPE_ERR	(JBERR_BASE 0x0E) //升级文件不适用于此 CPU 类型的主机
#define JBERR_ERROR_10054	(JBERR_BASE 0x0F) //连接被服务器强迫关闭!
#define JBERR_ERROR_10061	(JBERR_BASE 0x10) //服务器没有在指定端口打开服务!
#define JBERR_ERROR_10060	(JBERR_BASE 0x11) //没有发现指定 IP 的服务器!
#define JBERR_ERROR_10065	(JBERR_BASE 0x12) //网络未准备好!
#define JBERR_INITSURFACE	(JBERR_BASE 0x13) //初始化显示表面错误
#define JBERR_UNSUPPORT	(JBERR_BASE 0x14) //服务器不支持此功能
#define JBERR_TALK_REJECTED	(JBERR_BASE 0x15) //对讲请求被服务器拒绝
#define JBERR_TALK_INITAUDIO	(JBERR_BASE 0x16) //启动对讲时音频初始化失败
#define JBERR_OPEN_FILE	(JBERR_BASE 0x17) //打开文件错
#define JBERR_BIND_PORT	(JBERR_BASE 0x18) //绑定本地端口失败
#define JBERR_NO_FILE	(JBERR_BASE 0x19) //没有找到文件
#define JBERR_NOMORE_FILE	(JBERR_BASE 0x1A) //没有更多文件
#define JBERR_FILE_FINDING	(JBERR_BASE 0x1B) //正在查找
#define JBERR_DISK_NOTEXIST	(JBERR_BASE 0x1C) //格式化/分区的硬盘不存在
#define JBERR_FILE_ERROR	(JBERR_BASE 0x1D) //文件不正确或者不匹配此服务器
#define JBERR_UNINITOBJ	(JBERR_BASE 0x1E) //对象没有初始化，稍候再试
#define JBERR_UNKNOW_SERVER	(JBERR_BASE 0x1F) //服务器无法识别
#define JBERR_CHANNEL_NOT_OPEN	(JBERR_BASE 0x20) //通道没有打开，操作失败
#define JBERR_INVALID_FILE	(JBERR_BASE 0x21) //文件错误

```
#define JBERR_ENCRYPT_IC_NO_FIND      (JBERR_BASE|0x22) //没有发现加密 IC  
  
#define JBERR_ENCRYPT_IC_NO_MATCH    (JBERR_BASE|0x23) //加密 IC 校验失败  
  
#define JBERR_RTSP_GET_DESCRIBE    (JBERR_BASE|0x24)   //获取 RTSP 描述失败
```

```

#define JBERR_RTSP_SETUPAUDIO (JBERR_BASE|0x25) //RTSP 设置音频失败
#define JBERR_RTSP_SETUPVIDEO (JBERR_BASE|0x26) //RTSP 设置视频失败
#define JBERR_RTSP_NOSTREAM (JBERR_BASE|0x27) //RTSP 没有流数据
#define JBERR_RTSP_PLAY (JBERR_BASE|0x28) //RTSP 播放启动失败
#define JBERR_IP_ERROR (JBERR_BASE|0x29) //本地 IP 错误，不能连接指定服务器
#define JBERR_SERVER_UPDATAING (JBERR_BASE|0x2A) //服务器正在升级，不接受客户端的连接操作

```

可以调用 `JBNV_GetErrorMessage` 获得错误的描述信息

## 4.2 结构定义

### 4.2.1 服务器基础信息

```

typedef struct tagJBNV_SERVER_INFO
{
    DWORD     dwSize;           //结构大小
    DWORD     dwServerFlag;     //服务器类型
    DWORD     dwServerIp;       //服务器 IP(整数表示形式)
    char      szServerIp[16];   //服务器 IP(字符串表示形式)
    WORD      wServerPort;      //服务器端口
    WORD      wChannelNum;     //通道数量
    DWORD     dwVersionNum;    //版本
    char      szServerName[32]; //服务器名称
    DWORD     dwServerCPUType;  //当前 CPU 类型
    BYTE      bServerSerial[48]; //服务器序列号，具有唯一标识功能
    BYTE      byMACAddr[6];     //服务器的物理地址
    DWORD     dwAlarmInCount;   //告警输入个数
}

```

```
DWORD    dwAlarmOutCount;      //告警输出个数  
DWORD    dwSysFlags;          //系统支持的功能  
DWORD    dwUserRight;         //当前用户权限  
DWORD    dwNetPriviewRight;   //网络观看权限  
char     csServerDes[64];     //服务器描述  
}JBNV_SERVER_INFO,*PJBNV_SERVER_INFO;
```

此结构通过调用 JBNV\_GetServerInfo 获得

变量说明：

A、dwServerFlag 服务器类型，有以下了类型：

```
#define JB_NVS_FLAG           100001 //网络视频编码器  
#define JB_NVD_FLAG           100002 //网络视频解码器  
#define JB_NVR_FLAG           100003 //网络视频编码器（带硬盘）
```

B、dwServerCPUType，服务器的 CPU 类型，有以下类型：

```
#define    CPUTYPE_8120        8120  
#define    CPUTYPE_3510        3510
```

C、dwSysFlags，服务器支持的功能，有以下定义：

```
#define  JB_SYS_FLAG_ENCODE_D1      0x00000001 //支持 D1  
#define  JB_SYS_FLAG_HD_RECORD    0x00000002 //支持 IDE 录像
```

```
#define JB_SYS_FLAG_CI_IDENTIFY 0x00000004 //支持智能识别  
  
#define JB_SYS_FLAG_MD_RECORD 0x00000008 //支持 MD 卡、U 盘录像
```

D、dwUserRight，服务器用户权限

```
#define USER_RIGHT_PTZ 0x0001 //用户能够控制云台  
  
#define USER_RIGHT_SETUP 0x0002 //能够进行服务器参数配置  
  
#define USER_RIGHT_UPDATA 0x0004 //能够对服务器进行升级和重启操作。
```

“Admin” 用户拥有所有权限，且权限不可剥夺。

E、dwNetPriviewRight，用户对通道进行预览的权限。使用与操作方式，如用户对通道 2 拥有观看权限，则 dwNetPriviewRight 值为 0x02 ;如果同时对通道 2 和通道 4 有观看权限，则值为：

0x02 | 0x08 = 0xA

#### 4.2.2 服务器消息结构

```
typedef struct tagJB_SERVER_MSG{  
  
    DWORD dwMsg; //消息类型代码  
    SYSTEMTIME st; //服务器时间  
    DWORD dwChannel; //通道  
    cbSize; //附加数据长度  
};JB_SERVER_MSG,*PJB_SERVER_MSG;
```

结构说明：此结构是服务器发送给客户端的消息结构，使用在

```

LONG CMainDialog::DefWindowProc (UINT message, WPARAM wParam, LPARAM lParam)

{
    if(message == WM_JB_MESSAGE)

    {
        HANDLE hServer = (HANDLE)wParam;

        JB_SERVER_MSG *lpJbMsg = (JB_SERVER_MSG *)lParam;

        .....

    }
}

```

由下面函数

JBNV\_SetMessage(hServer ,GetSafeHwnd(),WM\_JB\_MESSAGE);设置

变量说明:

A、 dwMsg; 消息类型,以下为具体定义:

#define JB_MSG_VIDEOLOST	0x20000001	//视频丢失
#define JB_MSG_MOVEDECTECT	0x20000002	//移动报警
#define JB_MSG_SENSOR_ALARM	0x20000003	//探头报警
#define JB_MSG_RESETSERVER	0x20000004	//服务器因为某些参数被改变, 将要重新启动
#define JB_MSG_JPEGSNAP	0x20000005	//JPEG 抓图
#define JB_MSG_UPGRADE	0x20000006	//升级进度通知
#define JB_MSG_CRCERROR	0x20000007	//升级 CRC 错
#define JB_MSG_SERVER_BUSY	0x20000008	//服务器忙, 设置失败
#define JB_MSG_SERVER_BREAK	0x20000009	//服务器中断连接
#define JB_MSG_CHANNEL_BREAK	0x2000000A	//通道中断连接

```
#define JB_MSG_TALK_REQUEST      0x2000000B //远程服务器请求对讲
```

```
#define JB_MSG_UPGRADEOK          0x2000000C //升级完成  
#define JB_MSG_VIDEORESUME        0x2000000D //视频恢复  
#define JB_MSG_COMDATA           0x2000000E //串口数据  
#define JB_MSG_USERDATA          0x2000000F //用户数据
```

收到 JB\_MSG\_JPEGSNAP 时候，处理方式如下：

```
CString str = "c:\\out.jpg";  
  
HANDLE hFile = CreateFile(str, GENERIC_READ | GENERIC_WRITE, FILE_SHARE_READ |  
FILE_SHARE_WRITE, NULL, CREATE_NEW, FILE_FLAG_SEQUENTIAL_SCAN, NULL);  
if(hFile != INVALID_HANDLE_VALUE) {  
    WriteFile(hFile,(char *)lpJbMsg + sizeof(JB_SERVER_MSG),  
             lpJbMsg->cbSize,&dwWritten,0);  
  
    CloseHandle(hFile);  
}
```

收到 JB\_MSG\_COMDATA 消息时,处理方式如下:

```
JB_COM_DATA *lpcomdata = (JB_COM_DATA *)((char *)lpJbMsg + sizeof(JB_SERVER_MSG));
```

### 4.2.3 服务器流信息

```
typedef struct tagJBNV_CHANNEL_INFO  
{  
    DWORD dwSize;  
    DWORD dwStream1Height; //视频高(1)  
    DWORD dwStream1Width; //视频宽
```

```

DWORD dwStream1CodecID; //视频编码类型号 (MPEG4 为 0, JPEG2000 为 1,H264 为 2)

DWORD dwStream2Height; //视频高(2)

DWORD dwStream2Width; //视频宽

DWORD dwStream2CodecID; //视频编码类型号 (MPEG4 为 0, JPEG2000 为 1,H264 为 2)

DWORD dwAudioChannels; //音频通道数

DWORD dwAudioBits; //音频比特率

DWORD dwAudioSamples; //音频采样率

DWORD dwAudioFormatTag; //音频编码类型号 (MP3 为 0x55, G722 为 0x65)

char csChannelName[32]; //通道名称

}JBNV_CHANNEL_INFO,*PJBNV_CHANNEL_INFO;

```

此结构通过调用 JBNV\_GetChannelInfo 获得

参数说明: dwAudioFormatTag,有以下可能值:

#define WAVE_FORMAT_ALAW	0x0006
#define WAVE_FORMAT_MPEGLAYER3	0x0055
#define WAVE_FORMAT_G722_ADPCM	0x0065
#define WAVE_FORMAT_G711A	0x003E
#define WAVE_FORMAT_ADPCM	0x0002

#### 4.2.4 服务器探头信息

```
typedef struct tagJBNV_SENSOR_INFO
```

```
{  
  
    DWORD      dwSize;  
    DWORD      dwIndex;           //探头索引（输入为 0, 1, ..., 15, 输出为 16, 17, ... 31）  
    DWORD      dwSensorType;     //探头类型  
    char       csSensorName[32]; //探头名称  
  
} JBNV_SENSOR_INFO, *PJBNV_SENSOR_INFO;
```

此结构通过调用 JBNV\_GetSensorInfo 获得

已知的探头输入种类列表：

//0x0001 红外对射探头

//0x0002 户外探头

//0x0003 气体探头

//0x0004 门磁探头

//0x0005 玻璃破碎感应

//0x0006 有线红外探头

//0x0007 脚踏开关

//0x0008 烟感探头

//0x0009 手按开关

//0x000A 音频监听探头

//输出种类：

//0x0001 自动拨号器

```
//0x0002 蜂鸣器
```

```
//0x0003 灯具开关
```

```
//0x0004 机械开关
```

## 4.2.5 服务器设置结构

以下结构均通过调用 JBNV\_SetServerConfig 或 JBNV\_GetServerConfig 进行操作

### 4.2.5.1 通道 OSD 设置结构

```
//通道 OSD 显示以及通道名称设置
```

```
typedef struct tagJB_CHANNEL OSDINFO
{
    DWORD      dwSize;
    DWORD      dwChannel;
    BOOL       bShowTime;           //是否显示时间
    DWORD      dwTimeFormat;
    /*
    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
    */
}
```

```

BOOL bShowString;           //是否显示用户输入字符串

DWORD   dwStringx;         //x, 字符显示位置, 以 D1 大小为准, 系统根据当前输出自动进行调整

DWORD   dwStringy;         //y

char    csString[48];       //用户输入字符串

char    csOsdFontName[32];  //OSD 叠加显示要使用的字体类型。

} JB_CHANNEL OSDINFO, *LPJB_CHANNEL OSDINFO;

```

#### 4.2.5.2 通道编码设置结构

```

typedef struct tagJB_CHANNEL_CONFIG

{
    DWORD   dwSize;

    DWORD   dwChannel;          //通道号

    //主码流

    DWORD   nFrameRate;         //帧率 (1~25/30) PAL 为 25, NTSC 为 30

    DWORD   nStandard;          //制式 (0 为 NTSC 制, 1 为 PAL 制)

    DWORD   dwRateType;          //流模式(0 为定码流, 1 为变码流)

    DWORD   dwStreamFormat;      //格式 (0 为 CIF, 1 为 D1, 2 为 HALF-D1, 3 为 QCIF)

    DWORD   dwBitRate;           //码率 (16000~4096000)

    DWORD   dwImageQuality;      //编码质量(0~4), 0 为最好

    DWORD   nMaxKeyInterval;     //关键帧间隔(1~100)

    BOOL   bEncodeAudio;         //是否编码音频

    char   csChannelName[32];    //通道名称

} JB_CHANNEL_CONFIG, *LPJB_CHANNEL_CONFIG;

```

#### 4.2.5.3 通道颜色设置结构

```
typedef struct tagJB_CHANNEL_COLOR

{
    DWORD    dwSize;
    DWORD    dwChannel;
    DWORD    dwSetFlag;        //0, 设置但不保存;1, 保存参数;2, 恢复上一次保存的
    DWORD    dwHue;           //色调
    DWORD    dwColor;          //色彩
    DWORD    dwContrast;       //对比度
    DWORD    dwBrightness;      //亮度
    DWORD    dwDefinition;     //清晰度

} JB_CHANNEL_COLOR, *PJB_CHANNEL_COLOR;
```

#### 4.2.5.4 通道遮挡区域设置设置结构

```
typedef struct tagJB_CHANNEL_SHELTER

{
    DWORD    dwSize;
```

```
    DWORD      dwChannel;

    BOOL bEnable;      //是否进行区域遮挡

    RECT rcShelter[5]; //遮挡区域，最多支持 5 块的区域遮挡。RECT 以 D1 为准

} JB_CHANNEL_SHELTER, *PJB_CHANNEL_SHELTER;
```

#### 4.2.5.5 通道告警相关结构结构

```
//时间结构

typedef struct tagJB_SET_TIME{

    WORD wHour;           //小时 (0-23)

    WORD wMinute;         //分钟 (0-59)

} JB_SET_TIME;
```

```
//布防时间段

typedef struct tagJB_SCHED_TIME

{

    BOOL bEnable; JB_SET_TIME      BeginTime1;          //是否启用该时间段

                                         //时间段 1

    JB_SET_TIME   EndTime1;

    JB_SET_TIME   BeginTime2;      //时间段 2
```

```
    JB_SET_TIME    EndTime2;

} JB_SCHED_TIME;

//所有告警和图象抓拍均自动上传到客户端，不许要进行设置。
```

#### 4.2.5.5.1 视频移动告警相关结构

```
typedef struct tagJB_CHANNEL_MOTION_DETECT{

    DWORD          dwSize;
    DWORD          dwChannel;           //通道号
    BOOL           bEnable;            //是否进行布防
    DWORD          nDuration;          //报警 IO 端口输出持续时间(秒)
    BOOL           bShotSnap;          //是否自动抓图
    BOOL           bAlarmI00ut[4];      //探头输出
    DWORD          dwSensitive;        //灵敏度
    JB_SCHED_TIME  jbScheduleTime[8];   //布防时间段，星期一到星期天、每日。
    BYTE          *pbMotionArea[44*36]; //布防区域块组. 布防区域共有 44*36 个块, 每一个块分别用一个

} JB_CHANNEL_MOTION_DETECT, *PJB_CHANNEL_MOTION_DETECT;
```

#### 4.2.5.5.2 探头告警相关结构

```
typedef struct tagJB_SENSOR_ALARM

{

    DWORD          dwSize;
    int            nSensorID;          //探头索引
```

```

    DWORD          dwSensorType;           //探头类型
    char           szSensorName[32];        //探头名称
    BOOL           bEnable;                //是否进行布防
    int            nDuration;              //报警 IO 端口输出持续时间(秒)
    BOOL           bAlarmI00ut[4];         //探头联动输出
    BOOL           bPresetChannel[4];      //当发生告警时，是否进行云台预置点操作
    JB_SCHED_TIME jbScheduleTime[8];       //布防时间
    BOOL           bSnapShot;              //发生告警时，是否抓图
    int            nCaptureChannel;        //报警时，进行抓图动作的通道
    BOOL           bReQuestTalkback;       //是否请求对讲

} JB_SENSOR_ALARM, *PJB_SENSOR_ALARM;

```

#### 4.2.5.5.3 视频丢失告警相关结构

```

typedef struct tagJB_CHANNEL_VIDEO_LOST
{
    DWORD          dwSize;
    DWORD          dwChannel;             //通道号
    BOOL           bEnable;                //是否进行布防
    int            nDuration;              //报警 IO 端口输出持续时间(秒)
    BOOL           bAlarmI00ut[4];         //报警 IO 输出
    JB_SCHED_TIME jbScheduleTime[8];       //布防时间段

} JB_CHANNEL_VIDEO_LOST, *PJB_CHANNEL_VIDEO_LOST;

```

#### 4.2.5.6 探头输出相关结构

```

typedef struct tagJBNV_ALARM_OUT
{

```

```
    DWORD dwAlarmOutType; char
    szAlarmOutName[32];
                                //告警输出类型

                                //告警输出名称

} JBNV_ALARM_OUT, *PJBNV_ALARM_OUT;
```

不同的服务器探头数量不同。

```
#pragma warning( disable :4200 )

typedef struct tagJBNV_ALARM_OUT_INFO{

    DWORD          dwSize;

    JBNV_ALARM_OUT      jbao[];
                                //告警输出，根据探头输出个数决定设置多少

} JBNV_ALARM_OUT_INFO, *PJBNV_ALARM_OUT_INFO;

#pragma warning( default :4200 )
```

#### 4.2.5.7 COM 及云台设置结构

```
typedef struct tagJB_SERVER_COMINFO

{

    DWORD          dwSize;

    DWORD          dwChannel;           //通道号
```

```

    DWORD      dwBaudRate;           //波特率
    /*1200, 2400, 4800, 9600, 19200, 38400, 43000, 56000, 57600, 115200*/
    int       nDataBits;            //数据位
    int       nStopBits;             //停止位
    int       nParity;              //校验位
    int       nStreamControl;        //流控制
    int       nPrePos;               //云台预置位
    int       nCruise;                //云台巡航
    int       nTrack;                 //云台轨迹
    int       nPTZSpeed;              //云台速度
    int       nAddress;                //云台地址
    char      szPTZName[32];          //云台协议名称
}

} JB_SERVER_COMINFO, *PJB_SERVER_COMINFO;

```

当设置后，服务器在需要时会自动调用云台执行相应的设置云台预置位操作。

#### 4.2.5.8 透明串口设置结构

```

typedef struct tagJB_SERVER_COM2INFO
{
    DWORD      dwSize;
    DWORD      dwChannel;           //通道号
    DWORD      dwBaudRate;          //波特率
    int       nDataBits;            //数据位
    int       nStopBits;             //停止位
    int       nParity;              //校验位
    int       nStreamControl;        //流控制
    BOOL      bTransferState;        //透明传输

}

} JB_SERVER_COM2INFO, *PJB_SERVER_COM2INFO;

```

//云台协议

```
typedef struct tagJB_PTZ_PROTOCOL{  
  
    DWORD      dwSize;  
  
    DWORD      dwChannel;  
  
    char       szPTZName[32];           //协议名称  
    char       szPTZData[4 * 1024];     //协议数据  
  
} JB_PTZ_PROTOCOL, *PJB_PTZ_PROTOCOL;
```

```
#pragma warning( disable :4200 )
```

可以使用下面的结构直接通过 COM 口收发数据

```
typedef struct tagJB_COM_DATA  
{  
  
    DWORD      dwSize;  
  
    DWORD      dwChannel;  
    BOOL       bIs485;                //485 or 232  
    char       DataBuf[];             //最大 128
```

```
 } JB_COM_DATA;  
  
#pragma warning( default:4200 )
```

#### 4.2.5.9 网络设置结构

```
typedef struct tagJB_SERVER_NETWORK  
{  
    DWORD      dwSize;  
    DWORD      dwNetIpAddr;           //IP 地址  
    DWORD      dwNetMask;            //掩码  
    DWORD      dwGateway;            //网关  
    DWORD      dwMultiAddr;          //多播地址  
    DWORD      dwHttpPort;           //Http 端口  
    DWORD      dwDataPort;            //数据端口  
    DWORD      dwDNSServer;          //DNS 服务器  
    DWORD      dwTalkBackIp;          //服务器告警时，自动连接的对讲 IP  
    char       szMacAddr[6];          //网卡 MAC 地址  
    char       szServerName[32];        //服务器名称  
}; JB_SERVER_NETWORK,*PJB_SERVER_NETWORK;
```

#### 4.2.5.10 用户设置结构

```
typedef struct tagJB_SERVER_USER{  
    DWORD      dwSize;  
    DWORD      dwIndex;             //最多 5 个用户，第一个默认名称为 Admin.  
    BOOL       bEnable;              //是否启用  
    char       csUserName[32];  
    char       csPassword[32];  
    DWORD      dwUserRight;          /* 权限 */  
                           //0x01: 控制云台  
                           //0x02: 设置参数
```

```
//0x04: 升级, 格式化  
DWORD      dwNetPreviewRight;           //远程可以预览的通道 bit0 -- channel 1  
DWORD      dwUserIP;                  //用户 IP 地址(为 0 时表示允许任何地址)  
BYTE       byMACAddr[6];              //物理地址限制, 如果不填写, 则表示允许任何物理地址登入  
  
} JB_SERVER_USER, *PJB_SERVER_USER;
```

#### 4.2.5.11 在线通知设置结构

```
typedef struct tagJB_NOTIFY_SERVER  
{  
  
    DWORD      dwSize;  
    BOOL       bEnable;                //服务器是否定时发送在线消息给目标主机  
    DWORD      dwTimeDelay;            //服务器发送在线消息的间隔时间  
    DWORD      dwPort;                 //接收服务器在线消息的目标主机端口  
    char       szDNS[128];              //接收服务器在线消息的目标主机域名
```

```
} JB_NOTIFY_SERVER, *PJB_NOTIFY_SERVER;
```

#### 4.2.5.11 PPPOE 和 DDNS 设置结构

```
typedef struct tagJB_PPPOE_DDNS_CONFIG  
{  
    DWORD    dwSize;  
    BOOL     bEnablePPPOE;           //0-不启用, 1-启用  
    char     csPPPoEUserName[32];   //PPPoE 用户名  
    char     csPPPoEPASSWORD[32];   //PPPoE 密码  
    DWORD    dwPPPoEIP;            //PPPoE IP 地址(只读)  
    BOOL     bEnableDDNS;          //是否启用 DDNS  
    char     csDDNSName[64];        //在服务器注册的 DDNS 名称  
    char     csDNSAddress[64];      //DNS 服务器地址  
    DWORD    dwDNSPort;            //DNS 服务器端口, 默认为 6500  
}; JB_PPPOE_DDNS_CONFIG, *PJB_PPPOE_DDNS_CONFIG;
```

#### 4.2.5.12 告警输出设置结构

```
#define  SENSOR_OUT_1      0x0001    //探头 1 有输出  
#define  SENSOR_OUT_2      0x0002    //探头 2 有输出  
#define  SENSOR_OUT_3      0x0004    //探头 3 有输出  
#define  SENSOR_OUT_4      0x0008    //探头 4 有输出  
  
typedef struct tagJBNV_SENSOR_STATE{  
    DWORD    dwSize;  
    DWORD    dwSensorID;          //探头 ID  
    DWORD    dwSensorOut;         //探头输出状态  
}; JBNV_SENSOR_STATE;
```

#### 4.2.5.13 人脸信息结构

```

typedef struct tagDMS_FR_SAMPLE_S{

    DWORD      dwSize;           // 结构体大小
    DWORD      id;              //数据库唯一 ID
    char       uuid[36];         //UUID
    char       name[32];         //姓名
    int        age;             //年龄
    char       sex;              // 性别 ‘F’ / ‘M’
    char       role;             // 0, 所有, 1-白名单 2-黑名单, 3, VIP, 4, 陌生人
    char       identity_num[32]; // 身份证号 18 位长
    char       csOther[64];       // 附加信息
    char       Reserved[192];    //保留
    unsigned int feature_len;   // 特征值长度
    unsigned int data_len;       // 人脸图片数据长度

} DMS_FR_SAMPLE_S

```

#### 4.2.5.14 查询条件结构

```

typedef struct tagDMS_FR_SAMPLE_QUERY_COND_S{

    int        struct_size;     // 结构体大小
    int        total_size;       //数据库总的记录数
    int        query_size;       //当前查询结果的记录数
    sample[DMS_FR_QUERY_MAX_NUM]
DMS_FR_SAMPLE_S;                                // 不含图片数据、特征值

} DMS_FR_SAMPLE_QUERY_COND_S

```

#### 4.2.5.15 查询结果结构

```

typedef struct {
    int        struct_size;     // 结构体大小

```

```
    int          page_size;           //每页显示多少条记录
    int          page_index;          //页码 0,1,2,3
    int          role;                // 样本类型 0-所有, 其他同 DMS_FR_SAMPLE_S 的 role
} DMS_FR_SAMPLE_QUERY_RES_S
```

## 4.3 函数定义

所有函数返回值均为错误码定义，如果返回不为 0，请参考错误码定义。

### 4.3.1 服务器操作函数

```
DWORD WINAPI JBNV_OpenServer(LPCSTR lpServerDNS,WORD wServerPort,
                                LPCSTR lpszUserName,LPCSTR lpszPassword,LPVOID *lpHandle);
```

函数功能：打开服务器

参数说明：lpServerDNS 服务器地址，可以是 IP，也可以是 DNS 地址，不过本地电脑必须设置 DNS 服务器地址。

wServerPort 服务器端口，默认为 9200

lpszUserName 用户名

lpszPassword 密码

lpHandle 当打开成功后，会返回一个句柄给应用程序。

```
DWORD WINAPI JBNV_CloseServer(HANDLE hServer);
```

函数功能：关闭服务器

```
DWORD WINAPI JBNV_GetServerInfo(HANDLE hServer,JBNV_SERVER_INFO *lpsi);
```

函数功能：获取服务器信息

```
DWORD WINAPI JBNV_SetMessage(HANDLE hServer,HWND hMsgWnd,UINT nMessage);
```

函数功能：设置消息回调,设置后具体现象请参看 [JB\\_SERVER\\_MSG](#) 说明

```
DWORD WINAPI JBNV_SetStreamCallBack(HANDLE hServer,StreamCallBack lpscb,DWORD dwUser);
```

函数功能：设置数据流回调

StreamCallBack 函数定义如下示例：

```
static int CALLBACK JBNVRecvStreamDataCallback(  
    HANDLE hServer,  
    DWORD dwClientIndex,           //客户窗口索引，客户自定义//  
    WORD dwChannelNo,             //服务器通道号，从 0 开始//结构  
    PBYTE pbDataBuffer,  
    DWORD dwUser);
```

```
DWORD WINAPI JBNV_SetSecondSurface(HANDLE hServer,DWORD dwChannelNo,HWND hDispalyWnd,
```

```
DrawOSD pOSD,WORD dwUser);
```

函数功能：设置数据流的第二个显示表面

DrawOSD 函数定义如下示例：

```
static void CALLBACK JBNVOSDCallback(  
    HDC hDC,                      //hDC,此 DC 不能在回调里调用 ReleaseDC 释放  
    SIZE *pImageSize,              //图象区域大小，视频大小不同而不同  
    WORD dwUser);                 //用户数据
```

```
DWORD WINAPI JBNV_SetServerConfig(HANDLE hServer,WORD dwCmd,VOID *lpData,  
WORD dwSize);
```

函数功能：配置服务器参数

参数说明：

dwCmd 设置和获取的命令字，参考 4.4.1 章节的说明  
lpData 设置或获取的数据，可根据要求填写 dwSize 结构或数据的大小。

```
DWORD WINAPI JBNV_GetServerConfig(HANDLE hServer,WORD dwCmd,VOID *lpData,  
WORD dwSize);
```

函数功能：获取服务器参数

参数说明：

dwCmd 设置和获取的命令字，参考 4.4.2 章节的说明  
lpData 设置或获取的数据，可根据要求填写 dwSize 结构或数据的大小。

```
DWORD WINAPI JBNV_GetChannelInfo(HANDLE hServer,WORD dwChannelNo,  
JBNV_CHANNEL_INFO *lpStreamInfo);
```

函数功能：得到流信息

```
DWORD WINAPI JBNV_GetSensorInfo(HANDLE hServer,WORD dwSensorID,  
JBNV_SENSOR_INFO *lpSensor);
```

函数功能：得到探头信息

```
DWORD WINAPI JBNV_GetServerStatus(HANDLE hServer,JB_SERVER_STATE *lpServerState);
```

函数功能：得到服务器状态信息

```
DWORD WINAPI JBNV_UpdateServer(HANDLE hServer,LPCSTR lpszFileName);
```

函数功能：使用指定文件对服务器进行升级

## 4.3.2 通道操作函数

### 4.3.2.1 通道打开关闭函数

```
DWORD WINAPI JBNV_OpenChannel(HANDLE hServer,  
                                DWORD dwChannelNo,  
                                DWORD dwClientIndex,  
                                DWORD dwNetProtocol,  
                                DWORD dwStreamType,  
                                HWND hDispWnd,           //显示窗口  
                                BOOL bUserDecode,  
                                DWORD dwShowMode,  
                                LPVOID *lpHandle);
```

函数功能：打开服务器通道

参数说明：

dwChannelNo， 服务器通道

dwClientIndex 客户窗口索引， 客户自定义， 此参数在 SDK 内部仅保存， 不起任何作用， 只在

StreamCallBack 中使用， 具体请参看 **JBNV\_SetStreamCallBack** dwNetProtocol，  
网络协议， 0 为 TCP， 1 为 UDP， 2 为多播 dwStreamType， 码流类型， 目前仅支持大  
码流。0 为大码流， 1 为小码流 bUserDecode， 是否用户自己进行解码操作， 如果是，

SDK 将不进行解码显示操作。 dwShowMode 显示模式，0 为 Overlay 显示，1 为 RGB 显示。

lpHandle 如果打开通道成功，lpHandle 为返回的通道句柄  
DWORD WINAPI **JBNV\_CloseChannel**(HANDLE hChannel);

函数功能：关闭通道显示

DWORD WINAPI **JBNV\_GetRecvCount**(HANDLE hChannel, DWORD \*pdwRecvCount);

函数功能：得到此通道的网络数据接受量

参数说明：

pdwRecvCount 接收数据的指针，当函数成功执行，数据将被正确填写。

DWORD WINAPI **JBNV\_Mute**(HANDLE hChannel, BOOL bMute);

函数功能：打开关闭声音播放

### 4.3.2.2 通道抓图

DWORD WINAPI **JBNV\_CaptureBitmap**(HANDLE hChannel, BYTE \*in\_Buffer, DWORD \*nSize); 函数功能：本地抓取 BMP 图片，远程抓图请参看 4.4.1 章节的 CMD\_SNAPSHOT.

说明：

如果 in\_Buffer 输入为 NULL，则 nSize 返回存储 BITMAP 图片所需的空间  
成功执行，in\_Buffer 返回 BMP 数据，直接保存为 BMP 文件即可

### 4.3.2.3 通道缓冲配置

DWORD WINAPI **JBNV\_GetStreamBufferSize**(HANDLE hChannel, DWORD \*lpMin,  
DWORD \*lpMax, DWORD \*lpCurrent);

函数功能：获取播放缓冲信息

参数说明：

lpMin 最小缓冲区设置

lpMax 最大缓冲区设置

lpCurrent 当前缓冲数量

DWORD WINAPI **JBNV\_SetStreamBufferSize**(HANDLE hChannel, DWORD dwMin, DWORD dwMax);

函数功能：设置播放缓冲，如果 dwMin 为 0，视频将采用收到即播的策略。

如果当前缓冲数量大于 dwMax，系统自动抛弃多余的数据，

建议 dwMax 大于或等于 200。

#### 4.3.2.4 通道显示配置

DWORD WINAPI **JBNV\_SetDisplayRect**(HANDLE hChannel, RECT \*lpRect);

函数功能：设置显示区域，可以用于显示局部放大

lpRect 指的是视频区域的举行，比如要显示图象的左上角上一个区域，则 Rect(0,0,30,20) 最大不能超过一个 D1 的大小

DWORD WINAPI **JBNV\_UpdataImage**(HANDLE hChannel, RECT \*lpRect);

函数功能：更新显示区域

如果 lpRect 为空，则更新默认显示区域。

DWORD WINAPI **JBNV\_SetOSDCallback**(HANDLE hChannel, DrawOSD pOSD, DWORD dwUser);

函数功能：设置 OSD 回调显示

dwUser 为用户数据，将在 DrawOSD 回调给用户使用

DWORD WINAPI **JBNV\_SetDisplayhWnd**(HANDLE hChannel, Hwnd hDisplayWnd);

函数功能：设置流的显示窗口，可以更改当前视频的显示窗口

```
DWORD WINAPI JBNV_SetVideoCallback(HANDLE hChannel, VideoCallBack pvideocallback,  
DWORD dwUser);
```

函数功能：设置视频数据回调，此回调发生在解码后显示前

```
DWORD WINAPI JBNV_SetVideoStatic(HANDLE hChannel, BOOL bShowVideo);
```

函数功能：设置视频是否显示

#### 4.3.2.2 录像函数

```
DWORD WINAPI JBNV_SetPreRecord(HANDLE hChannel, DWORD dwSeconds);
```

函数功能：设置预录时间长度，默认为 0 秒

```
DWORD WINAPI JBNV_StartRecord(HANDLE hChannel, LPCSTR lpszFile, DWORD dwFileType);
```

函数功能：开始录像，当有预录数据时，首先把预录数据写入录像文件

```
DWORD WINAPI JBNV_GetRecorded(HANDLE hChannel, DWORD *lpdwWrited);
```

函数功能：获取已经录像写盘的数据大小

```
DWORD WINAPI JBNV_StopRecord(HANDLE hChannel);
```

函数功能：停止录像，当关闭通道时，自动停止录像

#### 4.3.3 对讲操作函数

```
DWORD WINAPI JBNV_TalkOpen(LPCSTR lpszServerIP, WORD wServerPort);
```

函数功能：连接服务器，开始对讲

```
DWORD WINAPI JBNV_TalkListen(LPCSTR lpServer, WORD nServerPort);
```

函数功能：设置监听服务器，如果连接多台服务器时，此函数起作用

```
DWORD WINAPI JBNV_TalkClose();
```

函数功能：关闭对讲。自动关闭已经存在的所有对讲连接

```
DWORD WINAPI JBNV_StartVoiceCom(LPCSTR lpszServerIP, WORD wServerPort,  
                                  VoiceDataCallBack lpVideoData, DWORD dwUser, LONG *lpHandle);
```

函数功能：开始语音对讲透传

```
DWORD WINAPI JBNV_StopVoiceCom();
```

函数功能：结束语音对讲透传

```
DWORD WINAPI JBNV_VoiceComSendData (LONG lVoiceComHandle, int nVoiceType,  
                                      char *pSendBuf, DWORD dwBufSize);
```

函数功能：开始透明传输后发送数据到设备端

参数说明：

LONG lVoiceComHandle	<b>JBNV_StartVoiceCom</b> lpHandle 的值
int nVoiceType	如下定义
#define NETTALK_FORMAT_WAV	1
#define NETTALK_FORMAT_GMS6102	
#define NETTALK_FORMAT_722	3
#define NETTALK_FORMAT_711	4 //压缩后的 G.711 音频数据
#define NETTALK_FORMAT_ADPCM	5 //未压缩原始数据

#### 4.3.4 云台操作函数

仅在服务器设置有云台协议时，函数调用才能正确执行。

```
DWORD WINAPI JBNV_PTZControl(HANDLE hServer, DWORD dwChannelNo, DWORD dwPTZCmd,
```

```
DWORD dwValue);
```

函数功能：云台控制

参数说明：dwPTZCmd 有以下定义：

#define PTZ_CMD_UP	1
#define PTZ_CMD_DOWN	2
#define PTZ_CMD_LEFT	3
#define PTZ_CMD_RIGHT	4
#define PTZ_CMD_FOCUS_SUB	5
#define PTZ_CMD_FOCUS_ADD	6
#define PTZ_CMD_ZOOM_SUB	7
#define PTZ_CMD_ZOOM_ADD	8
#define PTZ_CMD_IRIS_SUB	9
#define PTZ_CMD_IRIS_ADD	10
#define PTZ_CMD_STOP	11
#define PTZ_CMD_PRESET	12 //预置
#define PTZ_CMD_CALL	13 //调用
#define PTZ_CMD_AUTO_STRAT	14 //自动
#define PTZ_CMD_AUTO_STOP	15
#define PTZ_CMD_LIGHT_OPEN	16 //灯光
#define PTZ_CMD_LIGHT_CLOSE	17
#define PTZ_CMD_BRUSH_START	18 //雨刷

```
#define PTZ_CMD_BRUSH_STOP      19
#define PTZ_CMD_TRACK_START      20      //轨迹
#define PTZ_CMD_TRACK_STOP       21
#define PTZ_CMD_TRACK_RUN        22
```

#### 4.3.5 主机搜索注册函数

搜索主机

```
typedef struct tagJBNV_SERVER_PACK
{
    char     szIp[16];           //服务器 Ip
    WORD    wMediaPort;         //流端口
    WORD    wWebPort;           //Http 端口号
    WORD    wChannelCount;      //通道数量
    char     szServerName[32];   //服务器名
    DWORD   dwDeviceType;       //服务器类型
    DWORD   dwServerVersion;    //服务器版本
    WORD    wChannelStatic;     //通道状态(是否视频丢失)
    WORD    wSensorStatic;      //探头状态
    WORD    wAlarmOutStatic;    //报警输出状态
}JBNV_SERVER_PACK;
```

```
typedef struct tagJBNV_SERVER_MSG_DATA
{
    DWORD          dwSize;
    DWORD          dwPackFlag;
    JBNV_SERVER_PACK    jbServerPack;
```

```
JB_SERVER_MSG           jbMsg;  
}JBNV_SERVER_MSG_DATA;
```

函数原形:

```
DWORD WINAPI JBNV_SearchServer(HWND hWnd,DWORD dwMsg);
```

当使用 JBNV\_SearchServer， SDK 将会给系统发送指定消息，如果网络上存在对应视频服务器。

下面是示例:

```
JBNV_SearchServer(GetSafeHwnd(),WM_JBNV_SEARCHSERVER);  
  
case WM_JBNV_SEARCHSERVER:  
{  
    JBNV_SERVER_MSG_DATA *lpMsg = (JBNV_SERVER_MSG_DATA *)wParam;  
  
    if(lpMsg)  
    {  
        DVR_INFO dvrinfo;  
  
        memset(&dvrinfo,0,sizeof(DVR_INFO));  
  
        strcpy(dvrinfo.csServerAddress,lpMsg->jbServerPack.szIp);  
  
        strcpy(dvrinfo.csServerName,lpMsg->jbServerPack.szServerName);  
    }  
}
```

```
dvrinfo.dwChannelNumber = lpMsg->jbServerPack.wChannelCount;  
dvrinfo.wPort = lpMsg->jbServerPack.wMediaPort;  
dvrinfo.dwDeviceType = JB_NVS_DEV;  
pNetDlg->AddNetNode(&dvrinfo, FALSE);  
  
}  
  
}  
  
break;
```

### 4.3.6 其他辅助函数

函数功能：获取默认的错误描述信息

支持的语言：

```
#define LANGID_EN_US          MAKELANGID(LANG_ENGLISH,SUBLANG_DEFAULT)  
#define LANGID_CH           0x00000804
```

### 获得升级文件信息

```
DWORD WINAPI JBNV_GetUpgradeFileInfo(HANDLE hServer,  
                                     LPCSTR lpFile,  
                                     LPSTR lpDes);  
  
DWORD *lpFileVer,
```

```
//文件版本 //文件描述（需要 256 字符的 Buffer）  
//文件名 //文件长度
```

函数功能：获得升级文件的描述信息，并判断是否适合当前服务器

BOOL WINAPI **CheckIpIsInternet**(DWORD dwIp)

如果是互连网 IP，则返回 TRUE

DWORD WINAPI **GetDataCRC**(LPVOID lpData,DWORD dwSize)

获得指定长度数据的 CRC32 数值

DWORD WINAPI **GetWindowsMacInfo**(BYTE \*lpMac,DWORD \*lpdwSize)

获得当前计算机的 MAC 地址，仅支持 WIN2000 以上系统。

BOOL WINAPI **MakeWaterMark**(LPCSTR csFile)

给指定文件加上水印，如果成功则返回 TRUE

BOOL WINAPI **CheckWaterMark**(LPCSTR csFile)

检查指定文件上的水印，如果成功则返回 TRUE

## 4.4 服务器配置结构

具体结构信息可以参看 **JBNVSDK.h** 文件内的说明

### 4.4.1 设置服务器信息

```
#define CMD_REBOOT 0x00000001 //附加参数：无  
#define CMD_RESTORE 0x00000002 //附加参数：无  
#define CMD_UPDATEFLASH 0x00000003 //附加参数：无  
#define CMD_SNAPSHOT 0x00000004 //附加参数：int
```

```

#define CMD_SETSYSTIME 0x00000005 //附加参数: SYSTEMTIME
#define CMD_SET OSDINFO 0x00000006 //附加参数: JB_CHANNEL_OSDINFO
#define CMD_SET_SHELTER 0x00000007 //附加参数: JB_CHANNEL_SHELTER
#define CMD_SET_LOGO 0x00000008 //附加参数: JB_CHANNEL_LOGO 暂不支持
#define CMD_SET_CHANNEL_CONFIG 0x00000009 //附加参数: JB_CHANNEL_CONFIG
#define CMD_SET_COLOR 0x0000000A//附加参数: JB_CHANNEL_COLOR
#define CMD_SET_MOTION_DETECT 0x0000000B//附加参数: JB_CHANNEL_MOTION_DETECT
#define CMD_SET_SENSOR_ALARM 0x0000000C//附加参数: JB_SENSOR_ALARM
#define CMD_SET_VIDEO_LOST 0x0000000D//附加参数: JB_CHANNEL_VIDEO_LOST
#define CMD_SET_COMINFO 0x0000000E//附加参数: JB_SERVER_COMINFO
#define CMD_SET_USERINFO 0x0000000F//附加参数: JB_SERVER_USER
#define CMD_SET_NETWORK 0x00000010 //附加参数: JB_SERVER_NETWORK
#define CMD_UPLOAD_PTZ_PROTOCOL 0x00000011 //附加参数: JB_PTZ_PROTOCOL
#define CMD_SEND_COMDATA 0x00000012 //附加参数: JB_COM_DATA
#define CMD_SET_FTPUPDATA_PARAM 0x00000013 //附加参数: JB_FTPUPDATA_PARAM 暂不支持
#define CMD_CLEAR_ALARM_OUT 0x00000014 //附加参数: 无
#define CMD_SET_SERVER_COM2 0x00000015 //附加参数: JB_SERVER_COM2INFO
#define CMD_SET_ALARM_OUT 0x00000016 //附加参数: JBNV_ALARM_OUT_INFO
#define CMD_SET_NOTIFY_SERVER 0x00000017 //附加参数: JB_NOTIFY_SERVER
#define CMD_SET_PPPOE_DDNS 0x00000018 //附加参数: JB_PPPOE_DDNS_CONFIG
#define CMD_SET_SENSOR_STATE 0x00000019 //附加参数: JBNV_SENSOR_STATE


---


//Add 2007-08-30
#define CMD_SET_SERVER_RECORD 0x00000020 //附加参数: JB_SERVER_RECORD_SET
#define CMD_RECORD_BEGIN 0x00000021 //附加参数: 无
#define CMD_RECORD_STOP 0x00000022 //附加参数: 无

```

## 各命令含义

### **4.4.1.1**      CMD\_REBOOT

重新启动服务器

### **4.4.1.2**      CMD\_RESTORE

服务器恢复默认参数

### **4.4.1.3**      CMD\_UPDATEFLASH

更新 FLASH，把所做的设置保存到服务器的 FLASH 内，否则重新启动服务器后，修改会被取消

### **4.4.1.4**      CMD\_SNAPSHOT

服务器进行抓图操作，附加参数为要抓图的通道，lpData 必须为指向通道的指针。

### **4.4.1.5**      CMD\_SETSYSTIME

设置服务器时间，使用结构为 SYSTEMTIME

### **4.4.1.6**      CMD\_SET OSDINFO

设置通道 OSD 显示信息

### **4.4.1.7**      CMD\_SET\_SHELTER

设置通道遮挡

### **4.4.1.8**      CMD\_SET\_LOGO

设置通道 LOGO，本版本暂不支持

### **4.4.1.9**      CMD\_SET\_CHANNEL\_CONFIG

设置服务器编码信息

**4.4.1.10** CMD\_SET\_COLOR

设置通道的色彩信息

**4.4.1.11** CMD\_SET\_MOTION\_DETECT

设置通道移动告警检测

**4.4.1.12** CMD\_SET\_SENSOR\_ALARM

设置探头告警检查

**4.4.1.13** CMD\_SET\_VIDEO\_LOST

设置视频丢失检查

**4.4.1.14** CMD\_SET\_COMINFO

设置服务器的 COM 信息以及云台协议。

云台协议必须保存在系统的 \Program Files\Common Files\PTZ\ 目录下，后缀为 cod

**4.4.1.15** CMD\_SET\_USERINFO

设置用户密码和权限

**4.4.1.16** CMD\_SET\_NETWORK

设置服务器的网络配置

**4.4.1.17** CMD\_UPLOAD\_PTZ\_PROTOCOL

更新服务器上保存的云台协议。当调用 CMD\_SET\_COMINFO 时，SDK 自动查找\Program Files\Common Files\PTZ\ 目录的 COD 文件，然后执行此命令更新服务器云台协议。

**4.4.1.18** CMD\_SEND\_COMDATA 发送 COM 数据到服务器，服务器按照指定的串口通信频率发送此数据到 485 或 232

**4.4.1.19** CMD\_SET\_FTPUPDATA\_PARAM

设置服务器 FTP 上传报警抓拍图片。 (暂不支持)

**4.4.1.20** CMD\_CLEAR\_ALARM\_OUT

清除服务器上的告警输出

**4.4.1.21** CMD\_SET\_SERVER\_COM2

设置服务器的 COM2 信息

**4.4.1.22** CMD\_SET\_ALARM\_OUT

设置服务器告警输出信息

**4.4.1.23** CMD\_SET\_NOTIFY\_SERVER

设置服务器在线通知，自动向指定服务器指定端口发送 UDP 数据

**4.4.1.24** CMD\_SET\_PPPOE\_DDNS

设置服务器的 PPPOE，DDNS 配置。

**4.4.1.25** CMD\_SET\_SENSOR\_STATE

设置服务器的探头输出状态

**4.4.1.26** CMD\_SET\_SERVER\_RECORD

设置服务器的 MD 卡录像配置

**4.4.1.27** CMD\_RECORD\_BEGIN

服务器开始录像

**4.4.1.28** CMD\_RECORD\_STOP

服务器停止录像

## 4.4.2 人脸库管理命令

```
#define     CMD_FR_ADD_SAMPLE      0xA0100050 //附加参数: DMS_FR_SAMPLE_S  
#define     CMD_FR_DEL_SAMPLE      0xA0100051 //附加参数: DMS_FR_SAMPLE_S  
#define     CMD_FR_UPDATE_SAMPLE   0xA0100052 //附加参数: DMS_FR_SAMPLE_S  
#define     CMD_FR_QUERY_SAMPLE    0xA0100053 //附加参数: DMS_FR_SAMPLE_QUERY_COND_S  
#define     CMD_FR_REQ_SAMPLE_PIC  0xA0100054 //附加参数: DMS_FR_SAMPLE_S
```

### 各命令含义

#### 4.4.2.1 CMD\_FR\_ADD\_SAMPLE

新增样本

#### 4.4.2.2 CMD\_FR\_DEL\_SAMPLE

删除样本

#### 4.4.2.3 CMD\_FR\_UPDATE\_SAMPLE

更新样本

#### 4.4.2.4 CMD\_FR\_QUERY\_SAMPLE

按条件查询样本，请求 DMS\_FR\_SAMPLE\_QUERY\_COND\_S，返回 DMS\_FR\_SAMPLE\_QUERY\_RES\_S

#### 4.4.2.4 CMD\_FR\_REQ\_SAMPLE\_PIC

请求指定 ID 的图片数据

#### **4.4.3** 获取服务器信息

```

#define CMD_GET_SYSTIME 0x10000000 //附加参数: SYSTEMTIME
#define CMD_GET_OSDINFO 0x10000001 //附加参数: JB_CHANNEL_OSDINFO
#define CMD_GET_CHANNEL_CONFIG 0x10000002 //附加参数: JB_CHANNEL_CONFIG
#define CMD_GET_COLOR 0x10000003 //附加参数: JB_CHANNEL_COLOR
#define CMD_GET_MOTION_DETECT 0x10000004 //附加参数: JB_CHANNEL_MOTION_DETECT
#define CMD_GET_SENSOR_ALARM 0x10000005 //附加参数: JB_SENSOR_ALARM
#define CMD_GET_VIDEO_LOST 0x10000006 //附加参数: JB_CHANNEL_VIDEO_LOST
#define CMD_GET_COMINFO 0x10000007 //附加参数: JB_SERVER_COMINFO
#define CMD_GET_USERINFO 0x10000008 //附加参数: JB_SERVER_USER
#define CMD_GET_NETWORK 0x10000009 //附加参数: JB_SERVER_NETWORK
#define CMD_GET_FTPUPDATA_PARAM 0x1000000A //附加参数: JB_FTPUPDATA_PARAM
#define CMD_GET_PTZDATA 0x1000000B //附加参数: JB_PTZ_PROTOCOL
#define CMD_GET_SERVER_COM2 0x1000000C //附加参数: JB_SERVER_COM2INFO
#define CMD_GET_ALARM_OUT 0x1000000D //附加参数: JBNV_ALARM_OUT_INFO
#define CMD_GET_NOTIFY_SERVER 0x1000000E //附加参数: JB_NOTIFY_SERVER
#define CMD_GET_PPPOE_DDNS 0x1000000F //附加参数: JB_PPPOE_DDNS_CONFIG
#define CMD_GET_SENSOR_STATE 0x10000010 //附加参数: JBNV_SENSOR_STATE

//Add 2007-08-30
#define CMD_GET_SERVER_RECORD_SET 0x10000011 //附加参数: JB_SERVER_RECORD_SET
#define CMD_GET_SERVER_RECORD_STATE 0x10000012 //附加参数: JB_SERVER_RECORD_STATE
#define CMD_UNLOAD_DISK 0x10000013 //附加参数: 无

```

## 各命令含义

### 4.4.3.1 CMD\_GET\_SYSTIME

获取服务器的当前时间

**4.4.3.2**      CMD\_GET OSDINFO

获取服务器的 OSD 配置

**4.4.3.3**      CMD\_GET CHANNEL\_CONFIG

获取服务器编码配置

**4.4.3.4**      CMD\_GET COLOR

获取服务器的颜色配置

**4.4.3.5**      CMD\_GET MOTION\_DETECT

获取服务器的运动告警信息

**4.4.3.6**      CMD\_GET SENSOR\_ALARM

获取服务器的探头告警信息

**4.4.3.7**      CMD\_GET VIDEO\_LOST

获取服务器的视频丢失告警信息

**4.4.3.8**      CMD\_GET COMINFO

获取服务器的 COM 配置信息，以及云台信息

**4.4.3.9**      CMD\_GET USERINFO

获取用户的权限信息

**4.4.3.10**      CMD\_GET NETWORK

获取服务器的网络配置

**4.4.3.11**      CMD\_GET FTPUPDATA\_PARAM

获取服务器的 FTP 上传配置信息

**4.4.3.12** CMD\_GET\_PTZDATA

获取服务器的当前云台协议数据

**4.4.3.13** CMD\_GET\_SERVER\_COM2

获取 232 口的配置信息

**4.4.3.14** CMD\_GET\_ALARM\_OUT

获取探头输出配置信息

**4.4.3.15** CMD\_GET\_NOTIFY\_SERVER

获取服务器在线通知配置

**4.4.3.16** CMD\_GET\_PPPOE\_DDNS

获取服务器的 PPPOE 和 DDNS 配置信息

**4.4.3.17** CMD\_GET\_SENSOR\_STATE

获取服务器的探头状态

**4.4.3.18** CMD\_GET\_SERVER\_RECORD\_SET

获取服务器录像设置

**4.4.3.19** CMD\_GET\_SERVER\_RECORD\_STATE

获取服务器的录像状态

**4.4.3.20** CMD\_UNLOAD\_DISK

卸载服务器上挂载的 U 盘或者 MD 卡，当成功返回后才能拔除 U 盘或 SD 卡。

