# API接口 EsApi.dll (ver0.4)

1. 接口说明：

EsApi.dll是Windows X86 API接口, StdCall调用约定, Ansi字符编码。

函数(C语言)：

int WINAPI ES\_StartSession(LPCSTR Server, int Port, LPCSTR User, LPCSTR Password, INT\_PTR& Handle);

int WINAPI ES\_EndSession(INT\_PTR Handle);

int WINAPI ES\_SendKey(INT\_PTR Handle, unsigned int nChar, int iDown, bool bShiftDown,bool bCapLockDown);

int WINAPI ES\_SetRemoteMouse(INT\_PTR Handle, int x,int y, int z, BYTE bMask);

WORD\* WINAPI ES\_LockPicture(INT\_PTR Handle, int &Size, int &Width, int &Height);

int WINAPI ES\_UnlockPicture(INT\_PTR Handle);

typedef void(\_\_stdcall \*VideoSizeChangedCallback)(int newWidth, int newHeight);

typedef void(\_\_stdcall \*UpdatePicCallback)(INT\_PTR Param);

typedef void(\_\_stdcall \*ErrorCallback)(int err);

int WINAPI ES\_RegisterVideoSizeChangedCallback(INT\_PTR Handle, VideoSizeChangedCallback cb);

int WINAPI ES\_RegisterUpdatePicCallback(INT\_PTR Handle, UpdatePicCallback cb, INT\_PTR param);

int WINAPI ES\_RegisterErrCallback(INT\_PTR Handle, ErrorCallback cb);

函数说明：

1. int WINAPI ES\_StartSession(LPCSTR Server, int Port, LPCSTR User, LPCSTR Password, INT\_PTR & Handle);

开启一个会话。

参数：

Server：KVM的IP地址或域名

Port:连接端口，1024-65537（KVM出厂默认为6666）

User:用户名(KVM出厂默认为admin)，不能为NULL，空密码是””

Handle:如果会话成功建立，返回会话句柄。

返回：

0: 成功

-2：Server为NULL

-3：Port超出范围

-4：User为NULL

-5：Password为NULL

-1610612618: 访问失败，其他用户正在独占访问。

1. int WINAPI ES\_EndSession(INT\_PTR Handle);

结束一个会话。

参数：

Handle：会话句柄。

返回：

0：成功

-1：Handle无效

1. int WINAPI ES\_SendKey(INT\_PTR Handle, unsigned int nChar, int iDown, bool bShiftDown,bool bCapLockDown);

发送按键。

参数：

Handle：会话句柄。

nChar：按键值

iDown：

1：按下

0：弹起

bShiftDown：Shift键状态

bCapLockDown：CapLock键状态

返回：

0：成功

-1：Handle无效

1. int WINAPI ES\_SetRemoteMouse(INT\_PTR Handle, int x,int y, int z, BYTE bMask);

参数：

Handle：会话句柄。

x:X方向位移

y:Y方向位移

z:Z方向位移(滚轮)

bMask：可以按位或

#define BM\_LBUTTON 0x01 //鼠标左键

#define BM\_MBUTTON 0x02//鼠标中键

#define BM\_RBUTTON 0x04 //鼠标右键

#define BM\_SYNC 0x20 //先同步再移位

返回：

0：成功

-1：Handle无效

1. WORD\* WINAPI ES\_LockPicture(INT\_PTR Handle, int &Size, int &Width, int &Height);

锁定图像缓存。

参数：

Handle：会话句柄。

Size: 返回图像缓存大小

Width：返回图像宽

Height: 返回图像高

返回：

WORD\* 图像缓存指针，格式为RGB565，连续存储。

如果返回值不为NULL,则必须调用ES\_UnlockPicture以释放图像缓存锁。

1. int WINAPI ES\_UnlockPicture(INT\_PTR Handle);

解锁图像缓存

参数：

Handle：会话句柄。

返回：

0：成功

-1：Handle无效

1. int WINAPI ES\_RegisterVideoSizeChangedCallback(INT\_PTR Handle, VideoSizeChangedCallback cb);

注册显示模式改变回调函数

参数：

Handle：会话句柄。

Cb: 类型为VideoSizeChangedCallback的回调函数入口

返回：

0：成功

-1：Handle无效

1. int WINAPI ES\_RegisterUpdatePicCallback(INT\_PTR Handle, UpdatePicCallback cb, INT\_PTR param);

注册图像数据更新的回调函数

参数：

Handle：会话句柄。

Cb: 类型为UpdatePicCallback的回调函数入口

Param：这个参数直接传递给回调函数

返回：

0：成功

-1：Handle无效

1. int WINAPI ES\_RegisterErrCallback(INT\_PTR Handle, ErrorCallback cb);

注册错误回调函数

参数：

Handle：会话句柄。

Cb: 类型为ErrorCallback的回调函数入口

返回：

0：成功

-1：Handle无效

10. int WINAPI ES\_GetMouseMode(INT\_PTR Handle);

返回：

-1：Handle无效

0：相对同步

1: 绝对同步

1. 接口调用例程：

C#调用函数接口：

[DllImport("EsApi.dll", EntryPoint = "ES\_StartSession", CharSet = CharSet.Ansi)]

public static extern int ES\_StartSession(string Server, int Port, string User, string Password,ref IntPtr Handle);

[DllImport("EsApi.dll", EntryPoint = "ES\_EndSession", CharSet = CharSet.Ansi)]

public static extern int ES\_EndSession(IntPtr Handle);

[DllImport("EsApi.dll", EntryPoint = "ES\_LockPicture", CharSet = CharSet.Ansi]

public static extern IntPtr ES\_LockPicture(IntPtr Handle, ref int Size, ref int Width, ref int Height);

[DllImport("EsApi.dll", EntryPoint = "ES\_UnlockPicture", CharSet = CharSet.Ansi)]

public static extern int ES\_UnlockPicture(IntPtr Handle);

[DllImport("EsApi.dll", EntryPoint = "ES\_SendKey", CharSet = CharSet.Ansi)]

public static extern int ES\_SendKey(IntPtr Handle, uint nChar, int iDown, bool bShiftDown, bool bCapLockDown);

[DllImport("EsApi.dll", EntryPoint = "ES\_SetRemoteMouse", CharSet = CharSet.Ansi)]

public static extern int ES\_SetRemoteMouse(IntPtr Handle, int x, int y, int z, byte bMask);

public delegate void UpdatePicCallback(IntPtr lParam);

public delegate void VideoSizeChangedCallback(int newWidth, int newHeight);

public delegate void ErrorCallback(int err);

[DllImport("EsApi.dll", EntryPoint = "ES\_RegisterUpdatePicCallback", CharSet = CharSet.Ansi)]

public static extern int ES\_RegisterUpdatePicCallback(IntPtr Handle, UpdatePicCallback up\_callback, IntPtr param);

[DllImport("EsApi.dll", EntryPoint = "ES\_RegisterVideoSizeChangedCallback", CharSet = CharSet.Ansi)]

public static extern int ES\_RegisterVideoSizeChangedCallback(IntPtr Handle, VideoSizeChangedCallback sc\_callback);

[DllImport("EsApi.dll", EntryPoint = "ES\_RegisterErrCallback", CharSet = CharSet.Ansi)]

public static extern int ES\_RegisterErrCallback(IntPtr Handle, ErrorCallback er\_callback);

[DllImport("EsApi.dll", EntryPoint = "ES\_GetMouseMode ", CharSet = CharSet.Ansi)]

public static extern int ES\_GetMouseMode (IntPtr Handle);

版本：V0.4

增加ES\_GetMouseMode

-----------------------------------------------------------------------------------------------------------------------------------

版本：V0.3

版本更改：

1. 增加三个回调函数：

ES\_RegisterVideoSizeChangedCallback

ES\_RegisterVideoSizeChangedCallback

RegisterErrCallback

---------------------------------------------------------------------------------------------------------------------------------

版本：V0.2

版本更改：

1. 增加x64支持
2. Handle的类型更改为INT\_PTR