

**SADP 在线设备搜索**

**SDK 编程指南**

**V4.1**

# 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
1.1 SADP 介绍 .....	1
1.2 产品范围 .....	1
1.3 系统要求 .....	1
2 SDK 版本更新 .....	2
3 函数调用顺序 .....	3
4 函数说明 .....	4
4.1 SDK 版本信息和日志 .....	4
4.1.1 获取 SADP SDK 版本信息: SADP_GetSdpVersion .....	4
4.1.2 启用写日志文件: SADP_SetLogToFile .....	4
4.2 SADP 在线设备搜索 .....	4
4.2.1 启动 SADP: SADP_Start_V30 .....	4
4.2.2 停止 SADP 在线搜索: SADP_Stop .....	5
4.3 手动刷新 .....	5
4.3.1 手动刷新搜索: SADP_SendInquiry .....	5
4.4 设备配置 .....	5
4.4.1 修改设备网络参数: SADP_ModifyDeviceNetParam .....	5
4.4.2 设置设备 CMS 信息: SADP_SetCMSInfo .....	6
4.5 获取错误号 .....	6
4.5.1 获取错误号: SADP_GetLastError .....	6
4.6 清理搜索到的设备 .....	6
4.6.1 清理搜索到的设备: SADP_Clearup .....	6
4.7 获取设备信息 .....	7
4.7.1 获取设备信息: SADP_GetDeviceConfig .....	7
4.8 设备激活 .....	7
4.8.1 激活设备: SADP_ActivateDevice .....	7
4.9 设备密码恢复或重置 .....	8
4.9.1 密码恢复或重置: SADP_ResetPassword .....	8
5 结构体说明 .....	9
5.1 设备信息结构体: SADP_DEVICE_INFO .....	9
5.2 设备网络参数结构体: SADP_DEV_NET_PARAM .....	11
5.3 设备 CMS 参数结构体: SADP_CMS_PARAM .....	12
5.4 设备安全码结构体: SADP_SAFE_CODE .....	13
5.5 密码恢复或重置参数结构体: SADP_RESET_PARAM .....	13
6 附录 .....	14
6.1 错误号及说明 .....	14
6.2 SADP 启动失败排查 .....	14
6.3 附加说明 .....	16

# 1 SDK 简介

## 1.1 SADP 介绍

SADP 用于在线设备搜索，可搜索获取局域网里所有在线的我司设备，获取在线设备的序列号、IP 地址、端口等信息，并支持修改网络参数和恢复设备出厂密码。

SDK 中包含的内容如下：

Windows SDK	Sadp.h	头文件
	Sadp.lib	LIB 库文件
	Sadp.dll	DLL 库文件
	libeay32.dll	在激活和重置密码时需要加载
	npf.sys	32 位系统的驱动文件
	npf64.sys	64 位系统的驱动文件

SADP 服务依赖于列表所述驱动文件，需要将系统与系统相应的驱动文件，和 sadp.dll 放在的同一级目录，而且第一次安装时需要管理员权限。

## 1.2 产品范围

支持本公司发布的基线的后端、前端以及传输控制的各系列产品。

## 1.3 系统要求

**Windows 32 位设备网络 SDK：**

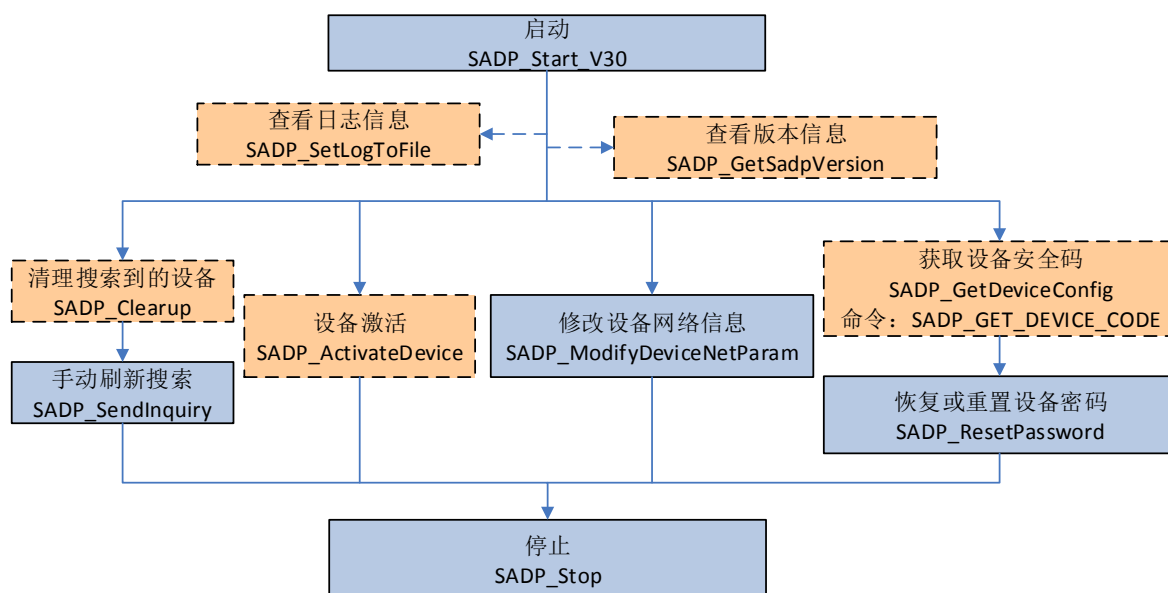
- Windows10、Windows8.1/8、Windows7、Windows Vista、Windows XP 以及 Windows Server 2008

## 2 SDK 版本更新

### Version 4.1.0.1 (build20150702)

- 新增获取设备安全码的接口：[SADP\\_GetDeviceConfig](#)，命令：SADP\_GET\_DEVICE\_CODE
- 新增设备激活接口：[SADP\\_ActivateDevice](#)。
- 新增密码重置接口：[SADP\\_ResetPassword](#)。
- [SADP\\_DEVICE\\_INFO](#)(设备信息)扩展：
  - 1)使用 1 个保留字节新增参数：byActivated(设备激活状态)。
  - 2) bySupport 新增取值：bySupport &0x20(是否支持恢复初始密码)、bySupport &0x40(是否支持重置密码)。
- 新增错误码：2019、2020、2021。

### 3 函数调用顺序



图中虚线框的流程是可选部分，不会影响其他流程和模块的功能使用，其中查看版本信息和获取错误码的接口可随时调用。

- [SADP\\_Start V30](#) 启动 SADP，在该接口设置的回调函数中可以获取到在线的设备信息，调用 [SADP\\_Stop](#) 停止 SADP 服务。SDK 默认 60s 自动刷新搜索在线设备，调用接口 [SADP\\_SendInquiry](#) 可以进行手动即时刷新搜索，刷新前可以调用 [SADP\\_Clearup](#) 清理设备尤其是下线设备。
- 启动 SADP SDK 写日志功能：[SADP\\_SetLogToFile](#)，可记录 ERROR 错误信息、DEBUG 调试信息和 INFO 普通信息等。
- SDK 版本信息、BUILD 信息，相关接口：[SADP\\_GetSdpVersion](#)。
- 出厂设备需要先激活，然后再使用激活使用的初始密码访问设备，相关接口：[SADP\\_ActivateDevice](#)
- 修改设备网络参数，包括设备 IP 地址、端口号、子网掩码、网关以及 DHCP 使能，相关接口：[SADP\\_ModifyDeviceNetParam](#)。
- 如果函数返回失败，可以调用接口 [SADP\\_GetLastError](#) 获取错误号来判断错误原因。

## 4 函数说明

### 4.1 SDK 版本信息和日志

#### 4.1.1 获取 SADP SDK 版本信息: **SADP\_GetSadpVersion**

函 数: unsigned int SADP\_GetSadpVersion(void)

参 数: 无

返回值: 版本号, 31~24 位值为 a, 23~16 位值为 b, 15~8 位值为 c, 7~0 位值为 d, 则版本号为 Va.b.c.d, 例如返回值为 0x3010103, 则版本号为 V3.1.1.3。

说 明: 获取 SADP SDK 版本信息

[返回目录](#)

#### 4.1.2 启用写日志文件: **SADP\_SetLogToFile**

函 数: BOOL SADP\_SetLogToFile(int nLogLevel, char const \*strLogDir, int bAutoDel)

参 数: [in] nLogLevel

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

目录路径字符串, 小于 256 个字符。为 NULL 时, Windows 默认路径"C:/SadpLog/"

[in] bAutoDel

是否删除超出的文件, 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 [SADP\\_GetLastError](#) 获取错误码, 通过错误码判断原因。

说 明: 启用写日志文件。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 (最多 10 个文件)。

[返回目录](#)

### 4.2 SADP 在线设备搜索

#### 4.2.1 启动 SADP: **SADP\_Start\_V30**

函 数: BOOL SADP\_Start\_V30(PDEVICE\_FIND\_CALLBACK pDeviceFindCallBack, int bInstallNPF, void \*pUserData)

参 数: [in] pDeviceFindCallBack

回调函数, 获取在线设备信息

[in] bInstallNPF

是否安装 NPF 服务, 只针对 window 系统, 默认安装: 0- 不安装,

非 0- 安装（安装需要管理员权限）

[in] pUserData

用户数据指针

```
typedef void (__stdcall * PDEVICE_FIND_CALLBACK)(const SADP_DEVICE_INFO *IpDeviceInfo, void *pUserData)
```

[out] lpDeviceInfo

搜索到的设备信息，详见 [SADP\\_DEVICE\\_INFO](#)

[out] pUserData

用户数据指针

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

说明： 启动 SADP 协议，开始搜索在线设备。

[返回目录](#)

## 4.2.2 停止 SADP 在线搜索: **SADP\_Stop**

函数： BOOL SADP\_Stop(void)

参数： 无

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

说明： 停止 SADP 协议，关闭在线设备搜索。该接口不能和其他接口并发调用，必须等其他接口完全返回后再调用，否则可能导致崩溃。

[返回目录](#)

## 4.3 手动刷新

### 4.3.1 手动刷新搜索: **SADP\_SendInquiry**

函数： BOOL SADP\_SendInquiry(void)

参数： 无

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

说明： 默认 60s 自动刷新搜索在线设备，调用该接口可以进行手动刷新。

[返回目录](#)

## 4.4 设备配置

### 4.4.1 修改设备网络参数: **SADP\_ModifyDeviceNetParam**

函数： BOOL SADP\_ModifyDeviceNetParam(const char\* sMAC, const char\* sPassword, const SADP\_DEV\_NET\_PARAM\* lpNetParam)

参数： [in] sMAC

设备物理地址

[in] sPassword

设备 admin 用户的密码

[in] lpNetParam

需要修改的网络参数，详见： [SADP\\_DEV\\_NET\\_PARAM](#)



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

说明: 修改设备 IP 地址、端口、网关、掩码、DHCP 等网络参数。

[返回目录](#)

## 4.4.2 设置设备 CMS 信息: **SADP\_SetCMSInfo**

函数: BOOL SADP\_SetCMSInfo(const char\* sMac, const SADP\_CMS\_PARAM \*lpCmsParam)

参数: [in] sMac 设备物理地址

[in] lpCmsParam 需要设置的 CMS 参数, 详见: [SADP\\_CMS\\_PARAM](#)

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

说明: 设置设备的 CMS 信息, 需要设备支持, 当前仅推模式设备 (如单兵、车载) 支持。

[返回目录](#)

## 4.5 获取错误号

### 4.5.1 获取错误号: **SADP\_GetLastError**

函数: unsigned int SADP\_GetLastError(void)

参数: 无

返回值: 返回最后操作的错误号。详见[错误码宏定义](#)

说明: 通过错误号可以判断接口返回失败的错误原因。

[返回目录](#)

## 4.6 清理搜索到的设备

### 4.6.1 清理搜索到的设备: **SADP\_Clearup**

函数: BOOL SADP\_Clearup(void)

参数: 无

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

说明: 清理搜索到的设备, 为了及时的更新上下线设备, 防止下线需要等待 120s 的情况。

[返回目录](#)

## 4.7 获取设备信息

### 4.7.1 获取设备信息: **SADP\_GetDeviceConfig**

函 数: BOOL SADP\_GetDeviceConfig(const char\* sDevSerialNO, DWORD dwCommand, void\* lpInBuffer, DWORD dwInBuffSize, void \*lpOutBuffer, DWORD dwOutBuffSize)

参 数: [in] sDevSerialNO 设备序列号  
[in] dwCommand 获取命令, 详见表 4.1  
[in] lpInBuffer 输入参数, 不同的命令对应不同的输入内容, 详见表 4.1  
[in] dwInBuffSize 输入缓冲区大小  
[out] lpOutBuffer 输出缓冲区, 不同的命令对应不同的输出内容, 详见表 4.1  
[in] dwOutBuffSize 输出缓冲区大小

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

说 明: 不同的获取功能对应不同的结构体和命令号, 如下表所示。

表 4.1 获取设备参数

dwCommand 宏定义	含义	lpInBuffer	lpOutBuffer	宏定义值
SADP_GET_DEVICE_CODE	获取设备安全码	NULL	<a href="#">SADP_SAFE_CODE</a>	1

[返回目录](#)

## 4.8 设备激活

### 4.8.1 激活设备: **SADP\_ActivateDevice**

函 数: BOOL SADP\_ActivateDevice(const char\* sDevSerialNO, const char\* sCommand)

参 数: [in] sDevSerialNO 设备序列号  
[in] sCommand 设置的密码, 密码等级弱或者以上

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

说 明: 出厂设备需要先激活, 然后再使用激活使用的初始密码访问设备。将密码输入分为数字(0~9)、小写字母(a~z)、大写字母(A~Z)、特殊符号(:\除外) 4 类, 等级分为 4 个等级, 如下所示:

- 等级 0 (风险密码): 密码长度小于 8 位, 或者只包含 4 类字符中的任意一类, 或者密码与用户名一样, 或者密码是用户名的倒写。例如: 12345、abcdef。
- 等级 1 (弱密码): 包含两类字符, 且组合为 (数字+小写字母) 或 (数字+大写字母), 且长度大于等于 8 位。例如: abc12345、123ABCDEF。
- 等级 2 (中密码): 包含两类字符, 且组合不能为 (数字+小写字母) 和 (数字+大写字母), 且长度大于等于 8 位。例如: 12345\*\*\*++、ABCDabcd。
- 等级 3 (强密码): 包含三类字符及以上, 且长度大于等于 8 位。例如: Abc12345、abc12345++。

[返回目录](#)

## 4.9 设备密码恢复或重置

### 4.9.1 密码恢复或重置: **SADP\_ResetPassword**

函 数: BOOL SADP\_ResetPassword(const char\* sDevSerialNO, const SADP\_RESET\_PARAM \*pResetParam)

参 数: [in] sDevSerialNO 设备序列号

[in] pResetParam 密码恢复或重置参数结构体, 详见: [SADP\\_RESET\\_PARAM](#)

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

说 明:

- 普通设备(设备信息 [SADP\\_DEVICE\\_INFO](#) 中 bySupport 字段 0x20 和 0x40 都为 0 的设备)直接调用该接口, 传入设备厂商提供的特殊字符串(SADP\_RESET\_PARAM 中 szCode, 其他参数无效), 进行密码恢复, 恢复为设备出厂密码。
- 支持恢复设备初始密码安全升级的设备(设备信息 [SADP\\_DEVICE\\_INFO](#) 中 bySupport 字段 0x20 为 1 且 0x40 为 0 的设备), 先调用获取设备信息接口([SADP\\_GetDeviceConfig](#)), 得到设备返回的安全码, 将安全码提供给设备厂商获取对应的加密后字符串, 再调用该接口传入加密字符串(SADP\_RESET\_PARAM 中 szCode, 其他参数无效), 恢复为设备出厂密码。
- 支持密码重置的设备(设备信息 [SADP\\_DEVICE\\_INFO](#) 中 bySupport 字段 0x40 为 1 的设备), 先调用获取设备信息接口([SADP\\_GetDeviceConfig](#)), 得到设备返回的随机串(安全码), 将安全码提供给设备厂商获取对应授权文件, 然后再调用该接口上传授权文件(SADP\_RESET\_PARAM 中 szAuthFile), 设备重置为新密码(SADP\_RESET\_PARAM 中 szPassword)。
- 密码恢复所需的特殊字符串或加密字符串、重置所需的加密文件, 不同的设备在不同的时间恢复或者重置都是不同的, 请联系设备厂商咨询和获取。

[返回目录](#)

## 5 结构体说明

### 5.1 设备信息结构体: **SADP\_DEVICE\_INFO**

```
typedef struct tagSADP_DEVICE_INFO
{
    char            szSeries[12];
    char            szSerialNO[48];
    char            szMAC[20];
    char            szIPv4Address[16];
    char            szIPv4SubnetMask[16];
    unsigned int    dwDeviceType;
    unsigned int    dwPort;
    unsigned int    dwNumberOfEncoders;
    unsigned int    dwNumberOfHardDisk;
    char            szDeviceSoftwareVersion[48];
    char            szDSPVersion[48];
    char            szBootTime[48];
    int             iResult;
    char            szDevDesc[24];
    char            szOEMInfo[24];
    char            szIPv4Gateway[16];
    char            szIPv6Address[46];
    char            szIPv6Gateway[46];
    unsigned char    byIPv6MaskLen;
    unsigned char    bySupport;
    unsigned char    byDhcpEnabled;
    unsigned char    byDeviceAbility;
    unsigned short   wHttpPort;
    unsigned short   wDigitalChannelNum;
    char            szCmsIPv4[16];
    unsigned short   wCmsPort;
    unsigned char    byOEMCode;
    unsigned char    byActivated;
    char            szBaseDesc[24];
    unsigned char    byRes[16];
}SADP_DEVICE_INFO, *LPSADP_DEVICE_INFO;
```

#### Members

*szSeries*

设备系列（保留）

*szSerialNO*

设备序列号

*szMAC*

设备物理地址

*szIPv4Address*

设备 IPv4 地址

*szIPv4SubnetMask*

设备 IPv4 子网掩码

*dwDeviceType*

设备类型，具体数值代表的设备型号请见[附录：设备类型](#)

*dwPort*

设备服务端口号

*dwNumberOfEncoders*

设备编码器个数，即设备编码通道个数。对于解码器，其值设为 0

*dwNumberOfHardDisk*

设备硬盘数目

*szDeviceSoftwareVersion*

设备软件版本号

*szDSPVersion*

设备 DSP 版本号

*szBootTime*

开机时间

*iResult*

信息类型，设备重启、设备下线、设备上线、设备更新等，详见下表

宏定义	值	含义
SADP_ADD	1	新设备上线，之前在 SADP 库列表中未出现的设备
SADP_UPDATE	2	在线的设备 IP、子网掩码、端口、硬盘或编码器个数改变
SADP_DEC	3	设备下线，设备自动发送下线消息或 30 秒内检测不到设备
SADP_RESTART	4	之前 SADP 库列表中出现过之后下线的设备再次上线
SADP_UPDATEFAIL	5	设备更新失败

*szDevDesc*

设备类型描述，与 *dwDeviceType* 对应

*szOEMInfo*

OEM 产商信息

*szIPv4Gateway*

设备 IPv4 网关

*szIPv6Address*

设备 IPv6 地址

*szIPv6Gateway*

设备 IPv6 网关

*byIPv6MaskLen*

IPv6 子网前缀长度

*bySupport*

按位表示，对应值为 1 表示支持：

*bySupport* & 0x01: 表示是否支持 ipv6

*bySupport* & 0x02: 表示是否支持修改 ipv6 参数

bySupport & 0x04: 表示是否支持 Dhcp  
 bySupport & 0x08: 表示是否支持 udp 多播  
 bySupport & 0x10: 表示是否含加密节点  
 bySupport & 0x20: 表示是否恢复初始密码  
 bySupport & 0x40: 表示是否支持重置密码

byDhcpEnabled

Dhcp 状态: 0- 不启用, 1- 启用

byDeviceAbility

设备能力集:

0- 设备不支持以下功能: 设备类型描述、OEM 厂商、IPv4 网关、IPv6 地址、IPv6 网关、IPv6 子网前缀、DHCP

1- 设备支持上述功能

wHttpPort

设备 HTTP 端口号

wDigitalChannelNum;

设备数字通道数目

szCMSIPv4

CMS IPv4 地址

wCmsPort

CMS 端口

byOEMCode

OEM 标识 0- 基线设备 1- OEM 设备

byActivated

设备是否激活: 0- 激活, 1- 未激活 (老的设备都是已激活状态)

szBaseDesc

基线短型号, 不随定制而修改的型号, 用于萤石平台进行型号对比

byRes

保留

[返回目录](#)

## 5.2 设备网络参数结构体: SADP\_DEV\_NET\_PARAM

```
typedef struct tagSADP_DEV_NET_PARAM
{
    char        szIPv4Address[16];
    char        szIPv4SubNetMask[16];
    char        szIPv4Gateway[16];
    char        szIPv6Address[128];
    char        szIPv6Gateway[128];
    unsigned short  wPort;
    unsigned char  byIPv6MaskLen;
    unsigned char  byDhcpEnable;
    unsigned char  byRes[128];
}SADP_DEV_NET_PARAM, *LPSADP_DEV_NET_PARAM;
```

## Members

*szIPv4Address*

IPv4 地址

*szIPv4SubNetMask*

IPv4 子网掩码

*szIPv4Gateway*

IPv4 网关

*szIPv6Address*

IPv6 地址

*szIPv6Gateway*

IPv6 网关

*wPort*

设备监听端口

*byIPv6MaskLen*

IPv6 掩码长度

*byDhcpEnable*

DHCP 使能: 0- 禁用, 1- 启用

*byRes*

保留

[返回目录](#)

## 5.3 设备 CMS 参数结构体: **SADP\_CMS\_PARAM**

```
typedef struct tagSADP_CMS_PARAM
```

```
{
```

```
    char            szPUID[32];
```

```
    char            szPassword[16];
```

```
    char            szCmsIPv4[16];
```

```
    char            szCmsIPv6[128];
```

```
    unsigned short   wCmsPort;
```

```
    unsigned char    byRes[30];
```

```
}SADP_CMS_PARAM, *LPSADP_CMS_PARAM;
```

## Members

*szPUID*

预分配的 PUID

*szPassword*

设置的登录密码

*szCmsIPv4*

CMS 服务器 IPv4 地址

*szCmsIPv6*

CMS 服务器 IPv6 地址

*wCmsPort*

CMS 服务器监听端口

*byRes*

保留

[返回目录](#)

## 5.4 设备安全码结构体: **SADP\_SAFE\_CODE**

```
typedef struct tagSADP_SAFE_CODE
{
    DWORD    dwCodeSize;
    char      szDeviceCode[MAX_DEVICE_CODE];
    BYTE      byRes[128];
}SADP_SAFE_CODE, *LPSADP_SAFE_CODE;
```

### Members

*dwCodeSize*

设备返回码长度

*szDeviceCode*

设备返回码

`#define MAX_DEVICE_CODE 128` //设备返回码最大长度

*byRes*

保留

[返回目录](#)

## 5.5 密码恢复或重置参数结构体: **SADP\_RESET\_PARAM**

```
typedef struct tagSADP_RESET_PARAM
{
    char      szCode[MAX_ENCRYPT_CODE];
    char      szAuthFile[MAX_FILE_PATH_LEN];
    char      szPassword[MAX_PASS_LEN];
    unsigned char  byRes[512];
}SADP_RESET_PARAM, *LSADP_RESET_PARAM;
```

### Members

*szCode*

恢复密码的特殊字符串或加密字符串

`#define MAX_ENCRYPT_CODE 256` //设备返回码加密后长度

*szAuthFile*

重置授权文件，包括文件名的绝对路径，例如：“D:/DeviceKey.xml”

`#define MAX_FILE_PATH_LEN 260` //文件最大路径长度

*szPassword*

新密码，重置之后的密码

`#define MAX_PASS_LEN 16` //设备最大密码长度

*byRes*

保留

[返回目录](#)



## 6 附录

### 6.1 错误号及说明

错误名称	错误值	说明
SADP_NOERROR	0	没有错误
SADP_ALLOC_RESOURCE_ERROR	2001	资源分配错误
SADP_NOT_START_ERROR	2002	SADP 未启动
SADP_NO_ADAPTER_ERROR	2003	无网卡
SADP_GET_ADAPTER_FAIL_ERROR	2004	获取网卡信息失败
SADP_PARAMETER_ERROR	2005	参数错误
SADP_OPEN_ADAPTER_FAIL_ERROR	2006	打开网卡失败
SADP_SEND_PACKET_FAIL_ERROR	2007	发送数据失败
SADP_SYSTEM_CALL_ERROR	2008	系统接口调用失败
SADP_DENY_OR_TIMEOUT_ERROR	2009	设备拒绝处理
SADP_NPF_INSTALL_FAILED	2010	安装 NPF 服务失败
SADP_TIMEOUT	2011	设备响应超时
SADP_CREATE_SOCKET_ERROR	2012	创建 socket 失败
SADP_BIND_SOCKET_ERROR	2013	绑定 socket 失败
SADP_JOIN_MULTI_CAST_ERROR	2014	加入多播组失败
SADP_NETWORK_SEND_ERROR	2015	发送出错
SADP_NETWORK_RECV_ERROR	2016	接收出错
SADP_XML_PARSE_ERROR	2017	多播 XML 解析出错
SADP_LOCKED	2018	设备锁定
SADP_NOT_ACTIVATED	2019	设备未激活
SADP_RISK_PASSWORD	2020	风险高的密码
SADP_HAS_ACTIVATED	2021	设备已激活

### 6.2 SADP 启动失败排查

接口返回失败，可以获取错误码，通过错误码判断原因，启动失败常见原因是 NPF 驱动安装问题，错误码为 SADP\_NPF\_INSTALL\_FAILED。

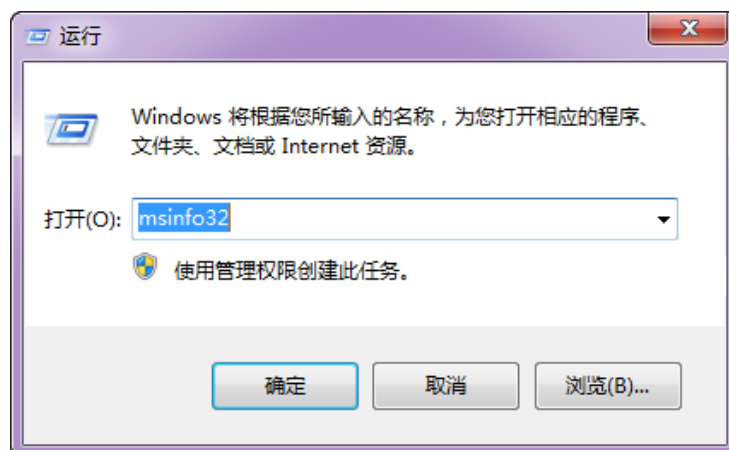
在使用 SADP 时，需要安装 NPF 服务（一款抓包必备的驱动服务，没有安装便抓不到包），用户有可以手动安装（比如安装 winpcap），也可以通过 SADP\_Start\_V30 接口来自动安装。无论是手动安装还是自动安装，均需要 Administrator 的权限

手动安装成功之后，SADP\_Start\_V30 接口中的 bInstallNPF 传入 false 即可，这里不讨论此种用法。

自动安装，即 bInstallNPF 需要传入 true，如果驱动服务安装失败，接口会返回 FALSE，错误码为 SADP\_NPF\_INSTALL\_FAILED，下面介绍一下出现这种情况的处理方法：

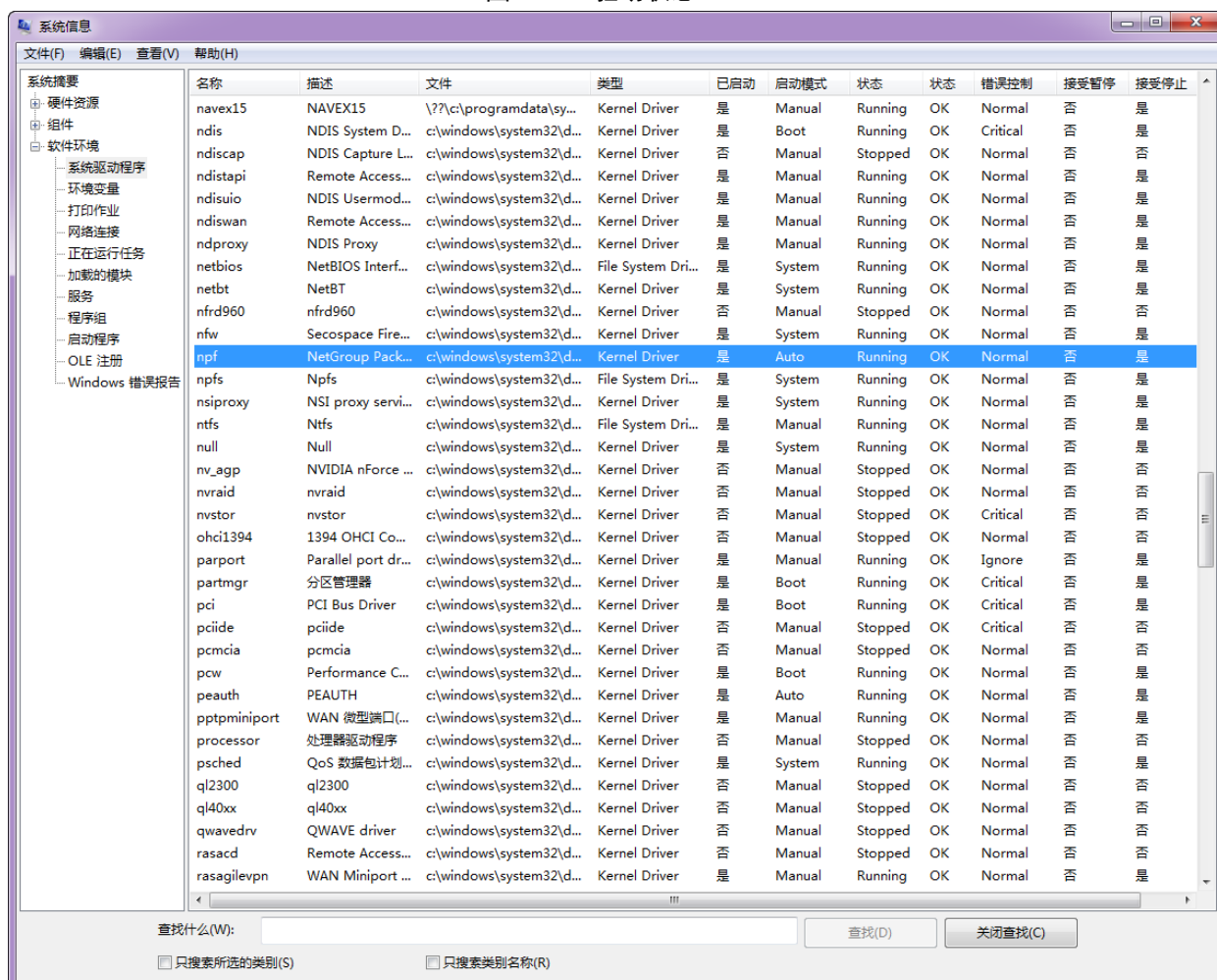
- 1.在运行中输入 msinfo32，见图 1。

图 1 运行 msinfo32



2.按确定键之后,弹出系统信息,选择软件环境-系统驱动程序-npf,见图 2。

图 2 NPF 驱动状态

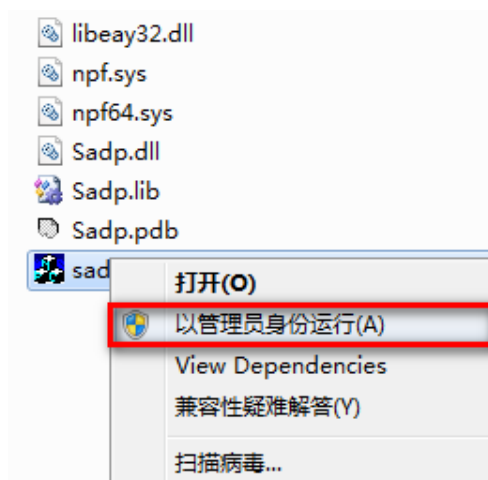


先确定 npf 是否安装成功,安装成功就是图 2 中的情况,状态为正在运行。如果安装失败, npf 的运行状态应该是停止的,注意,在 win7 等高版本系统下,在系统驱动信息中可能找不到 npf,也表示安装失败了。

3.如果 npf 安装失败了,请退出所有与 sadp 相关的应用程序,确保在 Administrator 下重新调用 SADP\_Start\_V30 接口(如果是使用 sadp demo,可以右键以管理员身份运行 sadpdlg.exe,见图 3), bInstallNPF

置为 true 重新安装服务（安装之前会删掉系统中残余的 npf）

图 3 运行 Demo



4.如果还是出现安装失败的情况，那么可以使用批处理文件手动删除 npf 服务，在文本文件中写入：

```
sc stop NPF
```

```
sc delete NPF
```

保存为 bat 文件，同样需要在 Administrator(以管理员身份)下运行，运行之后继续查看在图 2 中查看是否已经卸载掉了 npf，查看之前需要刷新一下，在图 2 中的查看-刷新。如果此时还在系统中，那么十有八九是使用 sadp 的相关程序未退出，或是没有用 Administrator 权限运行 bat 文件。

## 6.3 附加说明

如果 PC 上有防火墙或者带防火墙功能的杀毒软件，禁用 Sadp 的抓包服务，会导致搜索不到设备。所以使用 Sadp 之前确保未安装或禁用相关防火墙，比如带防火墙版的赛门铁克杀毒软件。