c++获取网卡MAC地址

转载

ccfxue 于 2016-04-19 10:57:06 发布 2638 收藏 2
分类专栏： MFC


MFC 专栏收录该内容
161 篇文章 0 订阅
订阅专栏

一台机器上可能有多个网卡，每一个网卡只有一个MAC地址，但是每一个网卡可能配置有多个IP地址；如平常的笔记本电脑中

typedef struct _IP_ADAPTER_INFO

{

struct _IP_ADAPTER_INFO* Next; //指向同类型节点，即下一个网卡（如果有多个的话）

char AdapterName[MAX_ADAPTER_NAME_LENGTH +4]; //网卡名

charDescription[MAX_ADAPTER_DESCRIPTION_LENGTH + 4]; //网卡描述信息

UINT AddressLength; //网卡MAC长度

BYTEAddress[MAX_ADAPTER_ADDRESS_LENGTH]; //网卡MAC地址

IP_ADDR_STRING lpAddressList; //网卡IP列表

…….

…….

} IP_ADAPTER_INFO,

由于可能有多个网卡，因此struct _IP_ADAPTER_INFO* Next字段为一个链表结构指针，由于一个网卡可能有多个IP，因此IP_ADDR_STRING字段应该也是一个链表结构，其信息如下所示：

typedef struct _IP_ADDR_STRING

{

struct _IP_ADDR_STRING* Next; //指向同类型节点，即下一个IP（如果有多IP的话）

IP_ADDRESS_STRING lpAddress; //IP地址信息

IP_MASK_STRING lpMask;

DWORD Context;

} IP_ADDR_STRING,

IP_ADDR_STRING结构也是一个链表节点

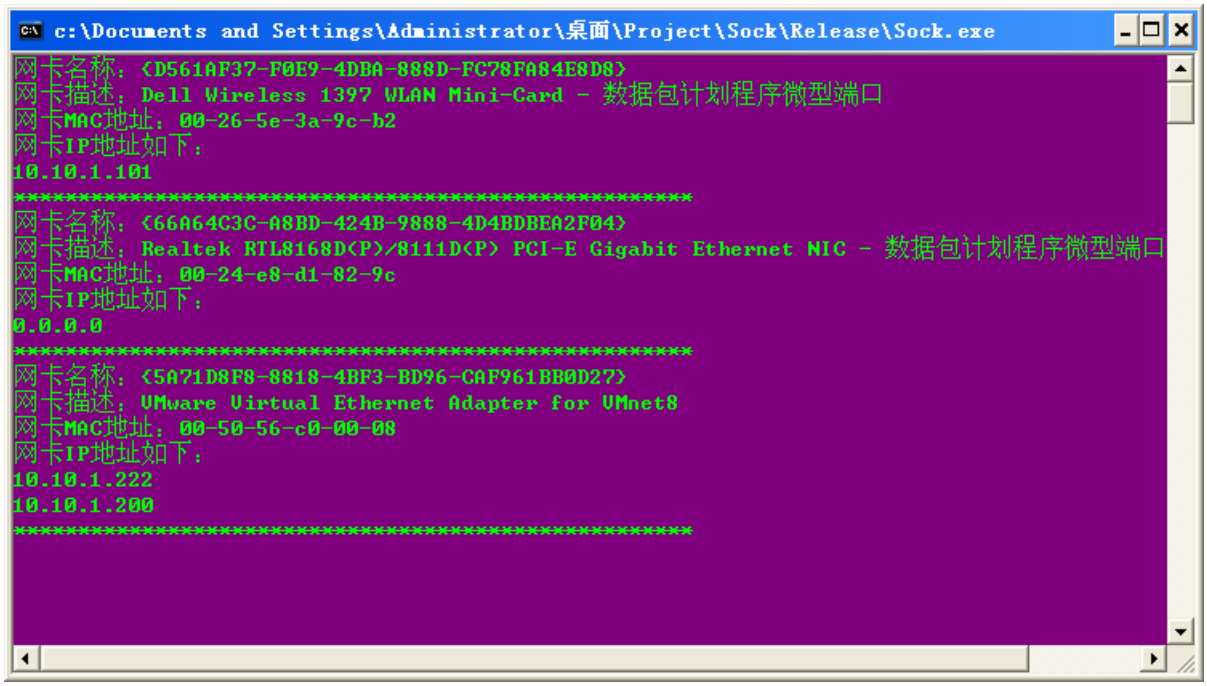
综上所述，用下图来描述网卡的结构存储信息，也许更明朗：

```
// 头文件包含
#include "stdafx.h"
#include <WinSock2.h>
#include <Iphlpapi.h>
#include <iostream>
```

```
#include <iostream>
using namespace std;
// 函数声明
void output(PIP_ADAPTER_INFO pIpAdapterInfo);
// 程序入口
int _tmain(int argc, _TCHAR* argv[])
{
   //PIP_ADAPTER_INFO结构体指针存储本机网卡信息
   PIP_ADAPTER_INFO pIpAdapterInfo = new IP_ADAPTER_INFO();
   //得到结构体大小,用于GetAdaptersInfo参数
   unsigned long stSize = sizeof(IP_ADAPTER_INFO);
   //调用GetAdaptersInfo函数,填充pIpAdapterInfo指针变量;其中stSize参数既是一个输入量也是一个输出量
   int nRel = GetAdaptersInfo(pIpAdapterInfo,&stSize);
   if (ERROR_BUFFER_OVERFLOW==nRel)
   {
      //如果函数返回的是ERROR_BUFFER_OVERFLOW
       //则说明GetAdaptersInfo参数传递的内存空间不够,同时其传出stSize,表示需要的空间大小
      //这也是说明为什么stSize既是一个输入量也是一个输出量
      //释放原来的内存空间
      delete pIpAdapterInfo;
      //重新申请内存空间用来存储所有网卡信息
      pIpAdapterInfo = (PIP_ADAPTER_INFO)new BYTE[stSize];
      //再次调用GetAdaptersInfo函数,填充pIpAdapterInfo指针变量
      nRel=GetAdaptersInfo(pIpAdapterInfo,&stSize);
   }
   if (ERROR_SUCCESS==nRel)
   {
      //输出网卡信息
      output(pIpAdapterInfo);
   }
   //释放内存空间
   if (pIpAdapterInfo)
   {
      delete pIpAdapterInfo;
   }
   getchar();
   return 0;
}
///函数作用,输出网卡信息
void output(PIP_ADAPTER_INFO pIpAdapterInfo)
{
   //可能有多网卡,因此通过循环去判断
   while (pIpAdapterInfo)
   {
      cout<<"网卡名称："<<pIpAdapterInfo->AdapterName<<endl;
      cout<<"网卡描述："<<pIpAdapterInfo->Description<<endl;
      cout<<"网卡MAC地址："<<pIpAdapterInfo->Address;
      for (UINT i = 0; i < pIpAdapterInfo->AddressLength; i++)
      if (i==pIpAdapterInfo->AddressLength-1)
      {
         printf("%02x\n", pIpAdapterInfo->Address[i]);
```

```
    }
    else
    {
      printf("%02x-", pIpAdapterInfo->Address[i]);
    }
    cout<<"网卡IP地址如下："<<endl;
    //可能网卡有多IP,因此通过循环去判断
    IP_ADDR_STRING *pIpAddrString =&(pIpAdapterInfo->IpAddressList);
    do
    {
      cout<<pIpAddrString->IpAddress.String<<endl;
      pIpAddrString=pIpAddrString->Next;
    } while (pIpAddrString);
    pIpAdapterInfo = pIpAdapterInfo->Next;
    cout<<"*****************************************************"<<endl;
  }
  return;
}
```

程序运行结果如下图所示：



获取网卡的MAC地址的方法很多，如：Netbios，SNMP，GetAdaptersInfo等。

获取网卡的**MAC**地址的方法很多，如：Netbios，SNMP，GetAdaptersInfo等。经过测试发现 Netbios 方法在网线拔出的情况下获取不到MAC，而 SNMP 方法有时会获取多个重复的网卡的MAC，试来试去还是 GetAdaptersInfo 方法比较好，网线拔出的情况下可以获取MAC，而且很准确，不会重复获取网卡。

GetAdaptersInfo 方法也不是十全十美，也存在些问题：

1）如何区分物理网卡和虚拟网卡；

2）如何区分无线网卡和有线网卡；

3）“禁用”的网卡获取不到。

关于问题1和问题2我的处理办法是：

区分物理网卡和虚拟网卡：pAdapter->Description中包含"PCI"是：物理网卡。（试了3台机器可以）

区分无线网卡和有线网卡：pAdapter->Type为71的是：无线网卡。（试了2个无线网卡也可以）

现在把代码贴出来和大家分享： #include"stdafx.h"

```
#include<atlbase.h>
#include<atlconv.h>
#include"iphlpapi.h"
#pragmacomment(lib,"Iphlpapi.lib")
intmain(intargc,char*argv[])
{
    PIP_ADAPTER_INFOpAdapterInfo;
    PIP_ADAPTER_INFOpAdapter=NULL;
    DWORDdwRetVal=0;
    pAdapterInfo=(IP_ADAPTER_INFO*)malloc(sizeof(IP_ADAPTER_INFO));
    ULONGulOutBufLen=sizeof(IP_ADAPTER_INFO);
    if(GetAdaptersInfo(pAdapterInfo,&ulOutBufLen)!=ERROR_SUCCESS)
    {
        GlobalFree(pAdapterInfo);
        pAdapterInfo=(IP_ADAPTER_INFO*)malloc(ulOutBufLen);
    }
    if((dwRetVal=GetAdaptersInfo(pAdapterInfo,&ulOutBufLen))==NO_ERROR)
    {
        pAdapter=pAdapterInfo;
        while(pAdapter)
        {
            if(
                strstr(pAdapter->Description,"PCI")>0//pAdapter->Description中包含"PCI"为：物理网卡
                ||pAdapter->Type==71                    //pAdapter->Type是71为：无线网卡
                )
            {
                printf("--------------------------------------------------------n");
                printf("AdapterName:t%sn",pAdapter->AdapterName);
                printf("AdapterDesc:t%sn",pAdapter->Description);
                printf("AdapterAddr:t");
                for(UINTi=0;i<pAdapter->AddressLength;i++)
                {
                    printf("%02X%c",pAdapter->Address[i],
                                i==pAdapter->AddressLength-1?"n":"-");
                }
                printf("AdapterType:t%dn",pAdapter->Type);
                printf("IPAddress:t%sn",pAdapter->IpAddressList.IpAddress.String);
                printf("IPMask:t%sn",pAdapter->IpAddressList.IpMask.String);
            }
            pAdapter=pAdapter->Next;
        }
    }
    else
    {
        printf("CalltoGetAdaptersInfofailed.n");
    }
    return0;
}
```

创作打卡挑战赛
赢取流量/现金/CSDN周边激励大奖