

# Re: How to know if a network adapter is connected

---

*Source:*

<http://www.tech-archive.net/Archive/WindowsCE/microsoft.public.windowsce.embedded.vc/2008-10/msg00052.htm>

---

- *From:* "fred\_d" <duchassin@xxxxxxxxxx>
  - *Date:* Mon, 13 Oct 2008 10:17:39 +0200
- 

Hello,

Binding a socket to an adapter instructs the NDIS to receive only from this socket. This is the only way I know to identify the network adapter, where a packet is received. You have to bind the adapter before calling any send/receive function. Sorry.

Yes i agree but if the adapter is not connected to the network, the bind() function return SOCKET\_ERROR. So i need to bind every socket each time i call the recv() ???

This is an extract of my source code:

```
DWORD WINAPI DHCPThread(LPVOID pVoid)
{
    SOCKET sockETH, sockUSB;
    struct sockaddr_in addrETH;
    struct sockaddr_in addrUSB;
    struct sockaddr_in addr;
    BYTE buf[548];
    int lenETH = sizeof(buf);
    int lenUSB = sizeof(buf);
    int one=1;
    int ret;
    WSADATA WSAData;

    char ChaineIP[25];
    int BindUSB, BindETH;
```

```
// Initliatize the Winsock dll version 2.0
WSAStartup(MAKEWORD(2,0), &WSAData);
```

## Re: How to know if a network adapter is connected

```
sockETH = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
sockUSB = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
if ( (sockETH == INVALID_SOCKET) || (sockUSB == INVALID_SOCKET) )
{
    closesocket(sockETH);
    closesocket(sockUSB);
    return 0;
} // if

VisuAdresseIPEthernet(); //--> 135.150.11.65 in Chaine[0]
strcpy(&ChaineIP[0], &Chaine[0]);
addrETH.sin_family = AF_INET;
addrETH.sin_addr.s_addr = inet_addr(&ChaineIP[0]);
addrETH.sin_port = htons(67);
BindETH = bind(sockETH, (LPSOCKADDR)&addrETH, sizeof(addrETH)); //bind
ethernet adapter

VisuAdresseIPUSB(); //--> 192.168.0.50 in Chaine[0]
strcpy(&ChaineIP[0], &Chaine[0]);
addrUSB.sin_family = AF_INET;
addrUSB.sin_addr.s_addr = inet_addr(&ChaineIP[0]);
addrUSB.sin_port = htons(67);
BindUSB = bind(sockUSB, (LPSOCKADDR)&addrUSB, sizeof(addrUSB)); //bind
USB adapter

ret=setsockopt(sockETH, SOL_SOCKET, SO_BROADCAST, (const char *) &one,
sizeof(one));
ret=setsockopt(sockUSB, SOL_SOCKET, SO_BROADCAST, (const char *) &one,
sizeof(one));

while (!g_stopDHCP)
{
    if ( (lenETH=recv(sockETH, buf, 548, 0))>0)|| (lenUSB=recv(sockUSB, buf,
548, 0))>0 )
    {

        //...code.....//
        //preparation of DHCP buffer.

        if (lenUSB > 0)
        {
            memcpy(&buf[0x3a-0x2a], &AddrIPDHCP_USB, 4);
            ret=sendto(sockUSB, buf, lenUSB, 0, (LPSOCKADDR)&addr, sizeof(addr));
        }
        else if (lenETH > 0)
        {
            memcpy(&buf[0x3a-0x2a], &AddrIPDHCP_ETH, 4);
            ret=sendto(sockETH, buf, lenETH, 0, (LPSOCKADDR)&addr, sizeof(addr));
        }
    }
}
```

Re: How to know if a network adapter is connected

```
} // if  
  
} // while  
  
shutdown(sockETH, SD_BOTH);  
shutdown(sockUSB, SD_BOTH);  
  
closesocket(sockETH);  
closesocket(sockUSB);  
  
return 0;  
  
}
```

I try to bind the socket each time before calling recv().  
If you see some errors please.

When you write a service  
(<http://msdn.microsoft.com/en-us/library/aa446909.aspx>) you get an IOCTL  
call (<http://msdn.microsoft.com/en-us/library/ms891090.aspx>), when the  
adapter state has changed. You can use this notification to update your  
sockets and bind to newly connected adapters.

I don't understand what you want to tell me. I doesn't write a service. This  
dhcp server is just a thread in my application. Maybe it's not very clean  
but it's the quite easy.

Maybe i could use the GetHostByName function like this:

```
pHost = gethostbyname("RNDISFN1");  
if (pHost!=NULL)  
{  
//USB connection is valid  
}
```

Thanks

.