Setting Up OS/2 Peer-to-Peer Networking & Coexistence of Warp & NT Machines

©1998, Frank R. Field, III (furd@mit.edu) (Last revision: February 13, 1998}

Ed. Original version is http://web.mit.edu/activities/os2/peer/WARPPEER.HTM. I have deleted some text trying to keep this short.

Purpose

I wanted to be able to access files on an NT 4 Workstation system from my Warp 4 system.

Nomenclature

In OS/2 Warp, peer to peer networking is called "File and Printer Sharing" (FPS). It allows users on a network to share files and use each other's hardware (like printers and modems) without a central server and the associated administrative & maintenance requirements. This capability was added to Warp 3 and it was then repackaged as Warp Connect; peer services come with all versions of Warp 4.

It's probably worth installing FPS when you install the operating system; since there are problems adding it after the initial OS installation, at least in Warp 4. Newsgroup messages suggest that the FPS installation routines have bugs that can keep the installation from taking place, apparently caused by long lines in the CONFIG.SYS file -- and we all know how long a LIBPATH statement can get over time!

Warp 4 achieves peer to peer through the use of NetBIOS, which is a local area networking protocol; it is not "routable," meaning that its use across (rather than within) local networks is difficult. Network operating system designers then developed NetBIOS over TCP/IP, using a routable protocol (TCP/IP) as the underpinnings for a application programming interface that looked just like NetBIOS (in the OSI network standard, NetBIOS operates at Session Layer, while TCP is used at Transport Layer).

Resources are made available by NetBIOS name, and names are defined according to a fairly easy standard: two backslashes, then the name of the machine on the network, then a backslash, and finally the name of the resource itself. For example \\FURDBOX\FILEDIR is the way NetBIOS wants to hear you name the resource FILEDIR on the computer known as FURDBOX on the network. Note that FURDBOX is not the name of your machine in the mit.edu domain; it can be, but there is no requirement that it match. Similarly, FILEDIR need not be the name of any file or file directory on the FURDBOX machine.

However, one key limitation is that NT seems to be able to give resources names that are longer than 8 characters. Warp's NetBIOS signals that such resources exist, but it cannot handle names of more than 8 characters. Warp gives you the relatively useless message "More data available" when you connect to machine with a longer name. Any machine with a resource having such a lengthy name will confuse the Warp network services enough that the GUI tools in the WPS will not work correctly for any of the resources available. I have found it to be a problem as of Warp 4, Fixpak 5.

Getting It To Work

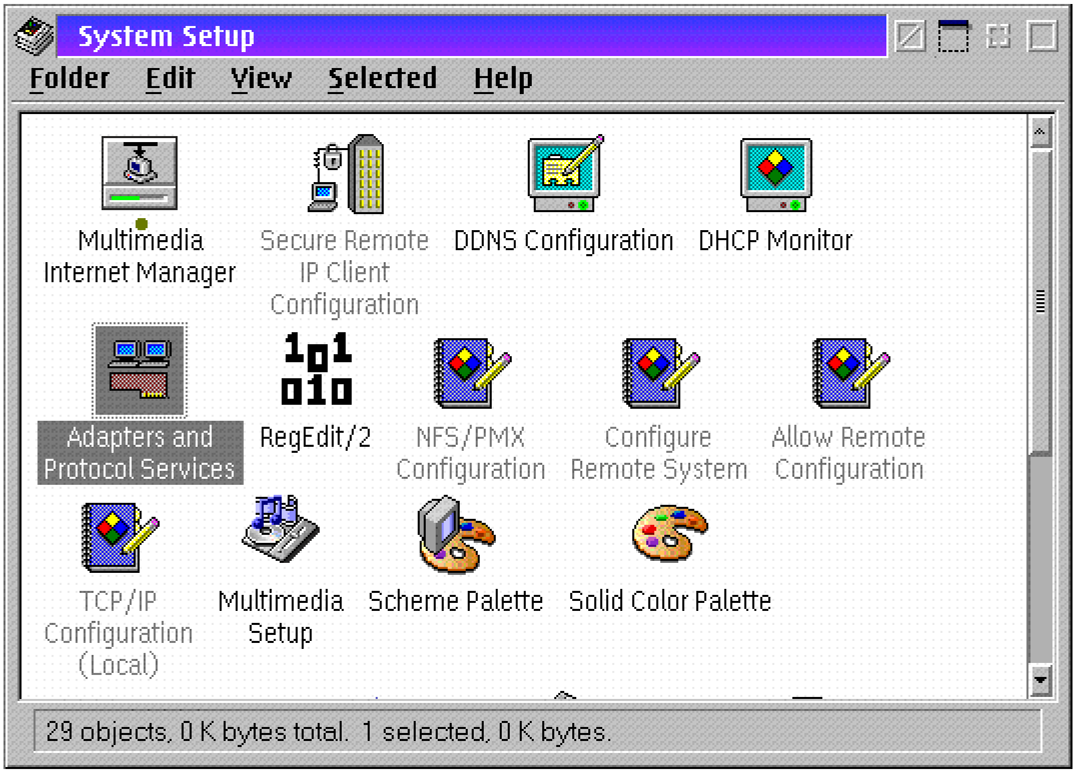
Step One: Install File and Printer Sharing.

Don't get ahead of yourself -- you will see some dialogs in the course of getting it installed that will look just like those in later steps; ignore them. Just get the darn thing installed. If you are successful, you will find, following the reboot, that the Desktop Connections object will include a new folder that is labeled Network. There may also be some new things on the screen during boot -- the key thing will be the addition of a line in your config.sys file that installs a new file system - NETWKSTA.200.

There are fixpacks for peer services. The most current as of February 1998 is ip08406.

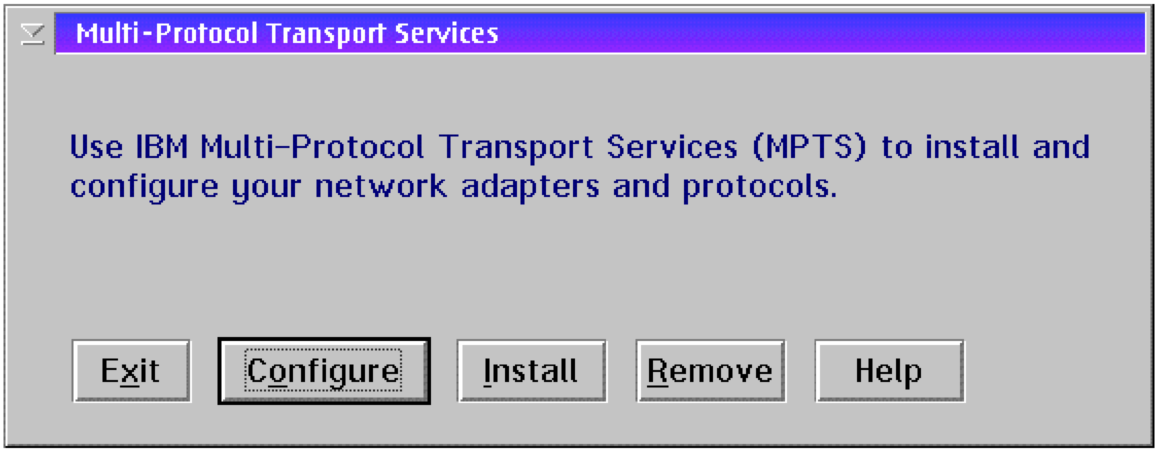
Step Two: Configure MPTS

The default transport installed for File and Printer Sharing is NetBIOS. We need to add another; NetBIOS over TCP/IP. Open the System Setup folder in your OS/2 System object on the desktop. In that folder, run the MPTS configuration program - double click on the Adapters and Protocol Services icon:



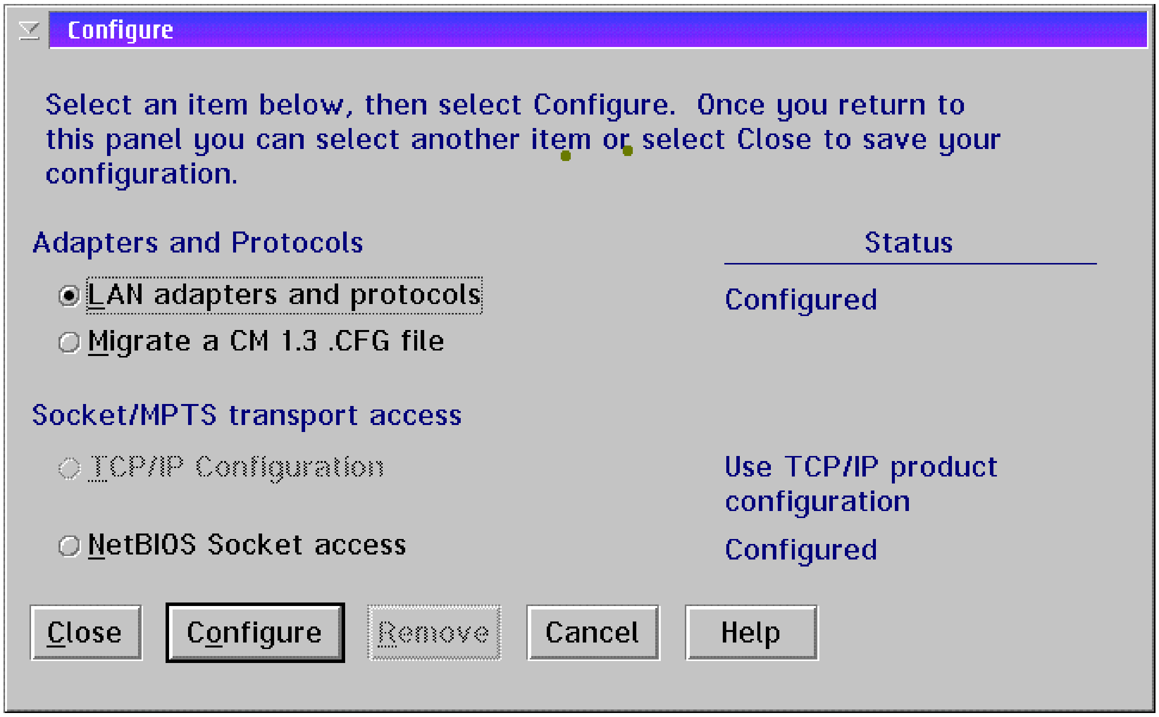
29 objects, 0 K bytes total. 1 selected, 0 K bytes.

You'll get the following dialog box - click on the Configure button:



Use IBM Multi-Protocol Transport Services (MPTS) to install and configure your network adapters and protocols.

Doing so will get you this dialog:



Configure

Select an item below, then select Configure. Once you return to this panel you can select another item or select Close to save your configuration.

Adapters and Protocols — Status

● LAN adapters and protocols — Configured

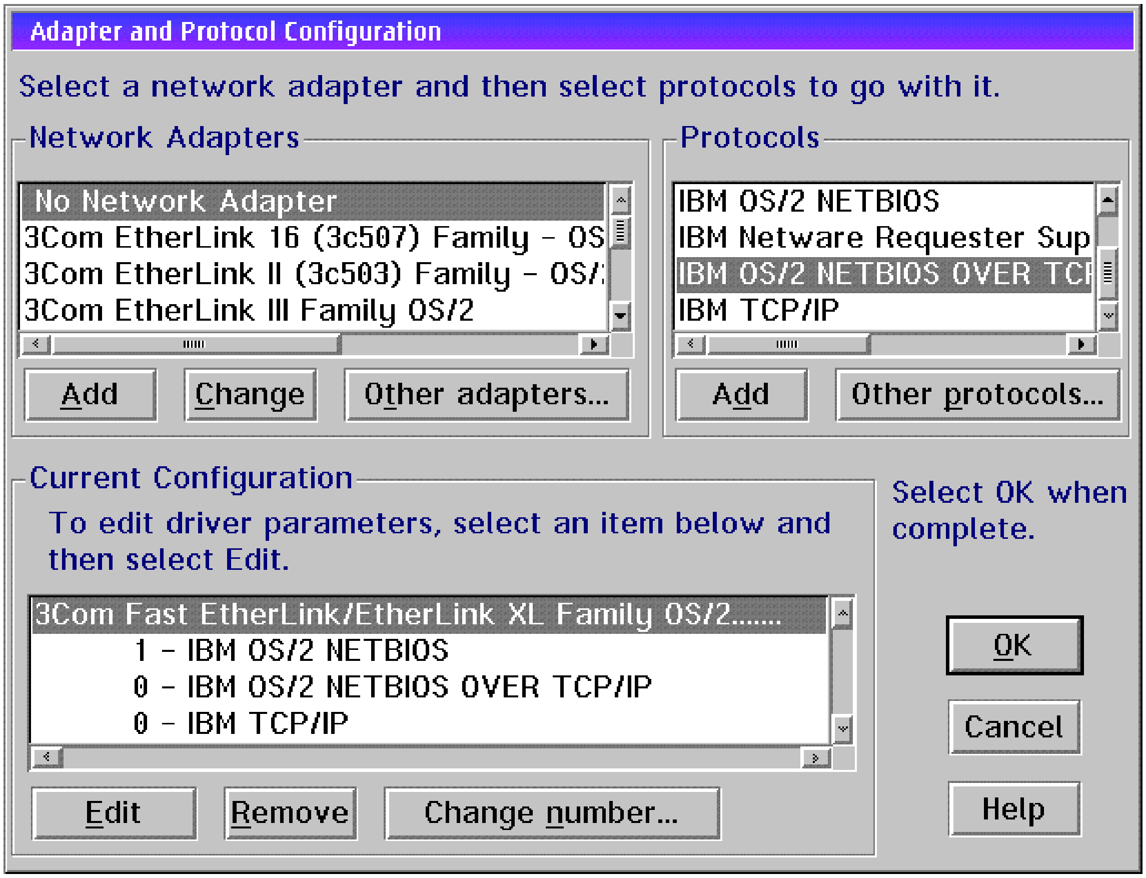
○ Migrate a CM 1.3 .CFG file

Socket/MPTS transport access

○ TCP/IP Configuration — Use TCP/IP product configuration

○ NetBIOS Socket access — Configured

Select the "LAN adapters and protocols" radio button and click on the Configure button, getting you to the Adapter and Protocol Configuration dialog:



Adapter and Protocol Configuration

Select a network adapter and then select protocols to go with it.

Network Adapters

```
No Network Adapter
3Com EtherLink 16 (3c507) Family - OS
3Com EtherLink II (3c503) Family - OS/:
3Com EtherLink III Family OS/2
```

Protocols

```
IBM OS/2 NETBIOS
IBM Netware Requester Sup
IBM OS/2 NETBIOS OVER TCI
IBM TCP/IP
```

Current Configuration

To edit driver parameters, select an item below and then select Edit.

```
3Com Fast EtherLink/EtherLink XL Family OS/2.......
          1 - IBM OS/2 NETBIOS
          0 - IBM OS/2 NETBIOS OVER TCP/IP
          0 - IBM TCP/IP
```

Select OK when complete.

When you first see this dialog, there will only be two Protocols listed in the bottom window: IBM TCP/IP and IBM OS/2 NETBIOS. They will both be listed with a leading "0 -" as the IBM TCP/IP is shown above. This leading number is the logical adapter number. The fact that a single physical adapter can be treated as multiple logical adapters is the reason for the "M" in MPTS. It is necessary because you can't have two approaches to NetBIOS on the same logical adapter; thus, you must use the "Change number" button after adding the IBM NetBIOS over TCP/IP protocol to the Current Configuration list.

According to the documentation, you shouldn't actually need to have the IBM OS/2 NetBIOS in the current configuration if you are not going to access devices on your local net. However, I found that I needed both protocols in the configuration -- although that may have been a consequence of other glitches that were eventually resolved. Unless you're trying to support 1000 connections (and you're running Warp Server), there's little lost by putting both in your configuration.

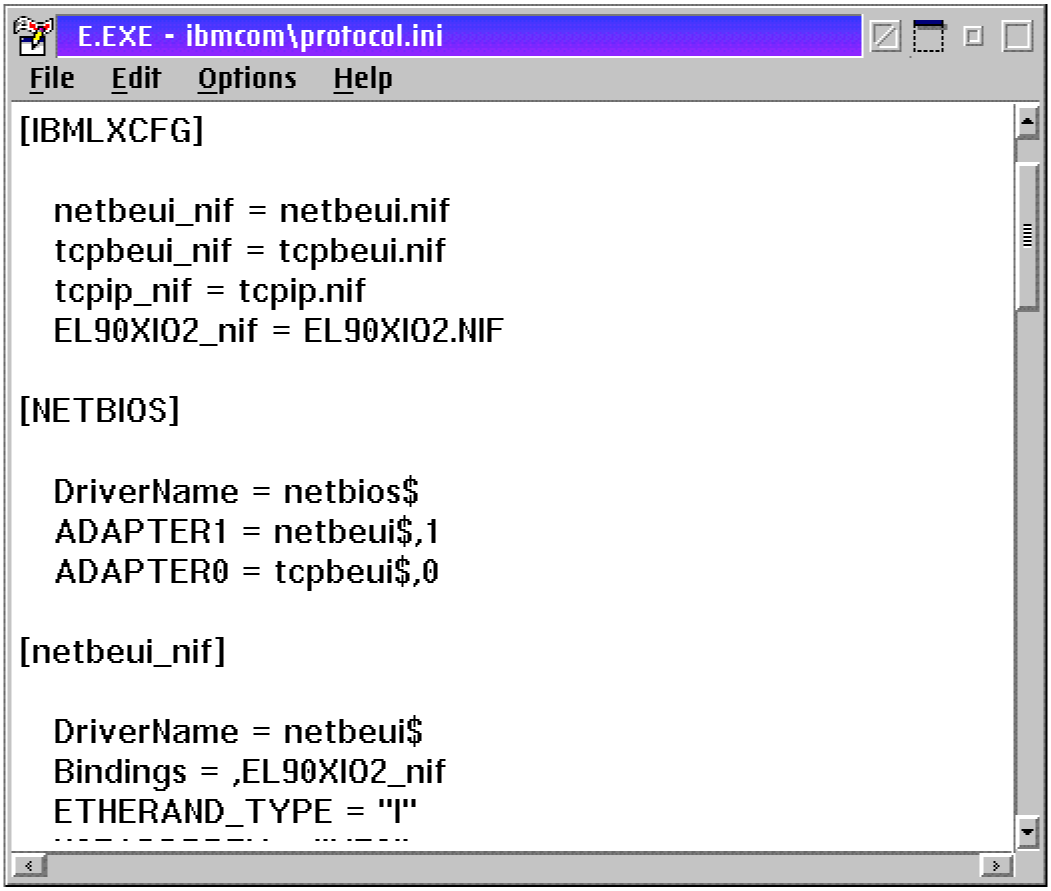
The configuration program won't let you close this dialog box until the two NetBIOS protocols are assigned to different logical adapters, so you can't get out of here without changing one of them. According to the documentation, it doesn't matter which is assigned to adapter 1 or 0; this is my configuration.

Once you have made these changes, select the OK, Close, and Exit buttons, respectively, to back out.

NOTE: The install/configuration program will tell you to reboot at this point - DO NOT REBOOT. There are a couple more steps to be taken before you reboot.

Step Three: Edit IBMLAN.INI

The MPTS Configure program should do this, but doesn't. Open the file \IBMCOM\PROTOCOL.INI, which can be found on your OS/2 boot drive. You should look for the ADAPTERn records (i.e., ADAPTER1= , ADAPTER2 =):



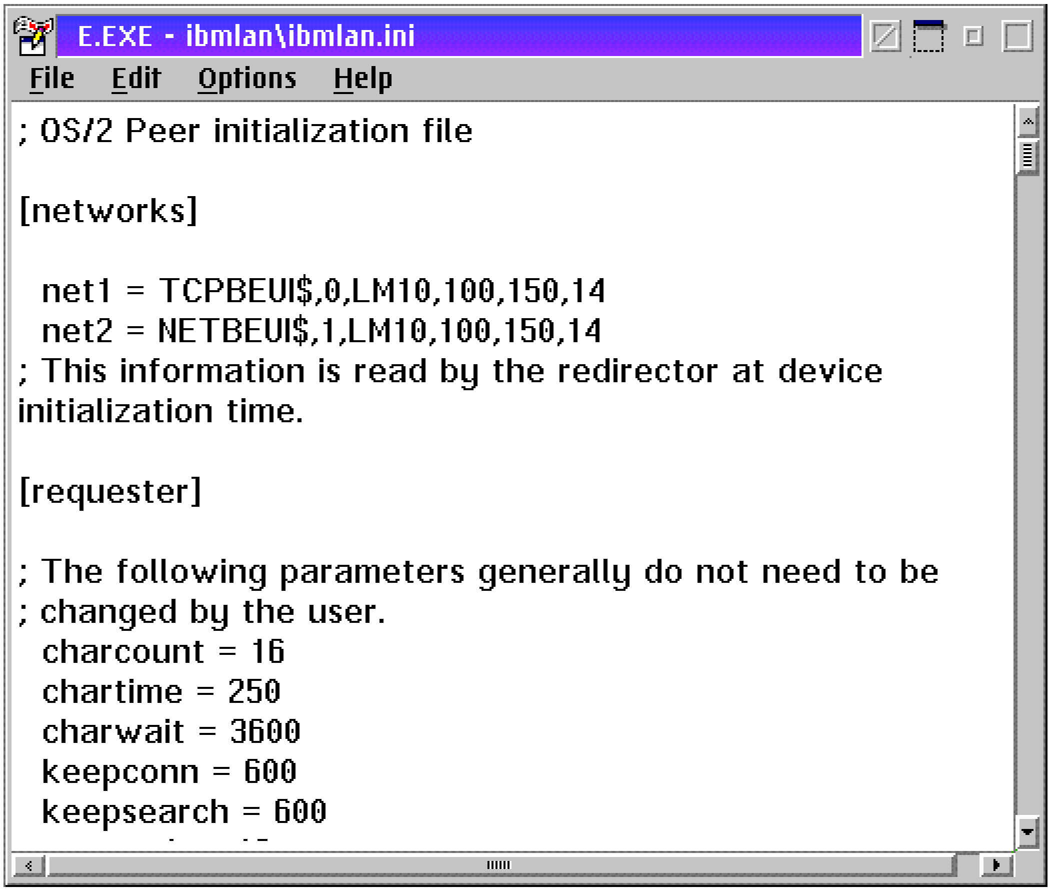
```
[IBMLXCFG]

  netbeui_nif = netbeui.nif
  tcpbeui_nif = tcpbeui.nif
  tcpip_nif = tcpip.nif
  EL90XIO2_nif = EL90XIO2.NIF

[NETBIOS]

  DriverName = netbios$
  ADAPTER1 = netbeui$,1
  ADAPTER0 = tcpbeui$,0

[netbeui_nif]

  DriverName = netbeui$
  Bindings = ,EL90XIO2_nif
  ETHERAND_TYPE = "I"
```

The key thing is to verify which adapter is associated with plain NetBIOS (ADAPTER1 = netbeui$) and which is associated with NetBIOS over TCP/IP (ADAPTER0 = tcpbeui$). These values should correspond with your choices made in the MPTS configuration program, but it pays to check.

You need this to edit the \IBMLAN\IBMLAN.INI file, on your OS/2 boot drive. For whatever reason, the MTPS configuration program will NOT update this file correctly.

When you open the IBMLAN.INI file, look at the beginning of the file for records like:

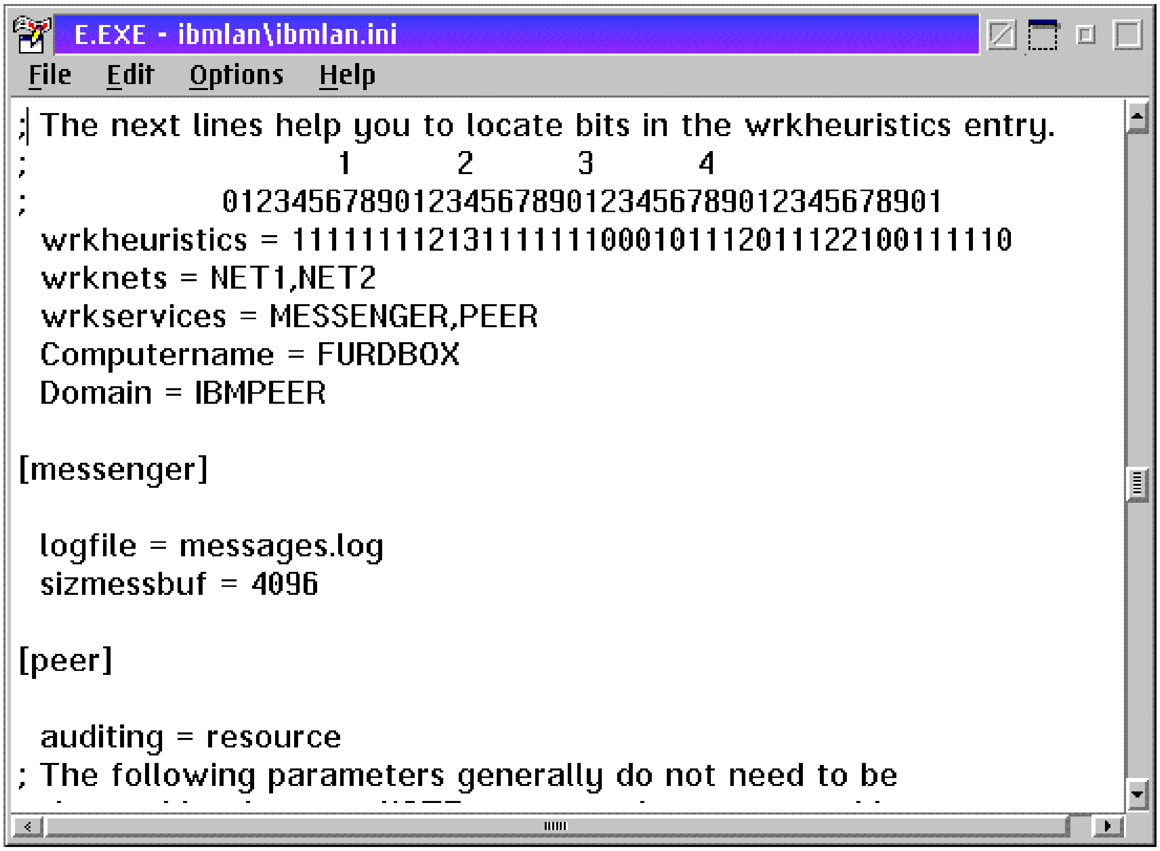
```
net1=NETBEUI$,0,LM10,100,150,14
```



```
; OS/2 Peer initialization file

[networks]

  net1 = TCPBEUI$,0,LM10,100,150,14
  net2 = NETBEUI$,1,LM10,100,150,14
; This information is read by the redirector at device
initialization time.

[requester]

; The following parameters generally do not need to be
; changed by the user.
  charcount = 16
  chartime = 250
  charwait = 3600
  keepconn = 600
  keepsearch = 600
```

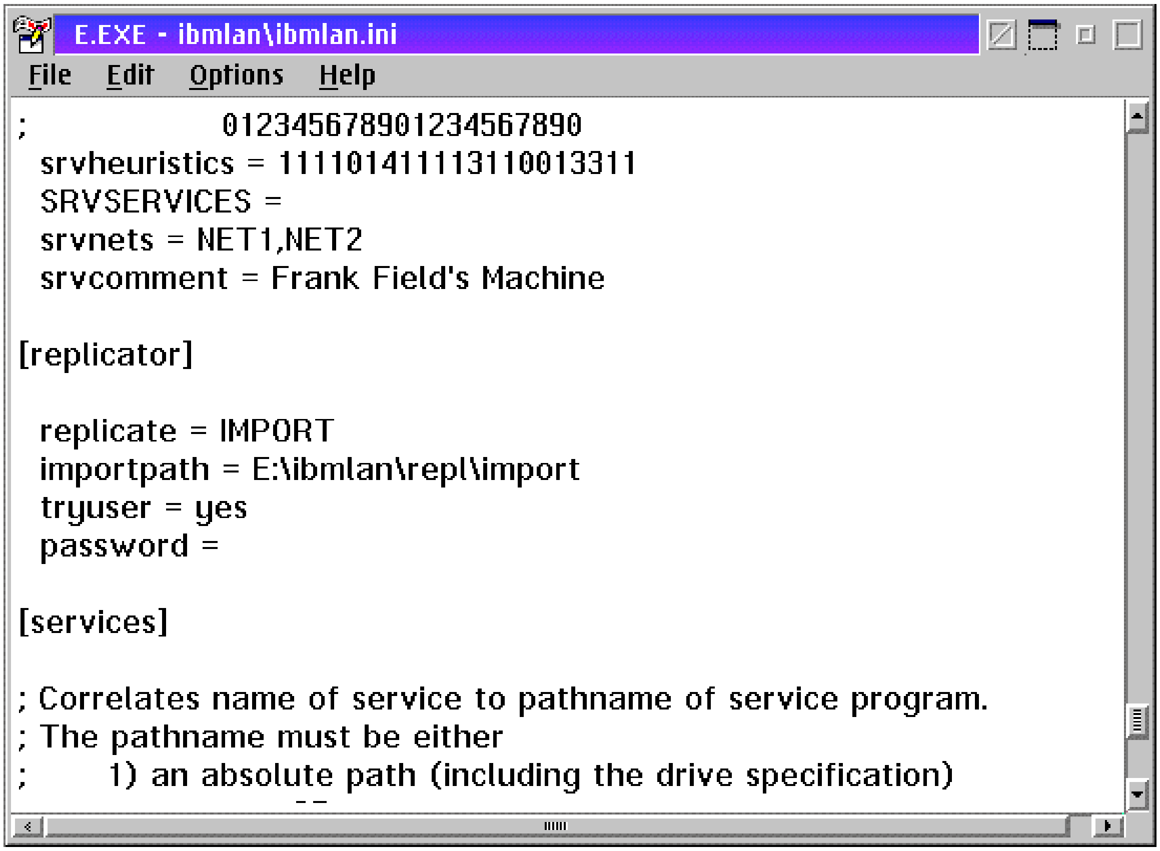
The odds are VERY good that you will only find only one of these records, but you must have two lines. One must refer to TCPBEUI$ and the other must refer to NETBEUI$, and the number immediately following must correspond with the logical adapter numbers selected in the MPTS configuration program (i.e., the net1= record must correspond with logical adapter 0, the net2= record must correspond with logical adapter 2, etc.). It's easy to add the second record; copy the existing record, and change the NETBEUI (or TCPBEUI) to the other, and get the adapter and net numbers right. Don't worry about the rest of the numbers in the record; just make sure you copy them exactly.

Next, search for a "wrknets =" record. You should find a record like "wrknets = NET1." You should change it so that it reads "wrknets = NET1,NET2" (like the following image). Again, don't worry about the other records; you just want to make sure to add the NET2 to the wrknets= record:



```
; The next lines help you to locate bits in the wrkheuristics entry.
;                          1         2         3         4
;                0123456789012345678901234567890123456789 01
  wrkheuristics = 111111112131111111000101112011122100111110
  wrknets = NET1,NET2
  wrkservices = MESSENGER,PEER
  Computername = FURDBOX
  Domain = IBMPEER

[messenger]

  logfile = messages.log
  sizmessbuf = 4096

[peer]

  auditing = resource
; The following parameters generally do not need to be
```

Finally, you need to search also for "srvnets =" -- and you'll do exactly the same thing: make sure that the "srvnets =" record refers to both NET1 and NET2 (i.e., srvnets = NET1,NET2).



```
;                012345678901234567890
  srvheuristics = 111101411113110013311
  SRVSERVICES =
  srvnets = NET1,NET2
  srvcomment = Frank Field's Machine

[replicator]

  replicate = IMPORT
  importpath = E:\ibmlan\repl\import
  tryuser = yes
  password =

[services]

; Correlates name of service to pathname of service program.
; The pathname must be either
;      1) an absolute path (including the drive specification)
```

Save this file.

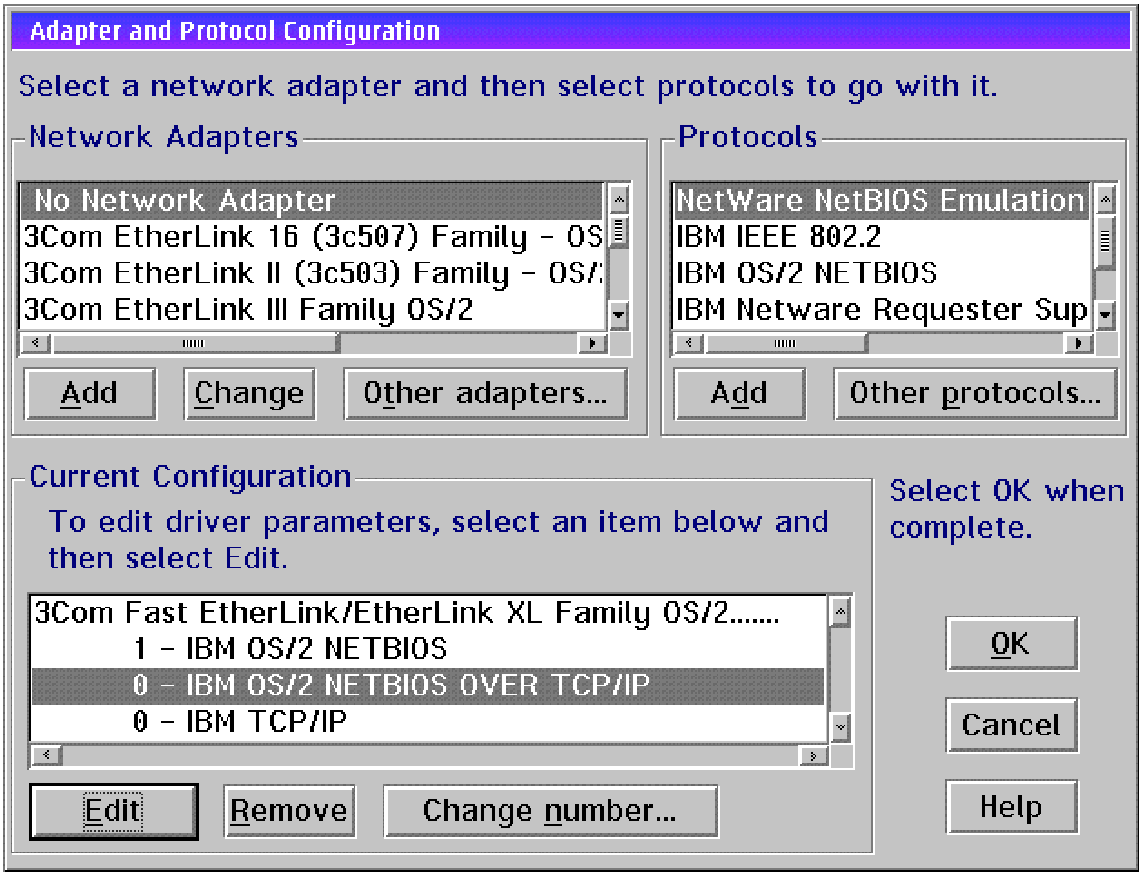
Step Four: Now You Can Reboot

You should get some startup messages describing the installation of TCPBEUI. If you get an error saying that the "NETWKSTA.200" is not a device driver, then you need to start this all over again. Keep slogging and make sure that you got all the NETn and protocol adapter numbers right. Eventually, you should get it all to boot correctly. And, when you do, you'll be able to get started on setting up your Peer work.

Step Five: Identifying Your Peer Machines

Now that you have a NetBIOS over TCP/IP setup, how to get your machine to talk to machines not on your local network. Basically, you have to tell your machine how to map an 8-character NetBIOS name to an Internet machine name. This is accomplished with two files stored in your \IBMCOM directory: RFCNAMES.LST and RFCBCST.LST. They can be edited via the MPTS configuration program (see step two) or they can be edited from any text editor.

Editing via MPTS requires you to restart the MPTS configuration utility and get back to the Adapter and Protocol Configuration dialog. Select the NetBIOS over TCP/IP protocol in the bottom window and click on the Edit button.



Adapter and Protocol Configuration

Select a network adapter and then select protocols to go with it.

Network Adapters

No Network Adapter
3Com EtherLink 16 (3c507) Family - OS
3Com EtherLink II (3c503) Family - OS/:
3Com EtherLink III Family OS/2

Protocols

NetWare NetBIOS Emulation
IBM IEEE 802.2
IBM OS/2 NETBIOS
IBM Netware Requester Sup

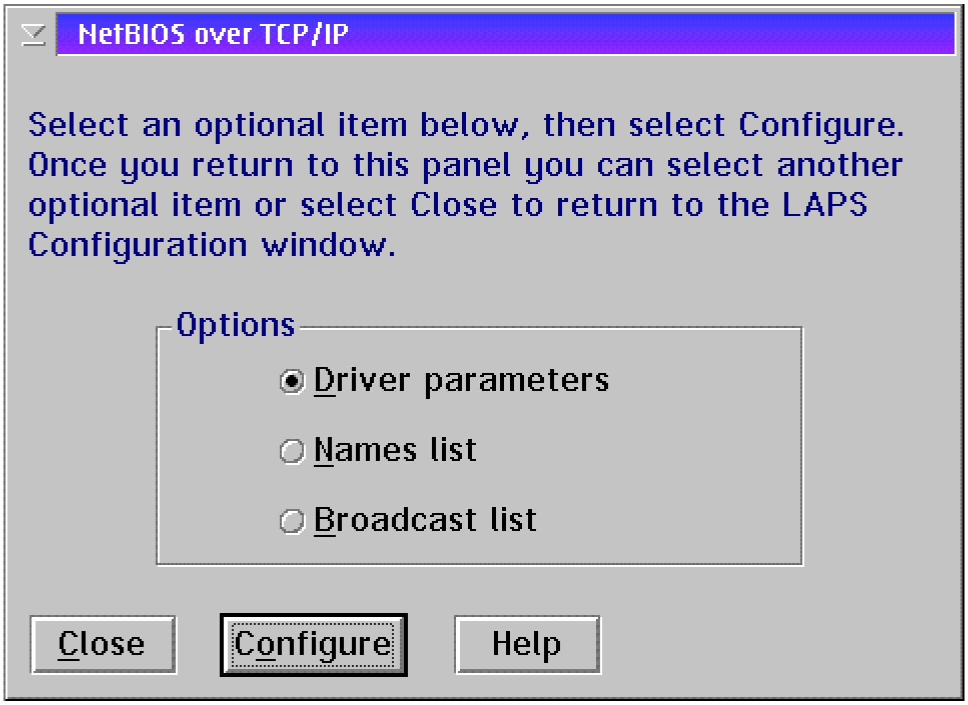
Current Configuration

To edit driver parameters, select an item below and then select Edit.

3Com Fast EtherLink/EtherLink XL Family OS/2.......
1 - IBM OS/2 NETBIOS
0 - IBM OS/2 NETBIOS OVER TCP/IP
0 - IBM TCP/IP

Select OK when complete.

Clicking on Edit yields a choice of editing Driver parameters, Names list, and Broadcast list:



NetBIOS over TCP/IP

Select an optional item below, then select Configure. Once you return to this panel you can select another optional item or select Close to return to the LAPS Configuration window.

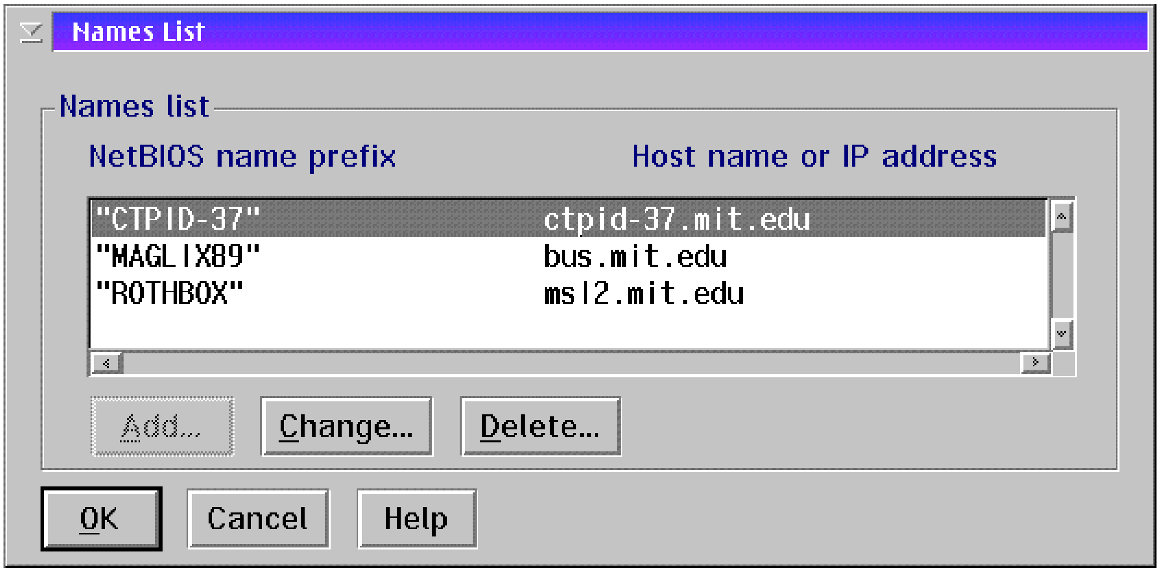
Options

● Driver parameters

○ Names list

○ Broadcast list

Selecting the Names List radio button and clicking on Configure yields a listbox:



Names List

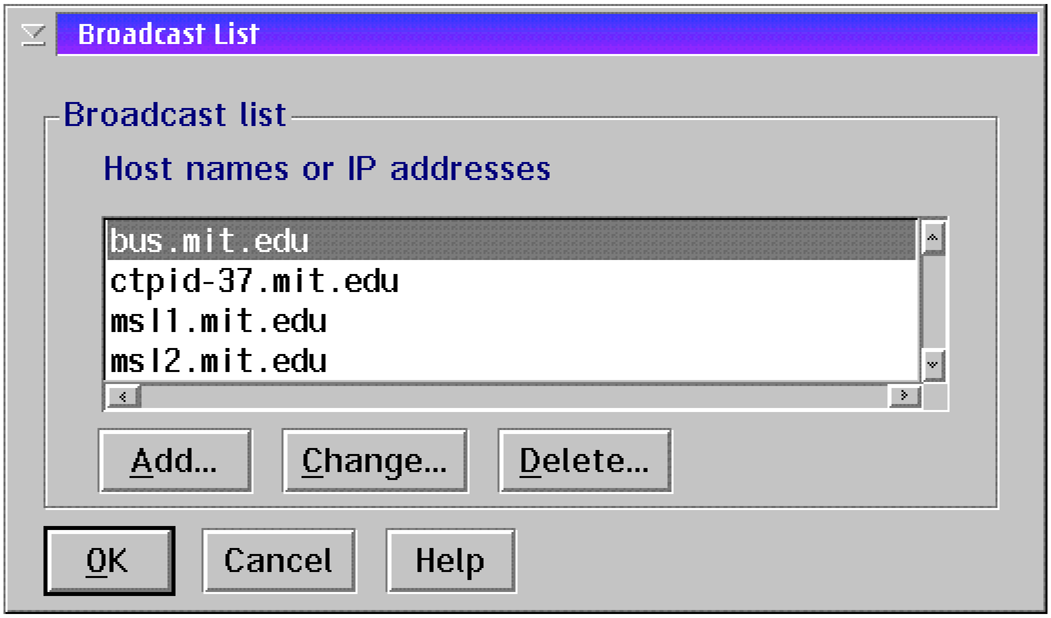
Names list

```
NetBIOS name prefix              Host name or IP address
"CTPID-37"                       ctpid-37.mit.edu
"MAGLIX89"                       bus.mit.edu
"ROTHBOX"                        msl2.mit.edu
```

Note that the Add... button in this image is greyed out - this is a known bug in many incarnations of the MPTS configuration program. You will always be able to add at least one NetBios name/hostname pair to this list; however, if your version of MPTS greys out the Add... button after adding the first name, you will have to edit the RFCNAMES.LST file by hand. The default protocol setup for NetBIOS over TCP/IP allows you to specify up to four names. If you can't get those names input via MPTS setup, any text editor can be used (see below).

Basically, this is where you tell your machine the IP address of the NetBIOS machine names that you want to access. This figure shows all the addresses as host names, but you can equally well use IP addresses like 18.178.0.30. The limit of four is defined in the protocol setup. If you really need more, then it's time to agitate for a NetBIOS nameserver. You may be able to use a WINS server, but I don't know anyone yet who has. (See RoknRob's web site for more information on WINS configuration.)

You also need to configure the Broadcast List. When you select that radio button and then click on Configure, you get:

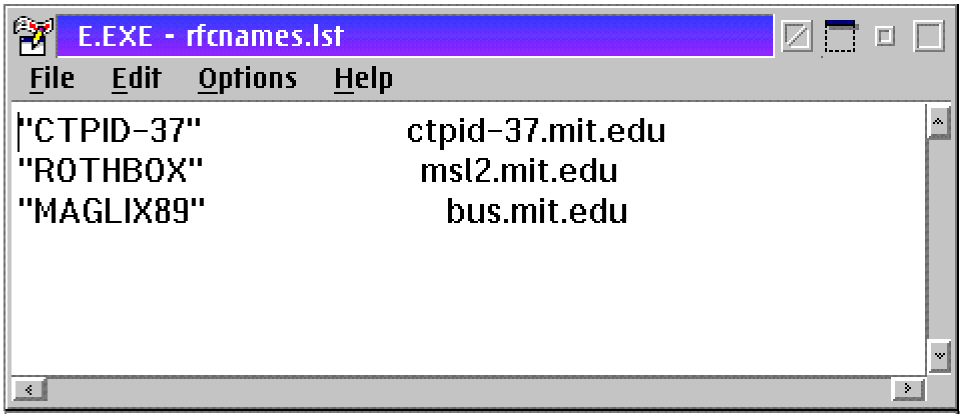


```
bus.mit.edu
ctpid-37.mit.edu
msl1.mit.edu
msl2.mit.edu
```

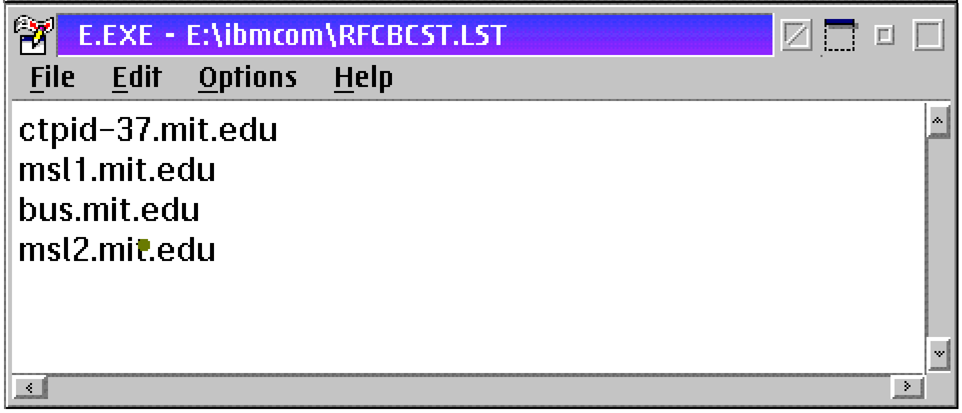
The NetBIOS protocol, as a non-routable protocol, only knows how to achieve certain tasks by basically sending a message to every machine it can find (broadcasting). As you might imagine, this is something of a problem for big networks, hence the need for nameservers. You need to put the same machines in this list that you put in the names list. Again, either hostnames or IP addresses are acceptable.

After you exit from this program, you may be told to reboot. That isn't necessary if all you've done is modify these two lists. Just go to the \IBMCOM directory and run RFCADDR from the command line: this registers the lists with the currently running peer daemon, so you're all set.

You can also edit RFCNAMES.LST and RFCBCST.LST with any text editor. The file format is straightforward: RFCNAMES.LST is composed of records that pair a NetBIOS name, inside of double quotes, with an IP hostname, and RFCBCST is just a list of hostnames:



```
"CTPID-37"              ctpid-37.mit.edu
"ROTHBOX"                msl2.mit.edu
"MAGLIX89"                bus.mit.edu
```



```
ctpid-37.mit.edu
msl1.mit.edu
bus.mit.edu
msl2.mit.edu
```

The only tricks are (a) to make sure that the NetBIOS names in the RFCNAMES.LST file are inside of double quotes; (b) put a space (or several) between the NetBIOS name and the hostname/IP address; and (c) make sure that every hostname/IP address that appears in the RFCNAMES.LST file also appears in the RFCBCST.LST file. You can edit these files at any time, but any changes you make will not be recognized by the peer daemon processes until you run RFCADDR after saving these files.

Done! Your machine is now peered to the MITNet. If you want to drive someone crazy, you can even use the Network Messaging tool to send messages to those machines whose names you have registered - and they'll be unable to get back to you if they're on an NT machine. To get those NT machines to recognize you, you'll have to get the users on the other end to make some changes to their machines.

Getting The NT Machines To Behave

There are three rules that the NT machines you wish to peer with must follow:

1. NT machines have an RFCNAMES.LST file, too. Of course, its name is different. RFCNAMES.LST on an NT machine is LMHOSTS (see http://publibfp.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/EZ30NZ00/1.3.14.1?DT=19970121125126). You also may want to register your machine name with the NT machine by adding its hostname and IP address to the HOSTS file (a sort of belt and suspenders approach to name resolution). LMHOSTS serves the same purpose as RFCNAMES.LST - it tells the NT machine where to find your NetBIOS-named machine when you aren't on the same subnet.

   Note: Searching for LMHOSTS and HOSTS files on a NT machine may be difficult. You may not be able to find those specific files; if not, you will find files called LMHOSTS.SAM and/or HOSTS.SAM (with the SAM being short for "sample.") The files have enough comments in them so you should be able to set them up correctly. But, you will have to save them without the .SAM file extension.

2. NT machines on the same subnet as you will also be invisible to your GUI resource browsing tools. There is a default registry setting - lmannounce - that has to be changed from the NT default of 0 to 1. Then, the NT machines will be visible to you. See RoknRob's WWW page for the nitty gritty, because each Windows variant has a different way to get the Windows machine to respond consistently with the NetBIOS LAN communication defaults.

3. NT resources must have NetBIOS names that are less than 9 characters in length. In this incarnation of Warp, longer names will confuse the graphical networking tools and you won't be able to use them to connect to resources. The command-line NET command will do the best it can and you can employ NET USE to connect to resources with legal names. However, NET VIEW will not necessarily be able to list everything - you'll get the extremely useless error message - More Data Available - but you won't get to see it.

Finally, NT permission setting is, without a doubt, the most confusing thing I've tried to deal with yet - and, as I'm sure you'll agree with me up to this point, LOTS of this is confusing. If you're working with someone who understands it, let them take care of it for you. If not, the IBM Redpiece on NT Administration is of some use, but I have some real gaps in my understanding.

On one hand, if the resource is something that you don't want to protect, it's pretty straightforward to give EVERYONE on the net access to the resource. Great, but not particularly attractive if you're trying to share work files. But giving password-restricted access to a peer resource is not trivial since every NT machine that you want to share with must have a local user registered on each NT machine that has the same USERID and PASSWORD; and that must match the USERID and PASSWORD that you use to log onto the network! This seems too insane to be true, but it's the only way that I've been able to set up password-restricted access to peer resources.