Protocol Manager Binding: NETBIND and the PROTOCOL.INI File

A Supplement to the Network Driver Interface Specification (NDIS)
Version 1.0.2

3165 Kifer Road
Santa Clara, California 95052-8145
Printed in the U.S.A.

Manual Part No. 6404-00
Published Month 1988

Disclaimer

3Com makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. 3Com shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Contents

Tables

Introduction

The PROTOCOL.INI file contains information about configuration and binding for all NDIS protocols and Media Access Control (MAC) modules.

Network drivers in the DOS and OS/2 LAN Manager products are loaded and configured with entries in the CONFIG.SYS (CONFIG.OS2 for dual boot systems) and PROTOCOL.INI files. When the LAN Manager starts up, the Protocol Manager program reads the PROTOCOL.INI file and binds the protocol and MAC modules together using the parameters and information supplied in the PROTOCOL.INI file.

This process has two parts. During the initial loading and starting of the operating system, the device drivers named in the CONFIG.SYS file are loaded and their initialization routines are executed. Among these device drivers are the protocol manager (PROTMAN.OS2 or PROTMAN.DOS) and both NDIS/MAC drivers and NDIS/protocol drivers. The protocol manager, as part of its initialization, reads the PROTOCOL.INI file and saves its contents.

The second part of the process occurs during run-time, when the program NETBIND.EXE is run either from the command line or from a batch file such as AUTOEXEC.BAT or STARTUP.CMD. At this time, the NETBIND program causes the protocol manager to call each NDIS module (MACs and protocols), pass each the PROTOCOL.INI contents, and bind all the modules together as specified within PROTOCOL.INI.

Changes you make to PROTOCOL.INI entries do not take effect until you restart the system.

Syntax of the PROTOCOL.INI File

The PROTOCOL.INI file consists of a series of named sections. The format for entries in the PROTOCOL.INI file is shown in the following example:

```
[ETHERLINK_PLUS]
  DRIVERNAME = ELNKPL$
  INTERRUPT = 3
  IOADDRESS = 0x300
  DMACHANNEL = 1
; Be sure the interrupt does not conflict with other adapter cards.
```

In this example:

- Names of sections are enclosed in square brackets [ETHERLINK_PLUS]. All sections contain the keyword **drivername**, (which identifies the driver). In addition, most sections have the keyword **bindings** (which serves to connect the various modules together). The section name itself is an arbitrary label that identifies the section to the other sections (for example, when resolving bindings).

- Entries in sections start at the beginning of a line and are followed by an equal sign (=) and the value assigned to the entry. There can be spaces before and/or after the equal sign.

- Comment text (text that is ignored by the system), starts with a semicolon at the beginning of each line. Blank lines are permitted for formatting purposes.

- Text in the PROTOCOL.INI file is changed to uppercase by the Protocol Manager, with the exception of comment text. Unless otherwise specified, text values can be composed of letters, numbers, or the following characters:

  $ % ; - _ @ { } ~ ‘ ! # ( )

What the PROTOCOL.INI File Contains

The PROTOCOL.INI file consists of different sections containing entries related configuring network drivers. The following table shows (in alphabetical order) typical PROTOCOL.INI sections and their functions. (Remember, section names are arbitrary so your PROTOCOL.INI file may be different.)

Table 1. PROTOCOL.INI File Sections Summary

Section	Function
EtherLink	Configuration values for a 3Com EtherLink adapter.
EtherLink II	Configuration values for a 3Com EtherLink II adapter.
EtherLink Plus	Configuration values for a 3Com EtherLink Plus adapter.
EtherLink/MC	Configuration values for a 3Com EtherLink/MC adapter.
TokenLink	Configuration values for a 3Com TokenLink adapter.
TokenLink Plus	Configuration values for a 3Com TokenLink Plus adapter.
3Server	Driver for 3Com 3Servers.

3Com EtherLink Adapter Section

The EtherLink driver has one required entry, DRIVERNAME. The following table shows (in alphabetical order), the possible entries and values for the EtherLink adapter section of the PROTOCOL.INI file.

Table 2. EtherLink Adapter Driver Entries

Entry	Explanation of Option Values
DMACHANNEL=	Used to indicate the current DMA jumper configuration of the adapter. NONE = Driver uses program I/O instead of DMA for data transfers. 1 = Use DMA channel 1. 2 = Use DMA channel 2. 3 = Use DMA channel 3. Default = 1 When running under DOS. Default = NONE When running under OS/2.
DMAMODE=	Specifies the DMA mode for the adapter. BURST = Use BURST mode. BYTE = Use BYTE mode. Default = BURST. The Adapter runs faster in BURST mode. If BURST mode causes contention with other DMA devices, use BYTE mode. NOTE: If DMACHANNEL is set to NONE, the DMAMODE entry is ignored.
DRIVERNAME=	ELNK$ (required entry). If there is a second EtherLink adapter, its name is ELINK2$.
INTERRUPT=	Indicates the current interrupt level configuration of the adapter. Available = 2 - 7. Default = 3. NOTE: On a 286 or 386 computer, interrupt level 2 = interrupt level 9 because of the cascaded PIC already using interrupt level 2.
IOADDRESS=	Indicates the current I/O address jumper configuration of the adapter. Available = 0x200 to 0x3f0 Default = 0x300 NOTE: The IOADDRESS entry *must* match the switch setting on the adapter card.

Table 2. EtherLink Adapter Driver Entries, cont.

Entry	Explanation of Option Values
MAXREQUESTS=	This is the number of general request queue entries. (General requests are SetPacketFilter, AddMultiCastAddress, and RequestInterrupt, etc.) Available = 8 to 32. Default = 8.
MAXTRANSMITS=	The number of transmit queue entries in the MAC driver. Available = 8 to 50. Default = 12.
NETADDRESS=	Used to override the network address of the card to a user specific address. This parameter is a hexadecimal string of exactly 12 digits, enclosed by double quotes (""), for example: NETADDRESS = "02608C010001".

3Com EtherLink II Adapter Section

The EtherLink II driver has one required entry, DRIVERNAME. The EtherLink II Adapter does not implement the LOOPBACK function, so you must add a LOOPBACK = 4 entry to the DLC section when using this adapter. The following table shows the possible entries and values for the EtherLink II adapter section of the PROTOCOL.INI file.

Table 3. EtherLink II Adapter Driver Entries

Entry	Explanation of Option Values
DATATRANSFER=	Sets the data transfer mode on the adapter card. Keywords are: BLOCK_DMA Only valid for 8086 and 8088 PCs.
DEMAND_DMA	Only valid for 8086 and 8088 PCs. If the DMACHANNEL entry is set to 1, the default for DATATRANSFER becomes DEMAND_DMA. PIO_BYTE For 80286 and 80386 computers only. PIO_WORD For 80286 and 80386 computers only. SINGLE_DMA Only valid for 8086 and 8088 PCs. **NOTE:** On an 80286 or 80386 PC, the DMACHANNEL keyword is ignored and the default becomes PIO_WORD.

Table 3. EtherLink II Adapter Driver Entries, cont.

Entry	Explanation of Option Values
DMACHANNEL=	Used to indicate the current DMA jumper configuration of the adapter. NONE = Driver uses program I/O instead of DMA for data transfers. 1 = Use DMA channel 1. 2 = Use DMA channel 2. 3 = Use DMA channel 3. Default = 1 When running under DOS. Default = NONE When running under OS/2.
DRIVERNAME=	ELNKII$ (required entry). If there is a second EtherLink II driver, its name is ELNKII2$.
INTERRUPT=	Sets the interrupt level for the adapter card. Available = 2-5. Default = 3. On a 286 or 386 computer, interrupt level 2 = interrupt level 9 because of the cascaded PIC already using interrupt level 2.
IOADDRESS=	Indicates the current I/O address jumper configuration of the adapter. Available = 0x200 to 0x3F0 Default = 0x300 **NOTE:** The IOADDRESS entry *must* match the switch setting on the adapter card.
MAXTRANSMITS=	The number of transmit queue entries in the MAC driver. Available = 8 to 50. Default = 8.
NETADDRESS=	Overrides the network address of the card to a user specific address. This parameter is a hexadecimal string of 12 digits in double quotes, such as NETADDRESS = "02608C010001".
TRANSCEIVER=	Sets the transceiver on the adapter card. EXTERNAL. Used for a DIX connection. ONBOARD. Default = ONBOARD.

3Com EtherLink Plus Adapter Section

The EtherLink Plus driver has one required entry, DRIVERNAME. The following table shows (in alphabetical order), the possible entries and values for the EtherLink Plus adapter section of the PROTOCOL.INI file.

Table 4. EtherLink Plus Adapter Driver Entries

Entry	Explanation of Option Values
DMACHANNEL=	Sets the DMA channel for the card. Available = 1-7 Default = 3 NOTE: The DMACHANNEL entry *must* match the jumper setting on the adapter card.
DRIVERNAME=	ELNKPL$ (required entry). If there is a second EtherLink Plus adapter, its name is ELNKPL2$.*
INTERRUPT=	Sets the interrupt level for the adapter card. Available = 2-15 Default = 3 On a 286 or 386 computer, interrupt level 2 is really interrupt level 9 because of the cascaded PIC already using interrupt level 2. NOTE: The INTERRUPT entry *must* match the jumper setting on the adapter card.
IOADDRESS=	Indicates the current I/O address jumper configuration of the adapter. 0 to 0x3F0 Default = 0x300 NOTE: The IOADDRESS entry *must* match the jumper setting on the adapter card.
MAXTRANSMITS=	The number of transmit queue entries in the MAC driver.* Available = 8 to 50 Default = 8.
NETADDRESS=	Overrides the network address of the card to a user specific address. This parameter is a hexadecimal string of 12 digits in double quotes, such as NETADDRESS = "02608C010001".

* Not supported in this release (3+Open 1.0.)

3Com EtherLink/MC Adapter Section

The EtherLink/MC driver has one required entry, DRIVERNAME. The following table shows (in alphabetical order), the possible entries and values for the EtherLink Plus adapter section of the PROTOCOL.INI file.

Table 5. EtherLink/MC Adapter Driver Entries

Entry	Explanation of Option Values
DRIVERNAME=	ELNKMC$ (required entry). If there is a second EtherLink/MC, its name is ELNKMC2$.
MAXTRANSMITS=	The number of transmit queue entries in the MAC driver. Available = 8 to 50 Default = 8.
NETADDRESS=	Overrides the network address of the card to a user specific address. This parameter is a hexadecimal string of 12 digits in double quotes, such as NETADDRESS = "02608C010001".
SLOTNUMBER=	Specifies the slot number in the computer within which the card resides. Available = 1-8 Default = The lowest numbered slot containing an adapter of this type. Use this parameter to override the default of the lowest numbered slot. For example, if you are using two adapter cards, both cannot be lowest numbered so you must specify this keyword for at least one of them.

3Com TokenLink Adapter Section

The TokenLink driver has one required entry, DRIVERNAME. The following table shows (in alphabetical order), the possible entries and values for the TokenLink adapter section of the PROTOCOL.INI file.

Table 6. TokenLink Adapter Driver Entries

Entry	Explanation of Option Values
DRIVERNAME=	TLNK$ (required entry)> If there is a second TokenLink adapter, its name is TLNK2$
NETADDRESS=	Overrides the network address of the card to a user specific address. This parameter is a hexadecimal string of 12 digits in double quotes (""), for example: NETADDRESS = "02608C11AA0C".
IOADDRESS=	Sets the (unique) I/O address of the card. 0x300 to 0x3F0 Default = 0x300 **NOTE:** This address *must* be the same as the address set on the adapter card.
DMACHANNEL=	Sets the DMA channel on the adapter card. 0 Use string I/O move instructions to transfer data from adapter cards to memory. An 8086 or 8088 computer uses programmed I/O. 1 Forces a 80286 or 80386 computer to use bus master mode. If bus master mode is not supported by your computer, string I/O move instructions are used instead. 2 Use DMA channel 2. 3 Forces a 80286 or 80386 computer to use bus master mode 4-7 Use the specified DMA channel. Default = 1.
INTERRUPT=	Sets the interrupt level for the adapter card. Available = 2-5 Default = 3. **NOTE:** On a 286 or a 386 computer, interrupt level 2 is really interrupt level 9 because of the cascaded PIC already using interrupt level 2.

Table 6 is continued on the next page.

Table 6. TokenLink Adapter Driver Entries, cont.

Entry	Explanation of Option Values
SPEED=	Sets the clock on the host computer to maximize data transfer between the host computer and the adapter card. 6 Inserts two wait states on the bus. 8 Inserts 1 wait state on the bus. 10 Does not insert any wait states on the bus. Default = 6. NOTE: Some experimentation may be required for best performance. If this parameter is set incorrectly, the adapter card may not run.
MAXTRANSMITS=	Specifies the number of transmit queue entries in the MAC driver. 6 to 50. Default = 6.

3Server Driver Section

The 3Server driver has one required entry, DRIVERNAME. The following table shows (in alphabetical order), the possible entries and values for the 3Server driver section of the PROTOCOL.INI file.

Table 7. 3Server Driver Entries

Entry	Explanation of Option Values
DRIVERNAME=	E3S386$ (Required entry)
NETADDRESS=	Overrides the network address of the card to a user specific address. This parameter is a hexadecimal string of 12 digits in double quotes (“”), for example: NETADDRESS = “02608C010001”.
RECVBUFS=	Configures the receive buffers for the driver. Available = 15-43 Default = 32.

A Sample PROTOCOL.INI File

The following is a sample PROTOCOL.INI file for a 3Com server. This file is actually a template for the PROTOCOL.INI file the installation process creates. The installation program takes this template, modifies it according to the particular installation choices specified, then creates the PROTOCOL.INI file. Therefore the exact contents of any particular PROTOCOL.INI file depend on the protocols and adapters specified during the installation process. Protocol sections labeled DLC, NETBEUI, and XNS_NB have been added for completeness.

```
[TOKENLINK]
        DRIVERNAME = TLNK$
        INTERRUPT = 3
        IOADDRESS = 0x300
        DMACHANNEL = 1

[ETHERLINK]
        DRIVERNAME = ELNK$
        IOADDRESS = 0x300
        DMACHANNEL = 1
        INTERRUPT = 3

[ETHERLINK_PLUS]
        DRIVERNAME = ELNKPL$
        IOADDRESS = 0x300
        DMACHANNEL = 1
        INTERRUPT = 3

[ETHERLINKII]
        DRIVERNAME = ELNKII$
        INTERRUPT = 3
        IOADDRESS = 0x300
        DMACHANNEL = 1
        XMITBUFS = 1

[ETHERLINK/MC]
        DRIVERNAME = ELNKMC$

[3SERVER]
        DRIVERNAME = E3S386$
        RECVBUFS = 32

[XNS_NB]
        DRIVERNAME = XNSNB$
        BINDINGS =
        MAXSESS = 32
        MAXNCB = 32
        MAXDGBUF = 2
```

```
        MAXSNBUF = 8
        MAXRECVBUF = 12
        RECVWINDOW =
        XMITWINDOW =
        MAXCACHE = 2
        MAXREMMAP = 2
        MAXLCLMAP = 2
        STOREFORWARD = NO
        HARDWAREBUF = NO
        LOCATOR = NO
        ROUTE1 =
        ROUTE2 =
        ROUTE3 =
        ROUTE4 =
        ROUTE5 =
        ROUTE6 =
        ROUTE7 =
        ROUTE8 =

[DLC]
        DRIVERNAME=MDLC$
        BINDINGS=

[NETBEUI]
        DRIVERNAME=MNETB$
        BINDINGS=DLC
```