Intel Network Drivers for DOS

DOS Driver

The NDIS2 (DOS) driver is provided solely for the purpose of loading other operating systems -- for example, during RIS or unattended installations. It is not intended as a high-performance driver.

You can find adapter drivers, PROTOCOL.INI files, and NET.CFG files in the PRO100\DOS or PRO1000\DOS directory on the Product CD or download folder. For additional unattended install information, see the text files in the operating system subdirectories under the APPS\SETUP\PUSH directory.

Automatic or Explicit Configuration of a Single NIC or Multiple NICs

When the driver finds that only one adapter is installed in the system, it will use that adapter regardless of whether or not parameters in PROTOCOL.INI are present or correct. If the parameters do not match the actual configuration, the driver will display warning messages indicating that the parameter was not used.

One instance of the driver must be loaded for each adapter that is activated. When multiple adapters are installed, the SLOT parameter becomes advisable but not required.

The determination as to which adapter each driver will control should be made by the user based on the protocol stack(s) bound to each driver, and based on the network that is connected to each adapter. The “BINDINGS” list in each protocol stack’s PROTOCOL.INI section establishes the relationship between protocol stacks and drivers. The SLOT parameter in the driver’s PROTOCOL.INI section establishes the relationship between drivers and adapters, and a value can be provided for each driver loaded. If a SLOT parameter is not specified, the first driver instance will load on the first NIC/Port found in the scanning list, the second driver instance will load on the second NIC/Port found in the scanning list, etc. When the driver detects multiple NICs/Ports it will report all of the possible slots. The only way for the driver to know which driver instance is being loaded is to use the DRIVERNAME parameter instance number. Therefore, it is essential that the DRIVERNAME parameter instance syntax defined below be used correctly.

The adapters are automatically configured by the PCI system BIOS when the system boots. The driver queries the PCI BIOS and obtains all of the adapter’s configuration information. BIOS scanning using mechanisms 1 and 2, as defined in the PCI BIOS specification, are supported. The SLOT number is actually the encoded value of the PCI adapter’s device location, which is defined as shown below. The SLOT value reported by the driver and entered by the user is the value of bits 0 through 15. In versions of the driver prior to 2.01, the SLOT value reported by the driver and entered by the user was shifted right by 3 bits (divided by 8) so that SLOT 0x0088 was actually entered into PROTOCOL.INI as 0x0011. This doesn’t allow for multi-function devices to be specified with this SLOT parameter. So starting with v2.01, the driver does not shift the input parameter by 3 bits and SLOT 0x0088 would be entered as 0x0088. This also allows for specifying slot 0x0081 = Bus 0 Device 16 Function 1. If the driver finds that the entered SLOT number is not found in its slot list table, it may be because the SLOT uses the older convention (shifted right). The driver then tries to match this old style slot parameter to a slot in the slot list and loads on that slot if it finds a match. This is done for backward compatibility.

Configuring with the PROTOCOL.INI File

The configuration parameters listed below are supported through the PROTOCOL.INI file. When the machine has a single adapter, all the parameters (except DRIVERNAME) are optional; when the machine has multiple adapters, some of the parameters are required.

DRIVERNAME

This is the only parameter required for all configurations. This parameter is essentially an "instance ID". Each instance of the driver must create a unique instance name, both to satisfy DOS driver requirements, and to make it possible to find the parameters for the instance in the PROTOCOL.INI file.

When the driver initializes, it tries to find previously loaded instances of itself. If none are found, the driver calls itself "E1000$", and looks for that name in the PROTOCOL.INI file to find its parameters. If one or more instances are found, the driver calls itself "E100x$", where 'x' is one more than the value used by the most recently loaded instance. So, in this scenario, the second driver calls itself "E1002$", the third calls itself "E1003$", and so on; there is no driver called "E1001$". Up to 10 drivers can be loaded in a single system in this way.

Syntax:	DRIVERNAME = [E1000$ | E1002$ | etc.]
Example:	DRIVERNAME = E1000$
Default:	None, this is a required parameter.
Normal Behavior:	The driver finds its section in PROTOCOL.INI by matching its instance ID to the value for this parameter.
Possible Errors:	The device driver uses a DOS function to display the name of the driver it is expecting. This function cannot display a '$' character. For this reason, the user may see a message referring to this value without the '$'; the user must remember to enter the '$' character as part of the parameter's value.

SPEEDDUPLEX

The parameter disables Auto-Speed-Detect and causes the adapter to function at the speed indicated. Do not include this parameter if you want your PRO/1000 adapter to connect at 1000Mbps.

Syntax:	SPEEDDUPLEX = [0 | 1 | 2 | 3]
Example:	SPEEDDUPLEX = 2
Default:	Parameter not included in PROTOCOL.INI
Normal Behavior:	0 = 10Mbps half duplex 1 = 10Mbps full duplex 2 = 100Mbps half duplex 3 = 100Mbps full duplex
Possible Errors:	If the SPEEDDUPLEX parameter is set to an invalid value: • The parameter is ignored and the default (Auto-Speed-Detect) is used • A message indicates a "Parameter value out of range" error

SLOT

This parameter makes it possible for the driver to uniquely identify which of the adapters is to be controlled by the driver. The parameter can be entered in hexadecimal or decimal.

Syntax:	SLOT = [0x0..0x1FFF]
	SLOT = [0..8191]

Examples:	SLOT = 0x1C
	SLOT = 28

Default:	The driver will Auto-Configure if possible.
Normal Behavior:	The driver uses the value of the parameter to decide which adapter to control.
Possible Errors:	If only one adapter is installed, and the value does not correctly indicate the adapter slot: • A message indicates that the value does not match the actual configuration • The driver finds the adapter and uses it If more than one adapter is installed, and the value does not correctly indicate an adapter slot: • A message indicates possible slots to use • The driver loads on the next available slot

NODE

This parameter sets the Individual Address of the adapter, overriding the value read from the EEPROM.

Syntax:	NODE = "12 hexadecimal digits" The value must be exactly 12 hexadecimal digits, enclosed in double quotes. The value can not be all zeros. The value can not have the Multicast bit set (LSB of 2nd digit = 1).
Example:	NODE = “00AA00123456”
Default:	Value from EEPROM installed on adapter
Normal Behavior:	The Current Station Address in the NDIS MAC Service-Specific Characteristics (MSSC) table is assigned the value of this parameter. The adapter hardware is programmed to receive frames with the destination address equal to the Current Station Address in the MSSC table. The Permanent Station Address in the MSSC table will be set to reflect the node address read from the adapter's EEPROM.
Possible Errors:	If any of the rules described above are violated, the driver treats this as a fatal error and an error message occurs, indicating the correct rules for forming a proper address.

ADVERTISE

This parameter can be used to restrict the speeds and duplexes advertised to a link partner during auto-negotiation. If AutoNeg = 1, this value is used to determine what speed and duplex combinations are advertised to the link partner. This field is treated as a bit mask.

Syntax:	ADVERTISE = [1 | 2 | 4 | 8 | 0x20 | 0x2F]:

	0x01 = 10 Half, 0x02 = 10 Full, 0x04 = 100 Half, 0x08 = 100 Full, 0x20 = 1000 Full, 0x2F = all rates
Example:	ADVERTISE = 1
Default:	0x2F (all rates are supported)
Normal Behavior:	By default all speed/duplex combinations are advertised.
Possible Errors:	An error message is displayed if the value given is out of range.

FLOWCONTROL

This parameter, which refers to IEEE 802.3x flow control, helps prevent packets from being dropped and can improve overall network performance. Specifically, the parameter determines what flow control capabilities the adapter advertises to its link partner when auto negotiation occurs. This setting does NOT force flow control to be used. It only affects the advertised capabilities.

***NOTES**:*

- Due to errata in the 82542 silicon, the chip is not able to receive PAUSE frames if the ReportTxEarly parameter is set to 1. Thus, if ReportTxEarly =1 and the driver is running on an adapter using this silicon (such as the PWLA8490), the driver will modify the FlowControl parameter to disable the ability to receive PAUSE frames.
- If half-duplex is forced or auto-negotiated, the driver will completely disable flow control.

Syntax:	FLOWCONTROL = [0 | 1 | 2 | 3 |0xFF]
Example:	FLOWCONTROL = 1
Default:	3
Normal Behavior:	0 = Disabled (No flow control capability) 1 = Receive Pause Frames (can receive and respond to PAUSE frames) 2 = Transmit Pause Frames (can send PAUSE frames) 3 = Both Enabled (can send and receive PAUSE frames) 0xFF = Hardware Default.
Possible Errors:	An error message is displayed if the value given is out of range.

USELASTSLOT

This parameter causes the driver to load on the device in the last slot found in the slot scan. The default behavior of the driver is to load on the first adapter found in the slot scan. This parameter forces the driver to load on the last one found instead.

Syntax:	UseLastSlot = [0 | any other value]
Example:	USELASTSLOT = 1
Default:	0
Normal Behavior:	0 = Disabled, any other value = Enabled
Possible Errors:	None

TXLOOPCOUNT

This parameter controls the number of times the transmit routine loops while waiting for a free transmit buffer. This parameter can affect Transmit performance.

Syntax:	TXLOOPCOUNT = <32-bit value>
Example:	TXLOOPCOUNT = 10000
Default:	1000
Normal Behavior:	Default
Possible Errors:	None

DOS ODI Client Driver

This driver is provided solely for the purpose of loading an OS. It is not intended as a high-performance driver. Please limit use to 1-2 frame types.

Location of Drivers:
10/100 Mbit: \PRO100\DOS\E100BODI.COM
Gigabit: \PRO1000\DOS\E1000ODI.COM (This directory also contains the PROTOCOL.INI file and NET.CFG file for gigabit adapters.)

***NOTES**:*

- If the connection fails, set the Spanning Tree Protocol (STP) option on the switch to "off".
- Novell no longer supports NETX.EXE or the VLM client software. However, as a convenience to Intel customers, a copy of NETX.EXE is provided on the Intel Configuration and Drivers CD. Intel recommends that you use the latest Novell Client32 software.
- With dual-port adapters, the first port must be used before the second port.

NETX General Installation

1. Copy the following files from the PRO100\DOS or PRO1000\DOS directory of the Intel media to the NWCLIENT directory on the hard drive:
   - LSL.COM
   - E100BODI.COM (10/100 Mbit) or E1000ODI.COM (gigabit)
   - IPXODI.COM
   - NETX.EXE
2. Configure AUTOEXEC.BAT and NET.CFG based on the sample files below. NET.CFG should be in the same directory as the driver.
3. Reboot and the client should connect to the server.

VLM Installation

This version of E100BODI.COM requires latest version of the VLM client (version 1.21 or greater).

Refer to the Novell NetWare documentation for complete installation instructions.

TIPS (DOS ODI)

NETX

- In CONFIG.SYS add LASTDRIVE=E.
- In NET.CFG add FIRST NETWORK DRIVE F (or some drive letter after the range of physical devices you've installed) under the NetWare DOS REQUESTER section (indented).

VLM

- In CONFIG.SYS add LASTDRIVE=Z.
- In NET.CFG, add FIRST NETWORK DRIVE F under the NetWare DOS REQUESTER section (indented). If F is already assigned, select a drive letter after the range of physical devices you have installed.
- Promiscuous mode support is not supported since ASM does not include a receive monitor stack. If needed, load RXMONSTK.COM (from the VLM client kit) after LSL.COM and before E100BODI.COM or E1000ODI.COM.
- This version of E100BODI.COM or E1000ODI.COM requires VLM client version 1.21 or higher. Attempting to use an older version of the VLM client may cause connection problems, slow response or computer system lock-ups.

10/100 Mbit Sample Files

AUTOEXEC.BAT:

```
CD\NWCLIENT
LSL
E100BODI
IPXODI
NETX (or VLM optional for NetWare 3.12 or higher)
```

NET.CFG:

```
LINK DRIVER E100BODI
FRAME ETHERNET_802.2 (generally NetWare 3.12 and later servers)
;FRAME ETHERNET_802.3 (generally NetWare 3.11 and earlier)
SPEED=100
PROTOCOL IPX E0 Ethernet_802.2

NETWARE DOS REQUESTER
FIRST NETWORK DRIVE = F (Used only with the VLM.EXE)
PB BUFFERS = 10
```

Gigabit Sample Files

AUTOEXEC.BAT:

```
CD\NWCLIENT
LSL
E1000ODI
IPXODI
```

```
NETX (or VLM optional for NetWare 3.12 or higher)
```

NET.CFG:

```
LINK DRIVER E1000ODI
FRAME ETHERNET_802.2 (generally NetWare 3.12 and later servers)
;FRAME ETHERNET_802.3 (generally NetWare 3.11 and earlier)
SPEEDDUPLEX = 3
PROTOCOL IPX E0 Ethernet_802.2

NETWARE DOS REQUESTER
FIRST NETWORK DRIVE = F (Used only with the VLM.EXE)
PB BUFFERS = 10
```

***NOTE**:* SPEEDDUPLEX parameter: 0=10Mbps half-duplex; 1=10Mbps full-duplex; 2=100Mbps half-duplex; 3=100Mbps full-duplex. The SPEEDDUPLEX keyword is only needed when you are forcing the card to a lower speed and duplex.The link partner must also be forced to the same setting. Do not use the SPEEDDUPLEX keyword if you want to use 1000Mbps.