



NetXen, Inc.

Intelligent NIC™ Getting Started Guide

Version 1.8

February 2, 2007

NetXen, Inc.

3965 Freedom Circle

Fourth Floor

Santa Clara, CA 95054

Telephone: (408) 330-0010

Web Site: <http://www.netxen.com>

Copyright

© Copyright 2006. NetXen, Inc. — All Rights Reserved.

Disclaimer

This is a preliminary document and is subject to change without prior notification and does not represent a commitment on the part of NetXen, Inc. The information contained herein is the proprietary and confidential information of NetXen or its licensors, and is supplied subject to, and may be used only by NetXen's customer in accordance with, a written agreement between NetXen and its customer. Except as may be explicitly set forth in such agreement, NetXen does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document.

Please contact NetXen, Inc. to obtain the latest revision of this document.

Trademarks

NetXen, the NetXen logo, and Intelligent NIC are trademarks of NetXen, Inc. Third Party brands and names are the property of their respective owners.

Revision History

Version	Date	Editor	Notes
1.0	3/15/06	SR	First release of document.
1.1	4/24/06	SR	Reorganize Section 3 for Windows and Linux.
1.2	5/2/06	SR	Added Section 1.1, Related Documents. Added Section 3. CD Content Organization. Added Section 4.7.3, Linux Optimization. Added Sections 5.3.1 through 5.3.4.
1.3	5/23/06	SR	Added Table 4, Advanced Properties Parameters. Rewrite of Section 4.1, Steps for Flashing on Windows. Added new screen shot in Section 3.2.2., Local Area Connection Properties. Changed minimum Linux memory to 1 Gbyte throughout. Miscellaneous edits from internal review.
1.4	6/02/06	SR	Replaced screen shot in step 7 of Section 3.2.1 (Windows Driver Installation) and updated table 4 accordingly to reflect changes.

Version	Date	Editor	Notes
1.5	6/21/06	SR	<p>Replaced all screen shots and associated text for Section 3.2.1, Windows installation.</p> <p>Added footnote to step 3 of Section 3.1, Installing and Connecting the Intelligent NIC Adapter.</p> <p>Rewrote Section 4.4.1, Uninstalling the NetXen NIC software.</p> <p>Update Flash command structures in Section 4. Firmware Updates.</p> <p>Added list of Flash utility commands to Section 4.3, Using the NetXen Flash Utility.</p> <p>Added NIC performance testing and One Server/One Client material to Section B.1.1, NTtcp.</p> <p>Deleted Section 4.2.2, Display Output.</p> <p>Deleted Single-Thread, Uniprocessor portion of Section B.1.1.</p> <p>Deleted Timestamps material from Section B.2, Windows Optimization.</p> <p>Changed recommended setting for Tcp1323Opts from 3 to 1 in Section B.2</p>
1.6	7/18/06	SR	<p>Added Quick Start section 3.</p> <p>Screenshot and text updates to Windows Installation section 4.</p> <p>Throughout section 5, updated the NX_NIC case and added footnote on page 16 accordingly.</p> <p>Updated command sequence in Section 5.2, Steps for Flashing on Windows.</p> <p>Added section 5.4.2, Disabling the NIC device in Windows.</p> <p>Added hex value table in Section B.2, Tcp1323Opts.</p> <p>Added Linux Prerequisite to Driver Installation in Section 3.</p> <p>Added RPM installation procedure to Section 3.</p> <p>Divided Section 4.3 into TAR and RPM installations.</p> <p>Created Section 4.3.2, RPM Installation</p> <p>Through Section 5, clarified Windows and Linux commands.</p> <p>Added RPM uninstall procedure in Section 5.4.3.</p>
1.7	9/05/06	SR	<p>Deleted Section 4.2.4, Verifying Connectivity on Another Machine.</p> <p>Added Section 5.6, Updating the Flash on Multiple Cards.</p> <p>Updated test listing in Section 7, Diagnostics.</p> <p>Added Appendix C, Diagnostic Error Codes.</p>
1.8	1/16/07 1/27/07	SR	<p>Added Section 8, Teaming</p> <p>Added Section 9, VLAN</p> <p>Added Section 10, Linux TOE</p> <p>Updated screen shots</p> <p>Added internal review edits</p>



1	Overview	1
1.1	Related Documents	1
1.2	NIC Versions	1
1.3	Currently Supported Features	1
2	System Requirements	2
2.1	Hardware Requirements	2
2.2	Software Requirements	3
3	Quick-Start Installation Instructions	4
4	Detailed Installation and Configuration Instructions	6
4.1	Installing and Connecting the iNIC Adapter	6
4.2	Windows Installation	6
4.2.1	New Driver Installation	6
4.2.2	Uninstall Procedure	12
4.2.3	Configuring the Network Interface Card	13
4.2.4	Verification of a Successful Installation	16
4.3	Linux Installation	16
4.3.1	Driver Installation (TAR format)	16
4.3.2	Driver Installation (RPM format)	17
4.3.3	Configuring the NIC on Linux	18
5	Firmware Updates	19
5.1	Using the NetXen Flash Utility	19
5.2	Windows Firmware Update	21
5.3	Linux Firmware Updates	21
5.3.1	Linux Firmware Update (TAR Format)	21
5.4	Updating the Flash on Multiple Cards	22
5.5	Disabling the Device in Windows	23
5.6	Uninstalling the Software	23
5.6.1	Uninstalling the NetXen NIC Driver Software in Windows	23
5.6.2	Uninstalling the NetXen NIC Software in Linux	23
6	Troubleshooting Guidelines	24
6.1	Unable to Boot the Host System with the NIC Card Installed	24
6.2	Unable to ping the NIC Installed on Linux Host	24
6.3	Gather System Information and Check for Errors on Linux Host	24
6.4	Caveats	25
6.4.1	MSI Interrupts for Linux	25
6.4.2	PCI-Express Slot	25
7	User Diagnostics	26
7.1	Tests Performed	26
7.2	Test Descriptions	26
7.2.1	Control Registers Test	26
7.2.2	Flash Memory Test	26
7.2.3	Interrupt Test	26
7.2.4	Internal Loop-Back Test	27
7.2.5	External Full Duplex Loop-Back Test	27
7.2.6	Link Test	27
7.2.7	All Tests	27

Table of Contents

8	Teaming Driver	28
8.1	Windows	28
8.1.1	Installation	28
8.1.2	Configuration	30
8.2	Linux	32
9	VLAN Configuration	33
9.1	Windows	33
9.1.1	NIC VLAN as Send Side	33
9.1.2	NIC VLAN as Receive Side	34
9.2	Linux	34
10	Linux TCP Offload Engine (TOE)	35
	Appendix A CD Content Organization	36
	A.1 Driver Updates	36
	Appendix B Performance Tools and Optimization	37
	B.1 Windows Performance Tools	37
	B.1.1 NTttcp (Windows)	37
	B.1.2 nttcps and nttcpr Command Line Options	38
	B.2 Windows Optimization — Tcp1323Opts	39
	B.3 Linux Performance Tools	39
	B.3.1 IPERF	39
	B.3.2 TTCP	40
	B.4 Linux Optimization	40
	Appendix C Diagnostic Error Codes	42

1 Overview

This guide describes the system requirements, installation, and setup required for the NetXen Intelligent NIC™ network interface card product release.

1.1 Related Documents

- Ethernet Driver Developer Reference Guide: For Open Source Software, Version 1.1.
- Release notes included in the software package for specific firmware and driver releases.

1.2 NIC Versions

All of the NIC products support Windows, Linux, and other operating systems. Check the release notes for an updated list.

Product Number	Product Definition
NXB-10GSXR	Single 10-GE port (optical fiber)
NXB-10CX4	Single 10-GE port (copper)
NXB-4CX4	Four 1-GE ports (copper)
Mezanine (Mez)	IBM Blade servers

1.3 Currently Supported Features

Feature	Windows	Linux
TCP, UDP, and IP checksum offload	Yes	Yes
TCP segmentation offload (TSO / LSO)	Yes	Yes
Interrupt coalescing	Yes	Yes
Interrupt distribution	No	Yes
VLAN	Yes	Yes
SNMP Ethernet MIB	Yes	Yes
Jumbo frames	Yes	Yes

Note: Check with NetXen periodically for software updates and additional features.

2 System Requirements

Before installing the product, verify the minimum system and software configuration requirements outlined below.

2.1 Hardware Requirements

Supported Server Systems (x86-based PCI Express Systems)

- Intel Xeon
- AMD Athlon 64
- AMD Opteron 64

Minimum Memory Requirements

- Windows: 2 Gbytes
- Linux: 1 Gbyte

Slot Type

- PCI Express, x8, x4

Cable and Interface Specifications

Table 1 NXB-10GXSX/NXB-10GSLR Cable Specifications

Parameter	NXB-10GXSX		NXB-10GSLR		Units
	50 μm	62.5 μm	9 μm	10 μm	
Wavelength	850	850	1310	1310	nm
Bandwidth (min)	500	500	n/a	n/a	MHz
Operating Range	300	300	10K	10k	m

Table 2 NXB-10CX4 Cable Specifications

802.3 Physical Layout	Network Type	Cable Length
10GBASE-CX4	LAN	15m (max)

2.2 Software Requirements

Supported OS Versions

NetXen supports both Windows and Linux operating systems. Check with NetXen for an updated list of supported OS and kernel versions.

Linux TCP Offload Engine (TOE)

IM (Teaming)

3 Quick-Start Installation Instructions

This section is for users already familiar with the Windows or Linux operating systems. This process provides only the general steps necessary for installing the NetXen NIC hardware, as well as Windows and Linux drivers. For those user’s requiring more detailed information, refer to Section 4 below.

Table 3 Quick Start Instructions — New NIC Driver Installation

NetXen NIC Hardware Installation	Windows Driver Installation	Linux Driver Installation
<p>Pre-Installation</p> <ol style="list-style-type: none"> 1. Verify the system meets the requirements described in Section 2. 2. Power down the system. <p>Installing the NetXen NIC Adapter</p> <ol style="list-style-type: none"> 1. Insert the NetXen NIC adapter card into an available PCI Express slot. 2. Connect the appropriate cables to the board. 3. Power on the system. 	<p>Driver Installation</p> <ol style="list-style-type: none"> 1. Copy all of the provided NIC driver files to a directory. 2. From the Device Manager, scan for new hardware. 3. When prompted to connect to Windows Update, Click <i>No, not at this time</i>, then click <i>Next</i>. 4. In the next window, click <i>Install from a list of specific location</i> and then click <i>Next</i>. 5. Select <i>Search for the test driver in these locations</i>. Select <i>Include this location in the search</i>. 6. Provide the path to the directory where the driver NXP2NIC.sys is located. For 32-bit operating systems, choose the <i>x86</i> directory. For 64-bit operating systems, choose the <i>x64</i> directory. <p>NIC Card Configuration</p> <ol style="list-style-type: none"> 1. Open the <i>Network Connections</i> window from the <i>Control Panel</i>. 2. Verify that NetXen NIC adapter icon says <i>Connected</i>. If not, right-click on the <i>NIC Adapter</i> icon and select <i>Properties</i>, then assign the IP and subnet addresses and click <i>OK</i>. 	<p>Prerequisite to Driver Installation</p> <ol style="list-style-type: none"> 1. Configure the Linux system with at least 1 GB of memory. <p>Driver Installation (tar.gz)</p> <ol style="list-style-type: none"> 1. Untar the file, build, install, and load the driver: <pre>tar xzf <driver file>.tar.gz cd <directory> make make install modprobe nx_nic</pre> <p>NIC Card Configuration</p> <ol style="list-style-type: none"> 1. Identify the NetXen Ethernet interface: <code>ethtool -i eth[x]</code> 2. Assign an IP address: <code>ifconfig eth[x] [IP address]</code> <p>Driver Installation (rpm)</p> <ol style="list-style-type: none"> 1. Install the RPM as follows^a: <code>rpm -i nx_nic-[version].src.rpm</code> 2. Create the binary RPM as follows: <code>rpmbuild -tb /usr/src/redhat/SOURCES/nx_nic-[version].tar.gz, (Redhat) or</code> <code>rpmbuild -tb /usr/src/packages/SOURCES/nx_nic-[version].tar.gz (SuSE)</code> 3. Install the binary as follows: <code>rpm -i /usr/src/redhat/RPMS/x86_64/nx_nic-[Version]-1.x86_64.rpm (Redhat),</code> or <code>rpm -i /usr/src/packages/RPMS/i586/nx_nic-[Version]-1.i586.rpm (SuSE)</code> <p>a. 3.1.44 is an example of [version]</p>

If a NetXen iNIC card is already installed on the system and a driver and/or firmware update is required, refer to the procedure in [Table 4](#) below.

Table 4 Quick Start Instructions — Updating Existing Driver and Firmware

Windows Driver/Firmware Update	Linux Driver/Firmware Update
<ol style="list-style-type: none"> 1. Open DOS Command window. 2. Go to the <i>utils</i> directory in the driver package provided. 3. The following binaries are required to be in the same directory from where the Flash command will be executed: <i>nxflash.exe</i> <i>phantom_obj</i> <i>bootld</i> 4. Run the following commands: <i>nxflash -i NX_NIC -u</i> Writing of the flash requires a few minutes. Please do not interrupt or reboot during this time, as it may corrupt the Flash. 5. Reboot the system. 6. Verify the firmware has been updated using the following command. <i>nxflash -i NX_NIC --info</i> 7. Verify the MAC address displayed matches the one printed on the card. 	<ol style="list-style-type: none"> 1. Run the following command from the bin directory of the driver installation: <i>nxflash -i eth[n] -u</i> 2. Reboot the system.

4 Detailed Installation and Configuration Instructions

4.1 Installing and Connecting the iNIC Adapter

The adapter card is installed by following these steps. Note that it may be useful to write down the MAC address printed on the sticker on the back of the card before installation. This will allow the address to be easily compared with the MAC address generated or modified once the card is installed.

- 1 Verify that the system is powered off.
- 2 Remove the system cover. Remove the screw and associated bracket covering the selected bus slot.
- 3 Once the bracket is removed, insert the adapter into the PCI Express slot¹. Apply uniform pressure on top of the board until it is firmly seated in the bus connector.
- 4 Secure the board using the screw removed in step 2.
- 5 Connect the appropriate cables to the board and replace the cover.
- 6 Power on the system.

4.2 Windows Installation

The following Windows iNIC driver-related files are included:

- *nxp2nic.inf*
- *NXP2NIC.sys*

4.2.1 New Driver Installation

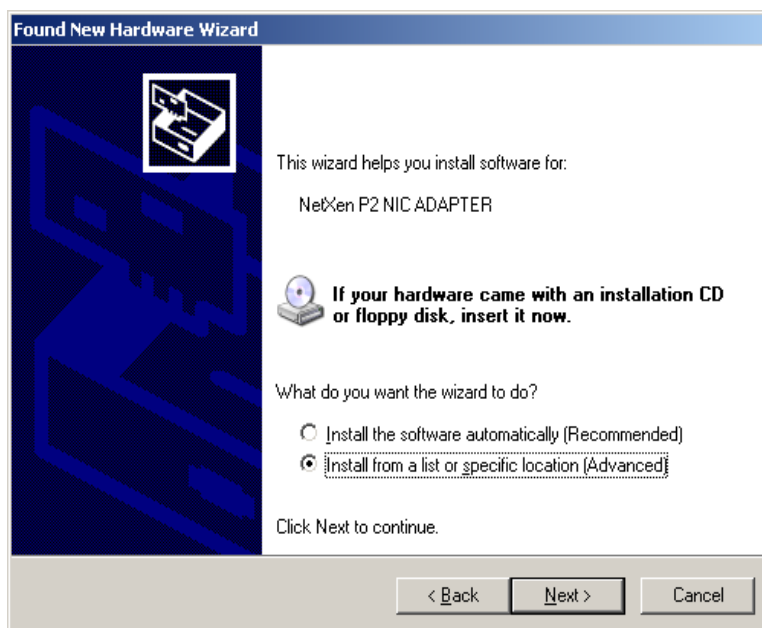
Note that the following procedure is for a new installation. If the driver has been previously installed, refer to Section 5.6.1 before executing this procedure.

- 1 Copy all the NIC files and directories from the software package to a directory on the system's hard drive.
- 2 After the server system is powered on, Windows detects the new card and displays the *Found New Hardware Wizard* window. This only happens only when there are no drivers already installed on the system. When this window appears, Windows asks if it can connect to "Windows Update" to search for the software. Click *No, not at this time*, and click *Next*.

1. Due to space constraints in some systems, when installing an XFP board, it may be necessary to first remove the XFP module from the board, insert the board into the PCI slot, then reconnect the XFP module to the board.

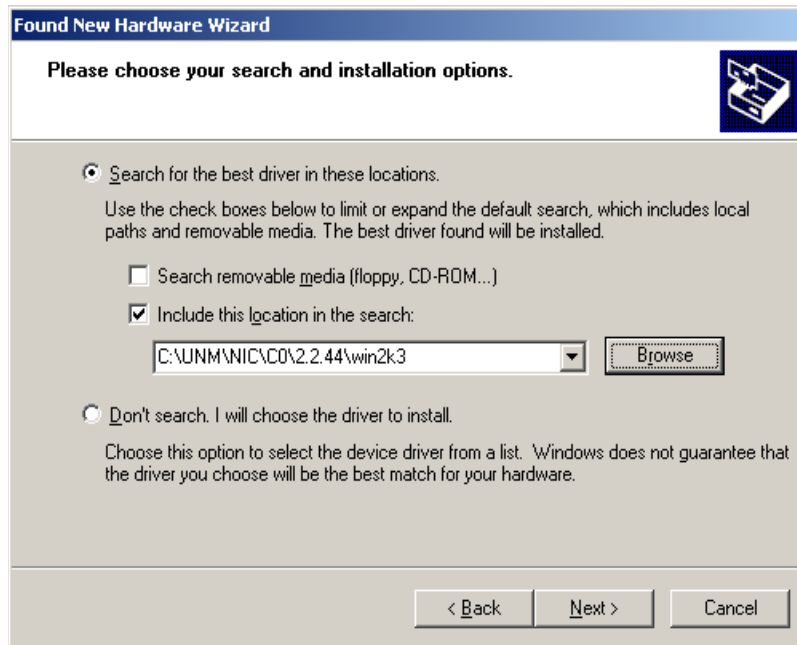


- 3 In the next screen, Windows indicates the software that will be installed. Click *Install from a list or specific location (Advanced)* and click *Next*, as follows:



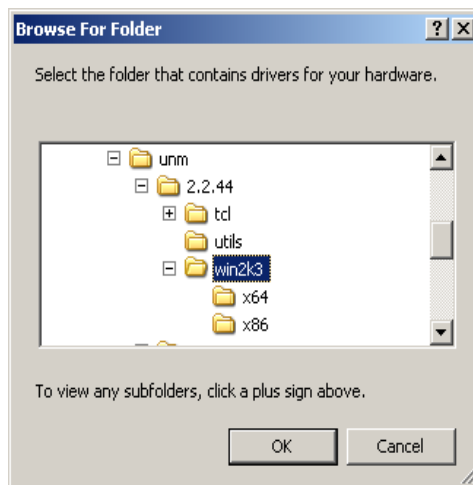
- 4 Next, select *Search for the best driver in these locations*, check the *Include this location in the search* checkbox. Type the appropriate path in the accompanying field and click *Next*. An example driver directory path is shown below:

4. Detailed Installation and Configuration Instructions

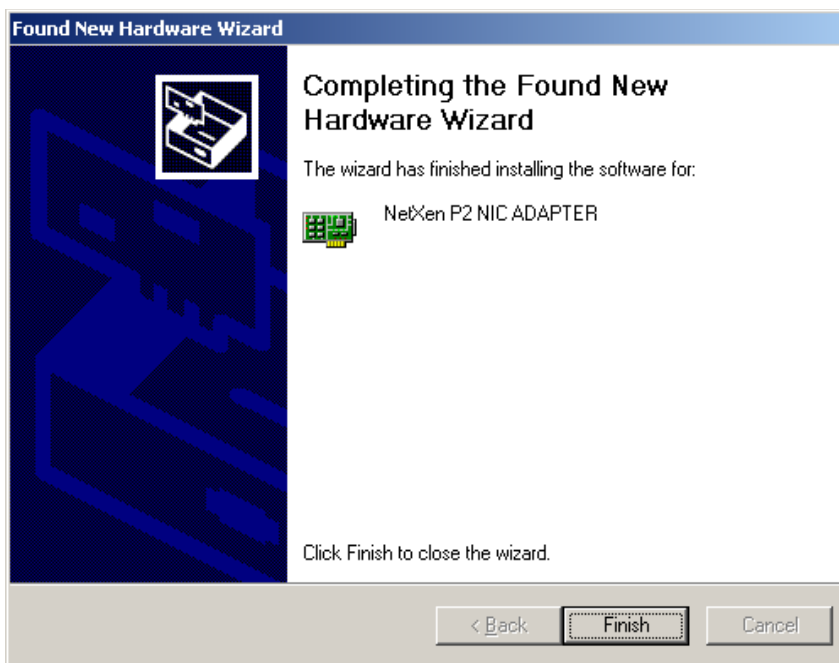


If the exact directory path is not known, click *Browse* to view the directory structure.

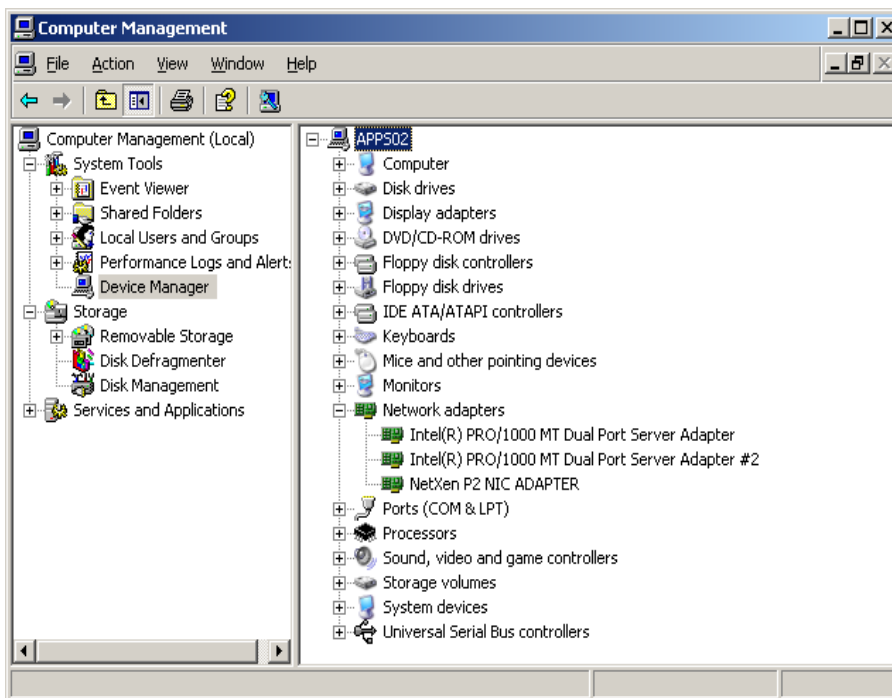
The Intelligent NIC card supports both 32-bit and 64-bit operating systems. The type of operating system can be specified by clicking *Browse* in the above window. This causes the following window to appear on the screen. The x86 directory indicates a 32-bit operating system, while the x64 directory indicates a 64-bit operating system. Highlight the appropriate directory (x86 or x64) and click *OK*.



- 5 Once the installation is complete, the following window will appear. Click *Finish* to complete the installation.

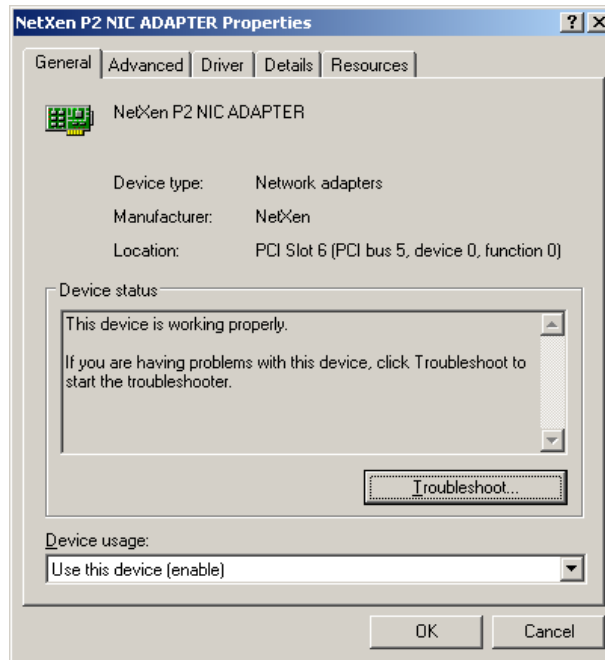


- To verify that the driver was properly installed, Right-click on *My Computer*, then select *Manage*. This opens the *Computer Management* window shown below. Double-click on *Device Manager*. A listing of devices appears in right side of the window. Under *Network Adapters*, the NetXen P2 NIC Adapter should be listed as shown below:

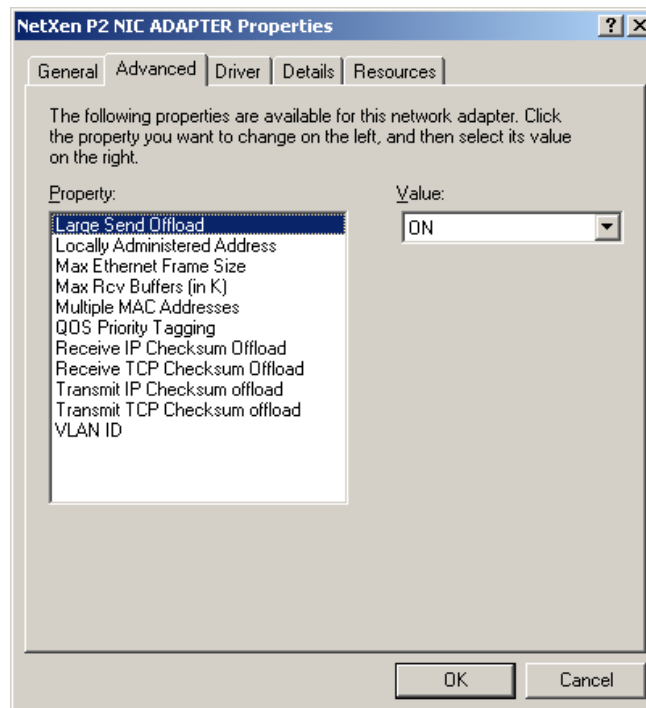


4. Detailed Installation and Configuration Instructions

- 7 In the above window, right-click on the *NetXen NIC ADAPTER* and select *Properties*. The *General* tab will be displayed as shown.



- 8 There are four tabs — *General*, *Advanced*, *Driver*, and *Resources*. Click on the *Advanced* tab. The following information is displayed.



The Advanced tab selections displayed above are defined as follows:

Table 5 Description of Advanced Properties Parameters

Parameter	Description
Large Send Offload	TCP Large send offload – Enable/Disable options. Registry Key: LargeSendOffload Default: Enabled. Size of LSO: 64K
Locally Administered Address	This is the Locally Administered Address (LAA) that users/administrators can set on this interface and this address overrides the permanent address of the adapter (that may have been flashed in to the hardware). Registry Key: NetworkAddress Format: Hex Options: Valid or not valid (if Valid, please enter the 8 hex bytes of MAC address)
Max Ethernet Frame Size	Ethernet frame size for packet transmission and receive (does not include MAC Header). Reg Key: MaxFrameSize Default: 1500 (corresponds to 1514 bytes on the wire + 4 bytes of CRC) Max: 8000 Min: 1500
Maximum Receive buffers (in K)	This is packet buffers used by the driver for receive (DMA memory and indicated to stack). Size of each buffer is 1536 bytes. Registry Key: MaxRxBuffers Min: 2K Max: 32K Default: 10K
Multiple MAC Addresses	This is for setting multiple MAC addresses (up to four) by the IM driver (exposed both by the registry key of type REG_MULTI_sz and by a special set of the OID supported by the miniport). Registry Key: NetworkAddressMMA Format: Hex without any separator between MAC addresses
QOS Priority Tagging	For supporting 802.1p priority tagging. Should be enabled to set VLAN ID. Registry Key: PQTagging Default: DISABLED Values: Enable/Disable
Receive IP Checksum Offload	This parameter is for enabling/disabling the receive IP checksum offload. Registry Key: RxIPChecksumOffload Default: ON Values: ON/OFF
Receive TCP Checksum Offload	This parameter is for enabling/disabling the receive TCP and receive UDP checksum offload. Registry Key: RxTCPChecksumOffload Default: ON Values: ON/OFF

Table 5 Description of Advanced Properties Parameters (Continued)

Parameter	Description
Transmit IP Checksum Offload	This parameter is for enabling/disabling the transmit IP checksum offload. Registry Key: TxIPChecksumOffload Default: ON Values: ON/OFF
Transmit TCP Checksum Offload	This parameter is for enabling/disabling the transmit TCP and transmit UDP checksum offload. Registry Key: TxTCPChecksumOffload Default: ON Values: ON/OFF
VLAN ID	For setting the VLAN ID for this interface (also exposed through the standard OID). Registry Key: VlanId Default: 0 (no VLAN) Range: 0 - 4095

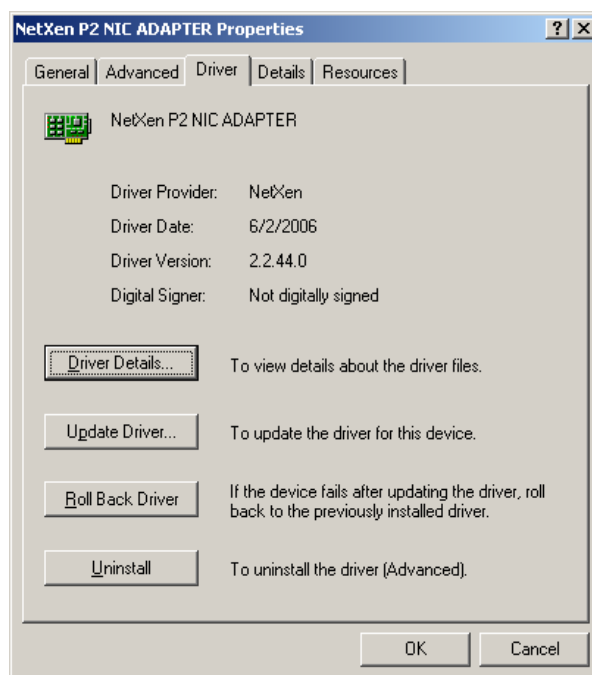
- Click on the *Driver* tab. The following information is displayed.

The driver can be updated from this window by clicking on *Update Driver*.

In addition, if there is a previous version of the driver on the system and the device fails after updating to a new driver, click on *Roll Back Driver* to roll back to the previously installed driver.

4.2.2 Uninstall Procedure

In the NetXen P2 NIC Adapter Properties window shown below, the device can be uninstalled by clicking *Uninstall*. Clicking *Uninstall* performs the same operation as described in Section 5.6.1.

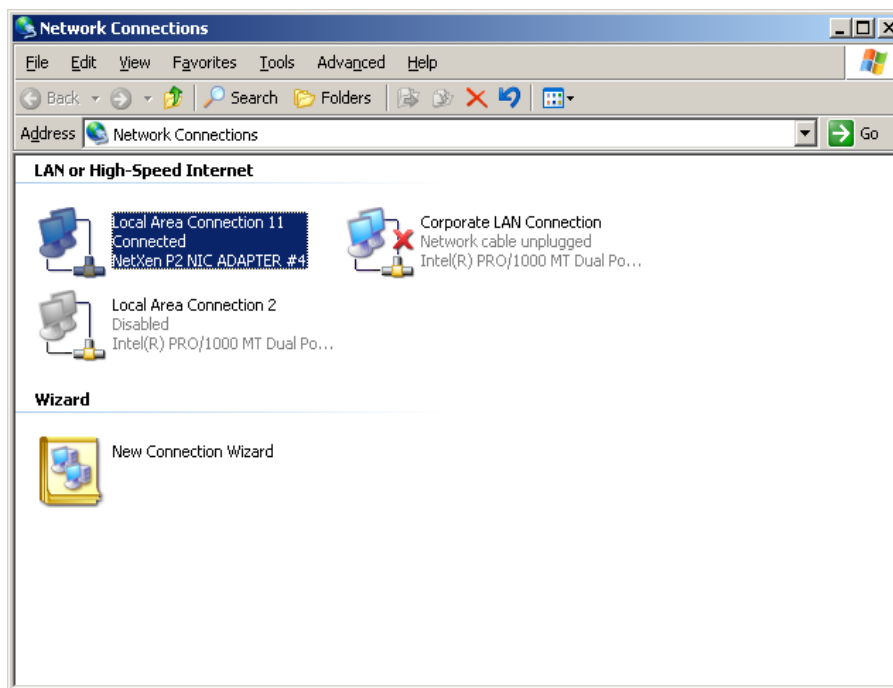


This completes the installation of the NetXen NIC software. Refer to the next section for information on configuring the NIC.

4.2.3 Configuring the Network Interface Card

- 1 Configure the network connections by opening the *Network Connections* window in the control panel, or directly from the desktop if the icon exists. The following window is displayed.

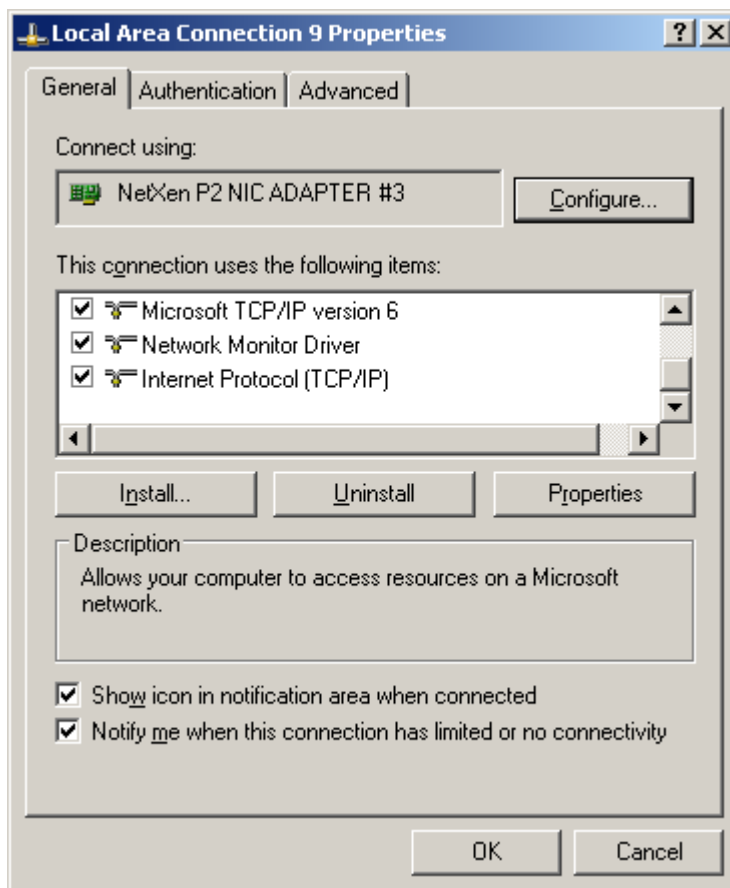
4. Detailed Installation and Configuration Instructions



Note that if the IP address has not been assigned automatically using DHCP, the NetXen NIC adapter entry above will say “Limited or no connectivity”. In this case, the address must be entered manually as described in Step 3 below.

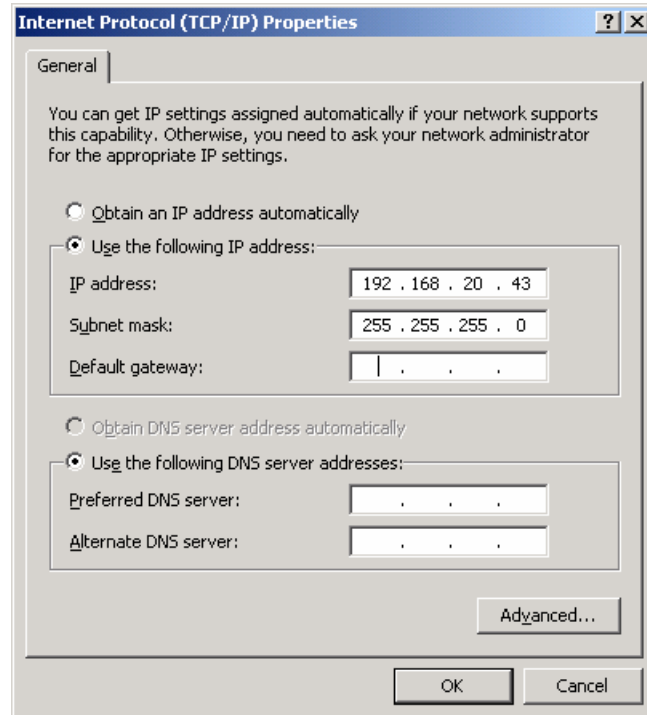
If a red ‘X’ appears in the NetXen P2 NIC Adapter icon, this indicates that the network link is down. Note that the cable must be connected before the verification process in Section 4.2.4 can be performed.

- 2 Right-click on the *NIC Adapter* button, and select *Properties*. In the *Local Area Connection Properties* window which appears (see below), single-click on the *Internet Protocol (TCP/IP)* entry below and click the *Properties* tab.



4. Detailed Installation and Configuration Instructions

- 3 In the *Internet Protocol (TCP/IP) Properties* window, if the IP address was assigned automatically using DHCP, the IP address and Subnet mask entries in the window below will already contains values. If these entries are blank, they must be assigned manually by typing in the appropriate values.



4.2.4 Verification of a Successful Installation

In the *Network Connections* window shown in step 1 of Section 4.2.3 above, right-click on the NetXen P2 NIC Adapter icon and highlight *Status*. If the installation was successful, a “device is working properly” message will be displayed.

Using the ping utility, verify connectivity with another card or switch.

4.3 Linux Installation

The Linux sources can be installed in either TAR or RPM format as described below.

Note that the following procedure is for a new installation. If the driver has been previously installed, refer to Section 5.6.2 before executing this procedure.

4.3.1 Driver Installation (TAR format)

Note: *Root level permissions are required for the Linux Kernel Module (LKM) driver installation.*

- 1 The Linux system must have at least 1 Gbyte of memory.
- 2 Copy the file *tar.gz* to the host system and follow the installation command sequence:

to build, install, and load the driver:

```
su
```

```
tar xvzf 3.x.yy.tar.gz
cd 3.x.yy
make
make install
modprobe nx_nic
```

Where 3.x.yy is the numbered release version of the software.

- 3 The **modprobe** command loads the driver, *nx_nic.ko*, which is installed in:

```
/lib/modules/<kernel version>/kernel/drivers/net
```

- 4 Verify that *nx_nic* driver is loaded:

```
lsmod | grep -i nx
```

4.3.2 Driver Installation (RPM format)

The following subsections describe how to install the Linux source in RPM format, create the binary, and install the binary.

Install the Linux Source in RPM Format

To install the Linux source in RPM format, execute the following command.

```
rpm -i nx_nic-[version]-1.src.rpm
```

The [version] is the numbered release that came with the package. A sample version number would be 3.4.1.

Create the Binary

To create the binary RPM for Redhat systems, execute the following command:

```
rpmbuild -tb usr/src/redhat/SOURCES/nx_nic-[version].tar.gz
```

To create the binary RPM for SuSE systems, execute the following command:

```
rpmbuild -tb usr/src/packages/SOURCES/nx_nic-[version].tar.gz
```

Install the Binary

To install the binary RPM for Redhat systems, execute the following command:

```
rpm -i usr/src/redhat/RPMS/x86_64/nx_nic-[version]-1.x86_64.rpm
```

To create the binary RPM for SuSE systems, execute the following command:

```
rpm -i usr/src/packages/RPMS/i586/nx_nic-[version]-1.i586.rpm
```

As indicated above, and example [version] number would be 3.4.1.

Note that the i586 designation in the path above could also be i686 or i386 or x86_64 depending on the system.

4.3.3 Configuring the NIC on Linux

Assign an IP address to the interface from the command line. For example:

- 1 Identify the Ethernet interface (eth1, eth2, etc.) and the *nx_nic* driver:

```
cat /proc/modules | grep -i nx
```

```
ifconfig -a
```

```
ethtool -i eth2
```

- 2 Assign an IP Address and activate the interface:

```
ifconfig eth2 down
```

```
ifconfig eth2 < IP Address > <netmask>
```

```
ifconfig eth2 up
```

- 3 Verify network services are running, and ping:

```
service network restart
```

```
ping <ip address>
```


5 Firmware Updates

The hardware shipped with the product contains the latest firmware. The firmware must be properly flashed in order for the adapter to function properly. On Linux systems, the tools package is required to flash the current release.

Check with NetXen periodically for firmware updates.

5.1 Using the NetXen Flash Utility

The NetXen Flash utility can be used to modify the contents of the Flash. When using the `nxflash.exe` utility, the following syntax should be used.

```
nxflash -i NX_NIC1 [command] (for Windows OS)
```

```
nxflash -i eth[n] [command] (for Linux OS)
```

In this command line, the `-i` indicates the interface command and the `NX_NIC` specifies the actual interface. the `[n]` indicates the number of the Ethernet interface.

Along with the interface, the available commands are listed in [Table 6](#).

Table 6 NetXen Flash Utility Commands

Command	Definition	Description
<code>-i</code>	interface	Specifies the interface type [NX_NIC, NX_NIC1, etc.]
<code>--info</code>	information	Prints firmware version and MAC address.
<code>--init force</code>	force initialization	Initializes the Flash. Must be done for a new or corrupted Flash.
<code>-u</code>	update	Writes the image area of the Flash, taking it from the files 'bootld' and 'phantom_obj'.
<code>-m</code>	MAC port address	Stores the MAC address of the specified port to the Flash. The address is entered as a string of hex digits.
<code>--recover</code>	recover	Copies an image from a secondary image area to the primary image area.
<code>-h</code>	help	Displays the help menu.

The following are some examples of how the `nxflash` tool may be used to modify the contents of the Flash:

- Modify MAC address
- Update firmware image
- Update boot area
- Get information about the flash contents

Each of these functions is described in detail below.

1 Modify the MAC address

1. Note that the interface type MUST BE ALL CAPITALS to be recognized as a valid interface.

5. Firmware Updates

The `-m` option may be used to modify the mac address. It takes two arguments - the port number and the mac address. The port number may be one of 0, 1, 2, or 3.

For a single port XG card, port 0 must be used. The new mac address must be specified as a hex string, without any spaces or colon (:) between them. Once the mac address is flashed, the driver must be unloaded and reloaded for the new mac address to take effect.

e.g. To flash a MAC address of 00:0e:1e:00:1a:04 to port 0, the command line would be:

```
nxflash -i NX_NIC -m 0 000e1e001a04 (for Windows)
```

```
nxflash -i eth[n] -m 000e1e001a04 (for Linux)
```

2 Update firmware image

The `-u` option may be used to update the firmware image. The following files must be present in the directory from where `nxf`lash is executed:

`phantom_obj` - This is the firmware image.

`bootld` - This is the boot loader.

These files are supplied as part of the software release. To update the firmware, change directory where the above files are located, and type the following:

```
nxflash -i NX_NIC -u (for Windows)
```

```
nxflash -i eth[n] -u (for Linux)
```

This will display the firmware version number of the installed image, and the firmware version number of the image that is being installed, and then display a prompt. If Y is entered, then the Flash update proceeds. The flash updates takes anywhere from 4 minutes to 8 minutes to complete. This must not be interrupted, otherwise the flash may get corrupted. Note that `-update` can be substituted for `-u` in the above command.

3 Update boot area for firmware recovery (not required for normal firmware update)

The `-init` option is used to update the boot area. The boot area contains crucial data that is used by the hardware to boot the chip. If that gets corrupted, then the card becomes unusable. In order to update the boot area, the following files must be present in the current directory:

`pegtune.bin`

To update the boot area, `cd` to the directory and type

```
nxflash -i NX_NIC --init force (for Windows)
```

```
nxflash -i eth[n] --init force (for Linux)
```

4 Get information

The `-info` option may be used to get information about the Flash. e.g.

```
nxflash -i NX_NIC --info (for Windows)
```

```
nxflash -i eth[n] --info (for Linux)
```

The following is an example output when the `-info` command is executed:

```
[root@apps03 bin]# nxflash -i eth2 --info
```

NX Interface is eth2
Supported flash
Firmware version: 3.4.2
Bios version: 255.255.65535
MAC Addr 0 : 00:0E:1E:00:03:78
MAC Addr 1 : 00:0E:1E:00:03:79
Subsys ID : 0x11223344

5.2 Windows Firmware Update

The following steps are used for updating the Windows OS firmware.

- 1 Open DOS Command window.
- 2 Go to the *utils* directory in the driver package provided.
- 3 The following binaries are required to be in the same directory from where the Flash command will be executed:

nxflash.exe
phantom_obj
bootld

- 4 Run the following commands:

nxflash -i NX_NIC -u

Writing of the flash requires a few minutes. Please do not interrupt or reboot during this time, as it may corrupt the Flash.

- 5 Reboot the system.
- 6 Verify the firmware has been updated using the following command.

nxflash -i NX_NIC --info

- 7 Verify the MAC address displayed matches the one printed on the card.

5.3 Linux Firmware Updates

The following steps are used for updating the Linux OS firmware.

- 1 Run the following command from the bin directory of the driver installation:

nxflash -i eth[n] -u

- 2 Reboot the system.

5.3.1 Linux Firmware Update (TAR Format)

The following steps are used for Flashing the Linux firmware in TAR format.

5. Firmware Updates

- 1 Untar the following driver and tools archive and build it.

```
tar xzf 3.x.yy-zz.tar.gz
cd 3.x.yy
make
make install
```

- 2 Load the driver
`modprobe eth[n]`

- 3 Update the firmware
`cd 3.x.yy/bin`
`../nxflash/bin -i eth[n] --init force`
`../nxflash/bin -i eth[n] -update`

Note that the command 3.x.yy above indicates the numbered release of the product. For example, 3.4.2.

At the prompt, press Y and then press enter.

The following is an example output when the update (-u) command is executed.

```
[root@apps03 bin]# nxflash -i eth2
NX Interface is eth2
Supported flash
Searching for NX NIC binaries
NIC binaries found in /usr/local/src/nx/3.4.4/bin
NIC image file path is /usr/local/src/nx/3.4.4/bin/phantom_obj
Currently in flash -> Firmware version: 3.4.2
Bios version: 255.255.65535
MAC Addr 0    : 00:0E:1E:00:03:78
MAC Addr 1    : 00:0E:1E:00:03:79
Subsys ID     : 0x11223344
>From Image file -> Firmware version: 3.4.4
Do you wish to upgrade? (Y/N)
```

- 4 Reboot the system.

5.4 Updating the Flash on Multiple Cards

The Flash can be updated on multiple cards on the same host using one of the following procedures.

- 1 If the updates are being done from the same directory tree, the following commands must be run sequentially after the Flash update has completed.

```
nxflash.exe -i NX_NIC0 -u
nxflash.exe -i NX_NIC1 -u
```

- 2 If the updates are being done from two different directory trees, the updates can be done simultaneously. Note that the following two commands are being executed from two separate directory trees.

```
c:\2.3.57-a\FlashUpdate\ nxflash.exe -i NX_NIC0 -u
c:\2.3.57-b\FlashUpdate\ nxflash.exe -i NX_NIC1 -u
```

5.5 Disabling the Device in Windows

To disable the NIC device in Windows, perform the following steps:

- 1 Right-click on *My Computer* and click on *Manage*. The *Computer Management* window is displayed.
- 2 Double click on *Device Manager*. The list of devices in the computer is displayed on the right side of the window.
- 3 Right-click on *NetXen P2 NIC Adapter*.
- 4 Click the *Disable* tab.
- 5 Click *Yes* to disable the device.

5.6 Uninstalling the Software

This section describes how to uninstall the NetXen NIC software from Windows and Linux systems.

5.6.1 Uninstalling the NetXen NIC Driver Software in Windows

To uninstall the NetXen NIC adapter driver software in Windows, perform the following steps:

- 1 Right-click on *My Computer* and click on *Manage*. The *Computer Management* window is displayed.
- 2 Double click on *Device Manager*. The list of devices in the computer is displayed on the right side of the window.
- 3 Right-click on *NetXen P2 NIC Adapter*.
- 4 Click *Uninstall* to perform the operation.

5.6.2 Uninstalling the NetXen NIC Software in Linux

In the Linux system, the TAR archive may be removed as follows:

```
rm -rf /usr/redhat/SOURCES/nx_nic-3.x.yy
rm -f /usr/redhat/SOURCES/nx_nic-3.x.yy.tar.gz
```

The driver may be removed from `/lib/modules` as follows:

```
rm -f /lib/modules/$(uname -r)/kernel/drivers/net/nx_nic.ko
```

The RPM package may be removed as follows:

```
rpm -e nx_nic-3.x.yy
```

6 Troubleshooting Guidelines

6.1 Unable to Boot the Host System with the NIC Card Installed

- Remove the NIC and reboot. Does the system boot without the NIC?
- Is the recommended minimum memory installed on the host system?
- Refer to Section 6.4, "Caveats" for the 2-Gbyte Windows or 1-GByte Linux configuration entries.
- Note any error message during the OS boot sequence.
- Is the ROM scan enabled in the BIOS? If so, disable it.
- Check for system error messages:
 - Dbgview.exe* (Windows event viewer)
 - dmesg* (Linux)

Use the NetXen diagnostics utility for further testing.

6.2 Unable to ping the NIC Installed on Linux Host

- Type **lspci -v** and look for vendor ID '4040'. Is the NIC visible on the PCI bus?
 - Is the NIC is placed in the PCIe x8 slot?

6.3 Gather System Information and Check for Errors on Linux Host

```
cat /proc/modules | grep -i nx
```

```
cat /proc/pci
```

```
cat /proc/interrupts
```

```
ethtool -i <network interface>
```

```
lspci -v
```

```
lspci -vt
```

- Check for error messages from the kernel:
 - Type **dmesg**
 - Type **cat /var/log/messages**
- Check routing table, gateway, and IP Address conflicts for all interfaces:
 - Type **route**
- Check network interface TX and RX counts during pings:
 - Type **ifconfig <network interface>**
- Use *ethereal*, *tethereal*, *nospace*, or *tcpdump* utilities to capture packets:

- Analyze the ping packets

6.4 Caveats

- Currently, 128 Mbyte of PCI memory space is used for this device/function.
- Jumbo frames of maximum size 8000 are supported.
- The current version of the Windows Win2k3 OS does not support MSI in the network drivers (NDIS 5.1) and as a result MSI is not used by the Intelligent NIC driver. However, hardware support is provided for NDIS versions 5.2 and 6.0.

6.4.1 MSI Interrupts for Linux

The Linux OS version supports MSI, and is used by the Intelligent NIC driver (if available).

- On the Linux host, type **cat /proc/interrupts** to check that PCI-MSI is enabled for the Ethernet interface

Determine if the installed Linux kernel has the MSI configuration option enabled (earlier versions of the kernel may require an MSI patch).

Type **make menuconfig** from the directory */usr/src/kernels/<kernel version>* and verify that MSI configuration option is supported/enabled.

6.4.2 PCI-Express Slot

The NetXen reference card can be located on the PCI-Express bus using the following Linux command.

NetXen device ID = 4040

```
$ lspci | grep -i 4040
```

7 User Diagnostics

The user diagnostics can be invoked using the following command:

nx_win_diag.exe (Windows)

[3.xx.yy]/nxdiag/nxdiag -i eth[n] (Linux)

7.1 Tests Performed

The user diagnostics utility performs the following tests.

- 1 Control registers test
- 2 Flash memory test
- 3 Interrupt test
- 4 Internal loop-back test
- 5 External full duplex loop-back test
- 6 Link Test
- 7 All tests

Each of these tests is described in the following subsections. Refer to Appendix C for a list of diagnostic error codes. Note that only tests which generate specific error codes are listed in Appendix C. If an error code is not listed for a specific test, it is because that test generates only a simple pass/fail indication and not an actual error code.

7.2 Test Descriptions

7.2.1 Control Registers Test

In this test the user diagnostic utility reads the values stored in the control registers. The setting of clearing of certain bits during the initialization process indicates whether the board has been configured correctly. The utility also performs a write/read back/compare operation to selected registers to verify each register's accessibility.

7.2.2 Flash Memory Test

In this test the user diagnostic utility reads the contents of the Flash memory, calculates the checksum, and compares the calculated checksum to the one stored in memory.

7.2.3 Interrupt Test

In this test the user diagnostic utility communicates with the driver to force the hardware to generate selected interrupts, then reads the interrupt status register to verify that the interrupt was generated and processed.

7.2.4 Internal Loop-Back Test

In this test the user diagnostic utility sets the Ethernet core (MAC) to perform a loopback operation. The utility sends packets to itself, then compares the contents of the packets to verify that the sent and received values are the same. No external hardware is required for the internal loopback test.

7.2.5 External Full Duplex Loop-Back Test

For CX4 applications, an external module is required to connect the four transmit lanes to the four corresponding receive lanes. This module must contain all of the associated handshake signals required to redirect the four transmit lanes to the four receive lanes. In this test the utility sends packets to itself through the optical transmit lane, reads the contents on the receive lane, then compares the contents of the packets to verify that the sent and received values are the same.

For XFP applications, an 850 nm multimode cable with LC connectors is required to externally connect the transmit lane to the receive lane.

7.2.6 Link Test

This test reports whether the link is up or down and also checked the carrier detect on the PHY.

7.2.7 All Tests

This option runs all of the above tests using a single command.

8 Teaming Driver

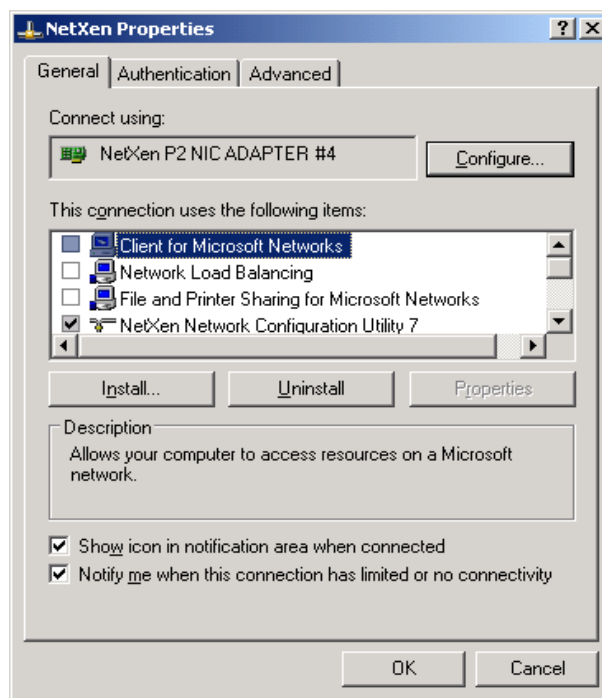
This section discusses the installation and configuration options for Windows and Linux.

8.1 Windows

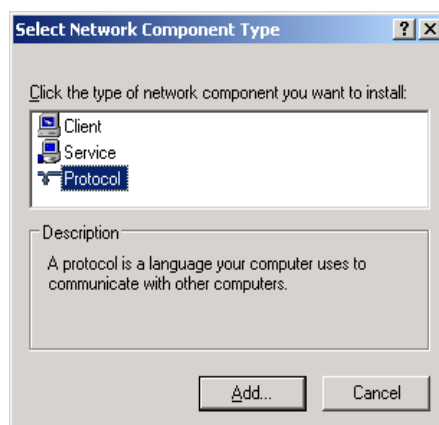
8.1.1 Installation

Once the Windows driver ****what driver**** has been installed and the firmware updated, the IM (Intermediate driver) can be installed using the following procedure:

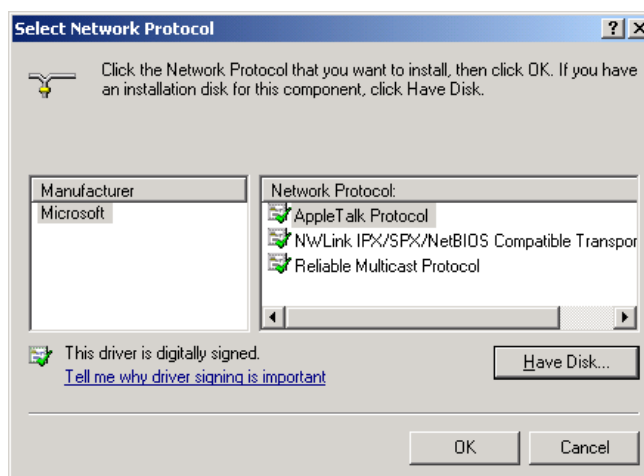
- 1 Install the Microsoft Visual C++ Distributable package (x86 or x64) -- vcredist_x86.exe or vcredist_x64.exe
 - The software is included in the NetXen driver package, or it can be downloaded directly from Microsoft.
 - Microsoft Visual C++ Redistributable Package. The Microsoft Visual C++ 2005 Redistributable Package (x86) installs runtime components of Visual C++ Libraries required to run applications developed with Visual C++ on a computer that does not have Visual C++ 2005 installed. This package installs runtime components of C Runtime (CRT), Standard C++, ATL, MFC, OpenMP and MSDIA libraries
- 2 Select and open "Network Connections" window.
- 3 Right Click on NetXen NIC Adapter icon to open the **NetXen Properties** window:



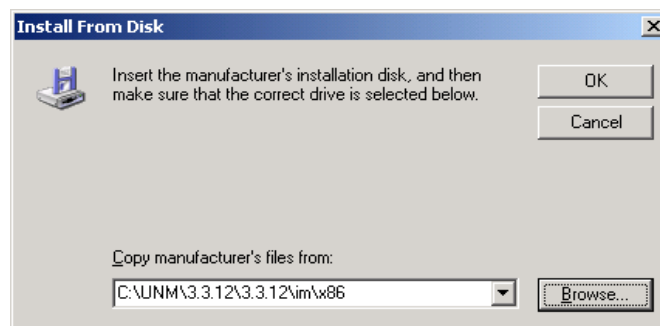
- 4 Select the appropriate item from the list and click *Install*, then *OK*. The **Select Network Component Type** window is displayed.



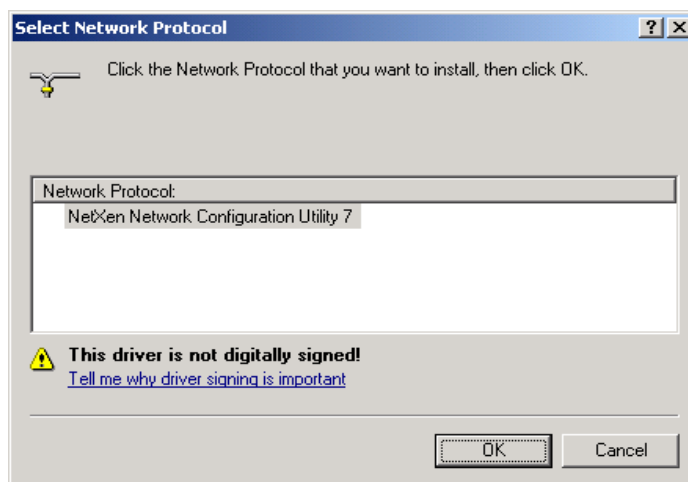
- 5 Select *Protocol* and click the *Add* button. This causes the **Select Network Protocol** window to be displayed as shown below.



- 6 Click on the *Have Disk* button in the **Select Network Protocol** window above. This displays the **Install From Disk** window as shown below.



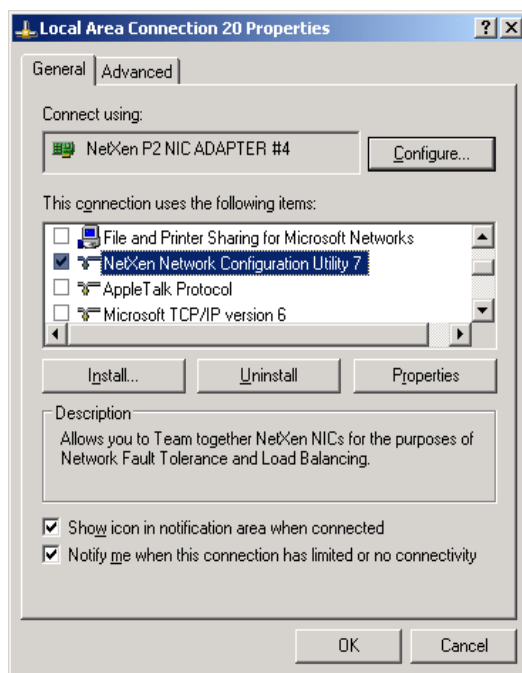
- 7 In the **Install From Disk** window, open and select the NetXen IM subdirectory containing the nxteam driver as shown and click *OK*. The **Select Network Protocol** window is again displayed with the information shown below.



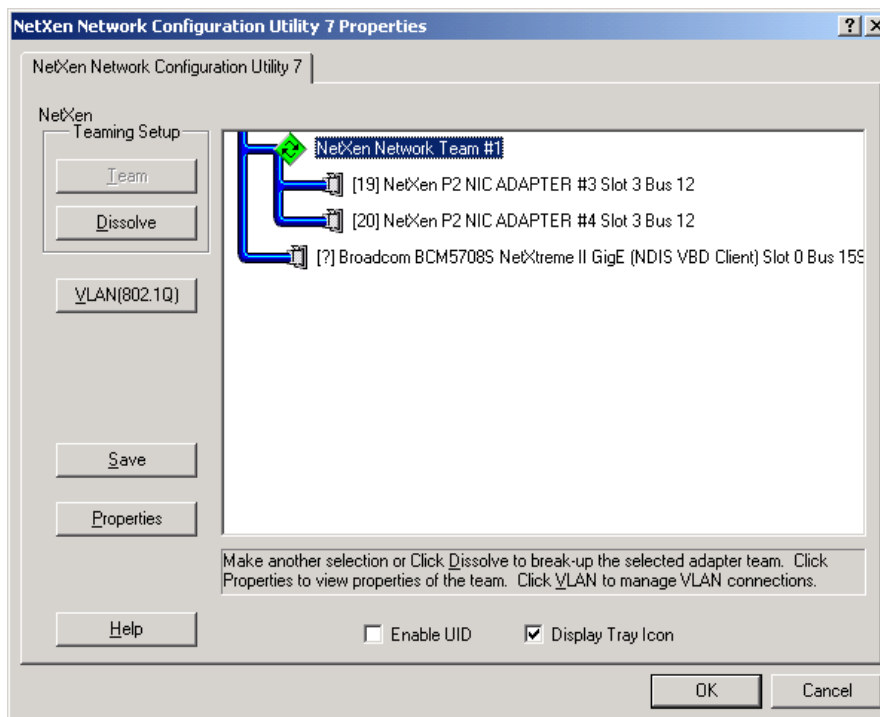
- 8 Select *NetXen Network Configuration Utility 7* and click *OK*. After the driver is installed, Windows will ask to reboot the host.

8.1.2 Configuration

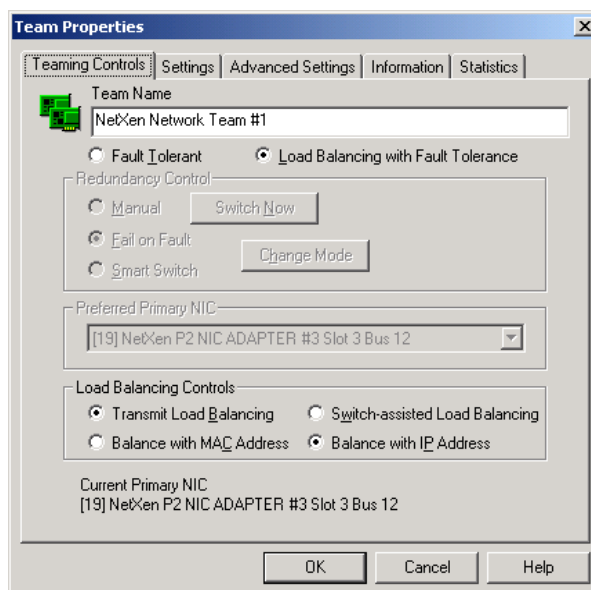
The icon for **Launch NetXen Network** is located on the toolbar icon tray. Double-click this icon to launch the Network Configuration GUI. Alternatively, you can open the NetXen NIC adapter properties, and click the *Properties* button to launch the NetXen Network Configuration Utility GUI to configure teaming. Either of these procedures causes the **Local Area Connection 20 Properties** window to be displayed as shown below.



Single click on the *NetXen Network Configuration Utility 7* entry, then click *Properties* to display the **NetXen Network Configuration 7 Properties** window shown below.

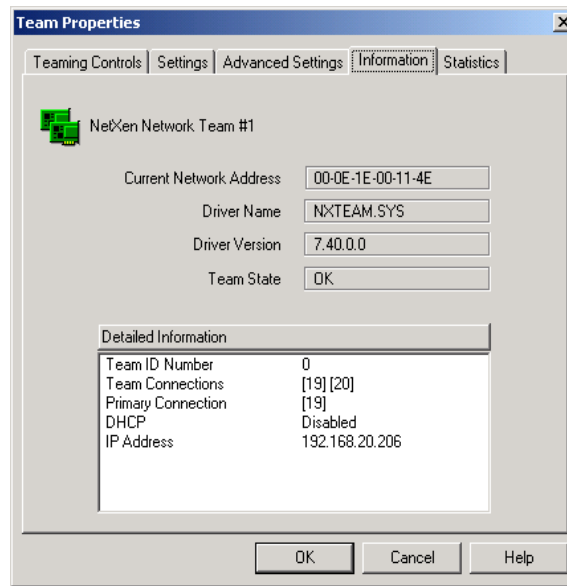


This **NetXen Network Configuration 7 Properties** window contains all the configuration options to setup the teaming properties for the iNIC and its ports. Double-click on the NetXen Network Team #1 entry to display the **Team Properties** window shown below.



In the **Team Properties** window, click on the *Information* tab to display general information about the board as shown below.

8. Teaming Driver



8.2 Linux

9 VLAN Configuration

9.1 Windows

This section describes the NIC VLAN on Windows using the miniport driver (NIC Driver). The IM Driver GUI can set multiple VLAN IDs, unlike the NIC Driver property sheet. The VLAN decision tables shown below describe how the packets are handled by the NIC driver.

9.1.1 NIC VLAN as Send Side

To generate the components of the packet's tag header a miniport driver:

- 1 Sets the tag header's CFI bit to zero.
- 2 Proceed as indicated in the following table, based on the following values:
 - Configured VLAN ID for the NIC.
 - Priority or VLAN ID in the packet's OOB data.
 - VLAN ID in the packet's tag header.

Table 7 NIC VLAN as Send Side

Configured VLAN ID for the NIC	Priority or VLAN ID in the Packet's OOB Data	VLAN ID in the Packet's Tag Header	Miniport Driver Action
Zero	Priority and VLAN ID are zero.	Any value or not present	Send the packet with no changes.
Zero	Priority or VLAN ID is nonzero.	Not present	Insert tag header into packet.Set VLAN ID in tag header to the value of VLAN ID in OOB data.Send packet.
Nonzero	VLAN ID is zero or matches the configured VLAN ID for the NIC.	Not present	Insert tag header into packet.Set VLAN ID in tag header to the value of configured VLAN ID for the NIC.Send packet.
Nonzero	VLAN ID matches the configured VLAN ID for the NIC.	Zero	Set VLAN ID in tag header to the value of configured VLAN ID for the NIC.Send packet.
Nonzero	Not present.	Matches the configured VLAN ID for the NIC.	Send the packet with no changes.

If the priority value that the miniport driver retrieves from the packet is zero, the miniport driver should not mark the packet with a priority value before transmitting the packet over the network. On Ethernet (802.3) networks, an unmarked packet is the same as a packet that is marked with the "best effort" priority value appended to it. Not marking packets ensures backward compatibility with devices on the network that do not support the tag header.

9. VLAN Configuration

Note that if a send packet's Ethernet header specifies a protocol type (EtherType) of 802.3ad or GARP VLAN Registration Protocol (GVRP), the miniport driver should transmit the packet without modifying it.

The following table indicates the circumstances under which a miniport driver should not send an 802.1Q-marked packet.

Table 8 Conditions for Sending an 802.1Q Packet

Configured VLAN ID for the NIC	Priority or VLAN ID in the Packet's OOB Data	VLAN ID in the Packet's Tag Header	Miniport Driver Action
Zero	VLAN ID does not match the configured VLAN ID for the NIC.	Any value or not present.	Fail to send.
Nonzero	Not present.	Does not match the configured VLAN ID for the NIC.	Fail to send.
Any	Non zero	Non zero	Fail to send because the VLAN ID should not be set in both OOB data and tag header.

9.1.2 NIC VLAN as Receive Side

The following table summarizes the actions that a miniport driver should perform on an 802.1Q receive packet:

Table 9 Conditions for Receiving an 802.1Q Packet

Configured VLAN ID for the NIC	Packet Contains a Tag Header?	VLAN ID in the Packet's Tag Header	Miniport Driver Action
Zero	Yes	Any value.	Remove priority and VLAN ID from the packet. Insert removed priority and VLAN ID into per-packet information. Indicate the packet.
Zero	No	Not applicable	Set the priority and VLAN ID in per-packet information to zero. Indicate the packet.
Nonzero	Yes	Matches the configured VLAN ID	Remove priority and VLAN ID from the packet. Insert removed priority and VLAN ID into per-packet information. Indicate the packet.
Nonzero	Yes	Does not match the configured VLAN ID	Do not indicate the packet.
Nonzero	No	Not applicable	Do not indicate the packet.

9.2 Linux

10 Linux TCP Offload Engine (TOE)

Appendix A CD Content Organization

The CD contains the latest driver files for Windows and Linux for the versions specified in the System Requirements section.

A.1 Driver Updates

The CD ships with the latest versions of the Windows and Linux driver files. However, software updates are made on a continual basis. Periodically check with NetXen for software updates.

Appendix B Performance Tools and Optimization

B.1 Windows Performance Tools

B.1.1 NTttcp (Windows)

There are two switches on *ntttcp* that boost performance: **-a** and **-rb**.

Running *ntttcp* without an optimal value for the **-a** switch and without the **-rb** switch is acceptable for the Gigabit NIC product.

In the 10G NIC product, however, and in particular with fewer threads on Windows, performance is limited by the Receive window for the most part and by the outstanding I/O (**-a**) value, regardless of hardware limitations.

For receive, an optimal value is **16** for **-a** and **512000** for **-rb** (or something similar). With these values, a single thread should perform closer to a multi-thread configuration.

```
ntttcpr -m 16,0,<rcv ipaddr> -n <# of tcp segs for snd/rcv> -a 16
```

For transmit, **-a** should be a value less than 8. The more threads, the lower the number should be.

Note: *The send and receive side NTTTTCP commands must use the receive side's IP address.*

NTttcp Example

Command Options ntttcpr	Description
-l	Message size in bytes. Default is 64K.
-m	Number of threads.
0	CPU number.
IP Address	Receive IP address. (Send side also uses receive side's IP Address).
-n	Number of TCP segments.
-a	Number of requests in parallel. (Send and receive side values must match).
-rb	Receive window size on the receiver.

Steps for NIC Performance Testing using NTTTTCP on Windows

- 1 Install Win2k3 SP1
- 2 Enable Windows Scaling
Set Tcp1323Opts parameter value to 1, if the parameter does not exist, create it:
"regedit" to HKLM\system\currentcontrolset\services\Tcpip\Parameters
- 3 The drivers property sheet default is 1500; for Jumbo, set the frame size to 8000.
- 4 Verify the system has the latest NTTTTCP version.

Unidirectional: One Server to One Client

Receiver:

```
ntttcpr.exe -m 8,0,<Receive Side IPAddr> -a 16 -n 500000 -rb 8000000 -p 5001  
-l 512k
```

Sender:

```
ntttcps.exe -m 8,0,<Receive Side IPAddr> -a 8 -n 500000 -p 5001 -l 512k
```

Unidirectional: One Server to One Client — Single Thread, Multiple Processors

If the system has more than one CPU, the load can be distributed across two CPUs and port numbers.

Send Side

```
ntttcps -m 1,0,192.168.20.2 -n 300000 -a 6 -p 5001  
ntttcps -m 1,1,192.168.20.2 -n 300000 -a 6 -p 5002
```

Receive Side

```
ntttcpr -m 1,1,192.168.20.2 -n 300000 -a 16 -rb 512000 -p 5001  
ntttcpr -m 1,1,192.168.20.2 -n 300000 -a 16 -rb 512000 -p 5002
```

B.1.2 nttcps and nttcpr Command Line Options

```
C:\NTTTCP_NEW>ntttcps
```

or

```
C:\NTTTCP_NEW>ntttcpr
```

```
ntttcps: [-l|-n|-p|-a|-t|-rb|-sb|-i|-u|-w|-x|-d|-v|-6] -m <mapping>  
[mapping]
```

-l	<Length of buffer>	[default: 64K]
-n	<Number of buffers>	[default: 20K]
-p	<Port base>	[default: 5001]
-a	[outstanding I/O]	[default: 2]
-x	[PacketArray size]	[default: 1]
-t	<Timeout Period>	[default: 30s]
-rb	<Receive buffer size>	[default: 64K]
-sb	<Send buffer size>	[default: 0]
-i	Infinite Loop	[Only UDP mode]
-u	UDP send/recv	
-w	WSARecv/WSASend	
-d	enable data verification	
-v	enable verbose mode	
-6	enable IPv6 mode	
-m	<mapping>	[mapping]

where a mapping is a session(s),processor,receiver IP set

e.g. `-m 4,0,1.2.3.4` sets up:
4 sessions on processor 0 to test a network on 1.2.3.4

B.2 Windows Optimization — Tcp1323Opts

This parameter determines whether TCP uses the time-stamping and window scaling features described in *RFC 1323, TCP Extensions for High Performance*.

Window scaling permits TCP to negotiate a scaling factor for the TCP receive window size, allowing for a very large TCP receive window of up to 1 Gbyte. The TCP receive window is the amount of data the sending host can send at one time on a connection.

To set the window scaling and timestamp parameters, click Start -> Run, then type *regedit* to start the registry editor. Open the following location:

HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

Data Type	Range	Recommended Value
REG_DWORD	0 1 2 3	1

Hex Value	Meaning
0	Timestamps and window scaling are disabled.
1	Window scaling is enabled.
2	Timestamps are enabled.
3	Timestamps and window scaling are enabled.

Window Scaling

For more efficient use of high bandwidth networks, a larger TCP window size may be used. The TCP window size field controls the flow of data and is limited to 2 bytes, or a window size of 65,535 bytes.

Since the size field cannot be expanded, a scaling factor is used. TCP window scale is an option used to increase the maximum window size from 65,535 bytes to 1 Gigabyte.

When the value for window size is added to the registry and its size is larger than the default value, Windows attempts to use a scale value that accommodates the new window size.

B.3 Linux Performance Tools

Once connectivity between the two cards is established, common performance tools can be used for performance measurements.

Note: *Various factors including tools, kernel, stacks, and related parameters affect the performance of a system. A discussion on performance criteria is beyond the scope for the getting started section.*

B.3.1 IPERF

The **-P** option determines the number of threads; default is one.

Server Side:

```
iperf -s -w256k -l 56k
```

Client Side:

```
iperf -c <ipAddr of Server> -w256k -P8 -l 56k
```

B.3.2 TTCP

For each thread, add an entry in the server and client side scripts.

Receive Side (Server):

```
% ttcp -r -s -l 65536 -fg -n 40000 -p 5001 &
% ttcp -r -s -l 65536 -fg -n 40000 -p 5002 &
...
```

Transmit Side (Client):

```
% ttcp -t -s -l 65536 -n 40000 -D -fg -p 5001 192.168.20.50 &
% ttcp -t -s -l 65536 -n 40000 -D -fg -p 5002 192.168.20.50 &
...
```

B.4 Linux Optimization

While optimization of the Linux operating system is a topic beyond the scope of this guide, below are several of the more commonly modified network-specific Linux tunable parameters.

These tunable parameters make the time default values for TCP/IP connection lower so that more connections can be handled by time on the TCP/IP protocol. The following parameters help to:

- Decrease the amount of time the Linux OS requires to close a connection.
- Reduce the amount of time required for a stale connection to be terminated.
- Turn off some IP extensions that are not needed.

The default setup for the TCP/IP parameters under Red Hat Linux are modified as follows:

Edit the `/etc/sysctl.conf` file, or use the `sysctl` command to set the values for these variables:

`net.ipv4.tcp_timestamps` - To turn off TCP timestamping (avoid the 12 byte Timestamp overhead).

`net.ipv4.tcp_sack` - To turn off TCP source acknowledge.

The `tcp_rmem` and `tcp_wmem` contain arrays of three parameter values. The three numbers represent minimum, default and maximum memory values. These three values are used to bound auto-tuning and balance memory usage while under global memory stress.

The tunable parameters for the TCP receive buffers are:

- *net.ipv4.tcp_rmem* - memory reserved for TCP receive buffers (reserved memory per connection default).
- *net.core.rmem_default* - TCP receive window default size
- *net.core.rmem_max* - maximum TCP receive window size

The tunable parameters for the TCP transmit buffers are:

- *net.ipv4.tcp_wmem* - memory reserved for TCP send buffers (reserved memory per connection default).
- *net.core.wmem_default* - TCP send window default size
- *net.core.wmem_max* - maximum TCP send window size

Other Linux tunable parameters are:

- *net.ipv4.tcp_mem* - default memory reserved.
- *net.core.optmem_max* - increase optional max memory to a higher value
- *net.core.netdev_max_backlog* - increase number of incoming connections backlog
- *net.core.somaxconn* - increase number of incoming connections backlog

The optimal values for these tunable parameters are typically workload-specific (and sometimes system-specific). A thorough familiarity with the affects of these parameter values, as well as an understanding of the target workload and system configuration (amount of memory, etc), will help in determining the optimal values for these parameters.

Appendix C Diagnostic Error Codes

The Diagnostic utility generates error codes based on the type of test being executed. Those tests that generate specific error codes are listed below. For all other tests, the utility provides only a pass/fail indication and does not generate specific error codes.

Diagnostic Error Codes

Test Type	Error Code	Description
Loopback	LB_TEST_OK	Loopback test has passed.
	LB_UCOPY_PARAM_ERR	Error copying parameters.
	LB_UCOPY_DATA_ERR	Error copying user data.
	LB_NOMEM_ERR	No memory error.
	LB_TX_NOSKB_ERR	No SKB for transmit error.
	LB_SHORT_DATA_ERR	Looped data short error.
	LB_SEQUENCE_ERR	Looped data out of sequence.
	LB_DATA_ERR	Looped data corrupted.
Control Registers	LB_ERR_CNT	Looped error count.
	CR_TEST_OK	Control register test passed.
	CR_ERROR	Error reading control register.
Command Interface	CR_ERRCNT	Control register error count.
	CIS_TEST_OK	Command interface test passed.
	CIS_WMARK	Number of free command descriptors below watermark.
	CIS_ERRCNT	Command interface error count.

Hardware	HW_TEST_OK	Hardware test has passed.
	HW_DMA_BZ_0	DMA channel 0 is busy.
	HW_DMA_BZ_1	DMA channel 1 is busy.
	HW_DMA_BZ_2	DMA channel 2 is busy.
	HW_DMA_BZ_3	DMA channel 3 is busy.
	HW_SRE_PBI_HALT	Segmentation and Reassembly Engine currently halted.
	HW_SRE_L1IPQ	Segmentation and Reassembly Engine currently paused due to L1 IPQ discard failure.
	HW_SRE_L2IPQ	Segmentation and Reassembly Engine currently paused due to L2 IPQ discard failure.
	HW_SRE_FREEBUF	Segmentation and Reassembly Engine free buffer list is currently empty.
	HW_IPQ	IPQ is currently not empty.
	HW_PQ_W_PAUSE	PQ write pause previously detected.
	HW_PQ_W_FULL	PQ write full previously detected.
	HW_IFQ_W_PAUSE	IFQ write pause previously detected.
	HW_IFQ_W_FULL	IFQ write full previously detected.
	HW_MEN_BP_TOUT	Memory backpressure timeout previously detected.
	HW_DOWN_BP_TOUT	Downstream backpressure timeout previously detected.
	HW_FBUFF_POOL_WM	Free buffer pool low watermark previously detected.
	HW_PBUF_ERR	Packet buffer error previously detected.
	HW_FM_MSG_HDR	FM message header error previously detected.
	HW_FM_MSG	FM message error previously detected.
	HW_EPG_CTRL_Q	Egress packet generator (EPG) control queue is backed up.
	HW_EPG_MSG_BUF	EPG message buffer error.
	HW_EPG_QREAD_TOUT	EPG read queue timeout.
	HW_EPG_QWRITE_TOUT	EPG write queue timeout.
	HW_EPG_CQ_W_FULL	EPG completion queue write full.
	HW_EPG_MSG_CHKSM	Egress packet generator (EPG) message checksum error.
	HW_EPG_MTLQ_TOUT	EPG MTL queue fetch timeout.
	HW_PEG0	Peg 0 is not used.
	HW_PEG1	Peg 1 is not used.
	HW_PEG2	Peg 2 is not used.
	HW_PEG3	Peg 3 is not used.
	HW_ERRCNT	Hardware error count.

