Performance Endpoints

Windows NT/2000/XP

November 2003

Contents

About This Book and the Library

This brief guide provides conceptual information about the free Performance Endpoint software NetIQ Corporation provides in association with its Network Performance Management products, and covers installation and configuration for one specific endpoint platform. For information about installing and configuring all the endpoint platforms, including HP-UX, IBM AIX, IBM MVS, Linux, Microsoft Windows 95, Windows 98, Windows Me, Windows CE, Windows NT, and Windows 2000, Windows XP, Windows Server 2003, Novell NetWare, Sun Solaris, Compaq Tru64 UNIX, FreeBSD UNIX, IBM OS/2, Linux IA-64, Microsoft Windows 3.1, Windows XP (64-bit), SCO Unixware, SGI IRIX, and Spirent TeraMetrics, is available in the HTML-formatted *Performance Endpoints* Guide in your product's Help system, or you can download other individual endpoint guides in .PDF format from the World Wide Web at http://www.netiq.com/support/pe/pe.asp.

Intended Audience

This book provides information about Performance Endpoint software for users of NetIQ End2End, Chariot, Qcheck, Vivinet Manager, and Vivinet Assessor.

Other Information in the Library

The library provides the following information resources:

User Guides for Chariot and End2End
: Provide general information about each product and guide you through installation and use.

Messages and Application Scripts Guide
: Describes the application scripts included with End2End and Chariot and provides a detailed reference to the error messages for all NetIQ Network Performance Management products.

NetIQ Products

NetIQ Corporation provides integrated products that simplify and unify systems management, security, and network performance management in your extended enterprise. These products also help organizations prepare for and migrate to Windows 2000 and Windows .NET. NetIQ Corporation offers the following solutions:

Performance and Availability Management
: These products allow you to manage, analyze, and report on the health, performance, and availability of your mission-critical Windows and UNIX applications and servers. With these products, you can pinpoint network problems and resolve them quickly and effectively.

Security Management and Administration

These products provide real-time Windows security event consolidation, configuration management, host-based intrusion detection, centralized assessment and incident management, vulnerability assessment and prevention, firewall log analysis and reporting, and Windows security administration. With these products, you can also manage group policy, administration workflows, and permissions on vital assets throughout your enterprise.

VoIP Management and Network Testing

These products enable you to evaluate your network for Voice over IP (VoIP) traffic before deployment, as well as manage and troubleshoot VoIP during and after deployment. With these products, you can also test application or hardware performance and predict the impact of network changes, such as adding users or new applications.

Web Analytics and Management

These products deliver important insight into every element of Web site visitor activity, as well as improved Web site performance and availability. These solutions enhance your e-business performance, resulting in higher returns on infrastructure and marketing investments and improved visitor-to-customer conversion rates.

Windows and Exchange Management

These products enable you to manage all Windows and Exchange essentials, from ensuring optimal availability and performance to seamless migration, secure administration, and in-depth analysis.

Contacting NetIQ Corporation

Please contact us with your questions and comments. We look forward to hearing from you. For support around the world, please contact your local partner. For a complete list of our partners, please see our Web site. If you cannot contact your partner, contact our Technical Support team.

Telephone: 503-223-3023

Email: networkinfo@netiq.com

Support: network-support@netiq.com

Web Site: www.netiq.com

Chapter 1
Microsoft Windows NT, Windows 2000, and Windows XP

This chapter explains the installation, configuration, and operation of the Performance Endpoint software for Microsoft Windows NT, Windows 2000, Windows XP, and Windows Server 2003. (The endpoint has not been renamed to reflect its support for Windows Server 2003.) Separate versions of the endpoint operate on the "x86" and "Alpha" versions of Windows NT. A separate version of the endpoint is also available for the 64-bit version of Windows XP; see our Web site for more information.

- x86 computers are commonly known as "PCs"; they contain CPUs made by Intel, AMD, Cyrix, and others.
- Alpha computers contain CPUs made by Compaq Corporation (formerly Digital Equipment Corporation, or DEC).

This endpoint now supports Chariot testing with the Microsoft Windows XP Tablet PC Edition.

The Performance Endpoint for the Windows 98 operating system has been archived at version 4.3. It will not support new features in recent releases of NetIQ products; however, it is still available on the NetIQ Web site at www.netiq.com/download/endpoints. The endpoint for the Windows Millennium Edition (Me) operating system is packaged with the endpoint for Windows NT/2000/XP, but has slightly different code, hardware and software requirements, and installation instructions.

Installation Requirements for Windows NT/2000/XP Endpoints

Here's what you need to run the endpoint program with Windows NT, Windows 2000, or Windows XP:

- A computer capable of running Windows NT, Windows 2000, Windows XP, or Windows Server 2003 well.

  For x86 computers, this implies a CPU such as an Intel 80486, a member of the Pentium family, or equivalent. A Pentium or better is recommended.

For Alpha computers, any system seems to give good performance.

- 32 MBytes of random access memory (RAM).

  The total RAM requirement depends on the RAM usage of the underlying protocol stack and the number of concurrent connection pairs. For very large tests involving hundreds of connections through a single endpoint, additional memory may be required.

- A hard disk with at least 8 MBytes of space available.

- Microsoft Windows NT version 4.0, Windows 2000, Windows XP, or Windows Server 2003.

  Both the Workstation and Server of these operating systems are supported.

  – for IP Multicast: Windows NT 4.0 with Service Pack 3 (or later), Windows 2000, or Windows XP is required.

  – for IP QoS: Windows 2000 requires the QoS Packet Scheduler.

- The latest service packs for Windows NT. On Windows NT with Service Pack 3, Microsoft Internet Explorer version 4.0 and higher is **required**. Use Service Pack 6a instead. Service Pack 6 is not supported.

See the `README` file for this endpoint to see the latest Microsoft service packs with which we've tested.

You also need compatible network protocol software:

for APPC, one of the following

Three APPC stacks for Windows NT or Windows 2000 are supported by the endpoint.

- IBM Personal Communications version 4.3 (PCOMM for Windows NT, also called eNetworks or SecureWay): runs on x86 computers where its communications APIs are installed.
- IBM Communications Server version 6.0 (for Windows NT and Windows 2000): runs only on the server computer of Communications Server's "split stack" model.
- Microsoft Windows SNA Server for x86: runs on either a client or the server computer of SNA Server's "split stack" model. We recommend version 4.0 of SNA Server for Windows NT 4.0, with the latest service packs.
- SNA Server 4.0 requires a fix to use fully qualified LU names. See the NetIQ Support Web site to download the fix, which we obtained from Microsoft.

for IPX and SPX

IPX and SPX software is provided as part of the network support in the Windows NT, Windows 2000, Windows XP, and Windows Server 2003 operating systems.

Microsoft improved their IPX/SPX support for Windows NT and later versions of Windows using "SPX II." SPX II is also present on Novell NetWare 4.x (or later). SPX II allows a window size greater than 1, and buffer sizes up to the size the underlying transport supports.

The SPX protocol supplied by Microsoft in Windows NT 4.0 is subject to slowdowns when running to itself, that is, with loopback.

Our software does not support connections between Windows NT and OS/2, using IPX or SPX.

for RTP, TCP, and UDP

TCP/IP software is provided as part of the network support with Windows NT, Windows 2000, Windows XP, and Windows Server 2003.

Microsoft's Service Pack 3 for Windows NT 4.0 fixes several TCP/IP bugs; Service Pack 3 (or later) is strongly recommended for users of Windows NT 4.0. Service Pack 3 (or later) is required for IP Multicast testing.

Quality of Service (QoS) support for TCP/IP is part of Microsoft Windows 2000, Windows XP, and Windows Server 2003. On Windows NT, Type of Service is available for UDP and RTP only. See the *User Guide* for Chariot for more information.

We recommend that you get up-to-date with the latest Windows service levels. "Getting the Latest Fixes and Service Updates" on page 19 discusses where to get the latest software upgrades.

There's one version of the endpoint for Windows NT 4.0 on x86 computers, and a separate version for Windows NT on Alpha. This software supports a range of underlying functions, which vary by operating system level and service pack. This functional support is summarized in the table we included in "Endpoint Capabilities" on page 25.

Endpoint Installation

We recommend configuring your networking software—and ensuring that it is working correctly—before installing our software. See the Help for your networking software, and see "Configuring Windows Endpoints" on page 10 for more assistance.

Note

Before installing the endpoint on Windows 2000, plan to close any other network applications. During the endpoint installation, Windows 2000 recycles the protocol stack, causing some client applications to lose connectivity to their servers. Some of these applications don't retry their connectivity before exiting and must be restarted.

The endpoint for Windows NT, Windows 2000, Windows XP, and Windows Server 2003 is installed and runs as a service. Only a user ID with Administrator authority is permitted to install services. To successfully install the endpoint, you must be logged in with Administrator authority. The permissions of the directory where the endpoint is installed must also be set to allow the `SYSTEM` (the operating system) full control access. Be sure to give the System "Full Control" permission on all files in the `NetIQ\Endpoint` directory or the directory where you've installed the endpoint, plus any relevant subdirectories, if any.

The security implementation in Windows Server 2003 differs noticeably from that in earlier versions of Windows. Before you install the endpoint on Windows Server 2003, make sure your user account is running in "Install" mode and not in "Execute" mode. To change the mode so that you have the necessary installation privileges, run the following at a command prompt:

```
change user /install
```

The installation on Windows Server 2003 will fail with the message "`The InstallShield-generated file that allows uninstallation is missing`" if you're trying to install from the wrong mode.

Following are directions for installing the endpoint **from a CD-ROM** and **from the World Wide Web**:

To install the endpoint from a CD-ROM, do the following:

1. Put the CD-ROM in your CD-ROM drive.
2. Go to a command prompt.
   - For x86 computers, go to the directory `WIN32` and enter the following:

     `[drive:]\Endpoint\Win32\gsendw32.exe`

   - For Alpha computers, go to the directory `archive\winnta` and enter the following:

     `[drive:]\Endpoint\archive\Winnta\setup.exe`
3. Select the directory where the endpoint will be installed. We recommend installing it on a local hard disk of the computer you're using. If you install on a LAN drive, the additional network traffic may influence your performance results. The default directory is `\Program Files\NetIQ\Endpoint`, on your boot drive.
4. If you have a previous installation of the endpoint, you will be asked if you want it removed. If you select "**Yes**," the previous installation is removed, and the new installation continues. If you select "**No**," the install program exits with no changes to your existing installation because a new version cannot be added until the old version is removed. The installation program adds the endpoint program as a service.
5. The next dialog box contains three checkboxes.

– Check the first check box to install pre-built data files. We recommend you leave this box checked. You can save a small amount of disk space by not installing the files used for compression testing -- but the defaults in many application scripts specify these files. If these `CMP` files are not installed, many application scripts cannot be used in tests until they are modified.

– Check the second check box to specify an LU alias for Microsoft SNA.

  If you plan to test with APPC using Microsoft SNA Server on this endpoint, check this box. The next screen prompts you to enter the APPC LU alias for this computer. If you need to specify an LU alias for SNA Server later, you can use our software's `SETALIAS` program. See the Support area of our Web site.

  If you enter an APPC LU Alias, it must be defined already at the SNA Server, and must be unique in the network. The LU Alias you enter won't take effect until after the computer is restarted (or the SnaBase service is stopped and restarted).

– Check the third check box to start the endpoint on installation. If you leave the box unchecked, the endpoint starts when you restart the computer. No window is shown while the endpoint is running because it runs as a service.

  A Windows NT, Windows 2000, Windows XP, or Windows Server 2003 service is controlled from the Services dialog inside the Control Panel. If you want to restart a service without restarting Windows, use the Services dialog box. For example, to start `SnaBase`, go to the Services dialog box, select the `SnaBase` line, and click **Start** (or **Play**). The status changes to "started" when the service is successfully started.

  You can also manually start the endpoint after installation. See "Starting the Endpoint" on page 14 for instructions.

6. Finally, you are asked whether you want to install application monitoring support. This option is ***only*** recommended if you're planning to use this endpoint for NetIQ End2End application monitoring. It's ***not*** recommended if you're installing the endpoint on a server. The `README` file contains a list of significant operating restrictions. Click **Next** to accept the default option, which does not install the extra support. The endpoint installation copies the necessary files to your hard disk.

Note
Application monitoring support should ***not*** be installed on a server unless it has been thoroughly tested beforehand. Interaction problems may arise; proceed with caution. Consult the endpoint `README` file for more information.

The installation is now complete; you can remove the CD-ROM from its drive.

To prevent the endpoint from running automatically on startup, see the section titled "Disabling Automatic Startup" on page 17.

When you've completed installation, refer to "Configuring Windows Endpoints" on page 10 to make sure your endpoint is ready for testing and monitoring.

To install an endpoint you've downloaded from the World Wide Web, do the following:

1. Save the `gsendw32.exe` file to a local directory.
2. Use the Windows Explorer to navigate to the file and double-click to start the installation.
3. The first screen after the Setup dialog box lets you select the directory where the endpoint will be installed. We recommend installing it on a local hard disk of the computer you're using. If you install on a LAN drive, the additional network traffic may influence your performance results. The default directory is `\Program Files\NetIQ\Endpoint`, on your boot drive.
4. If you have a previous installation of the endpoint, you will be asked if you want it removed. If you select "**Yes**," the previous installation is removed, and the new installation continues. If you select "**No**," the install program exits with no changes to your existing installation because a new version cannot be added until the old version is removed. It then adds Endpoint (the endpoint program) as a service.
5. The next dialog contains three check boxes.
   - Check the first check box to install pre-built data files. We recommend you leave this box checked. You can save a small amount of disk space by not installing the files used for compression testing -- but the defaults in many application scripts specify these files. If these `.CMP` files are not installed, many application scripts cannot be used in tests until they are modified.
   - Check the second check box to specify an LU alias for Microsoft SNA.

     If you plan to test with APPC using Microsoft SNA Server on this endpoint, check this box. The next screen prompts you to enter the APPC LU alias for this computer. If you need to specify an LU alias for SNA Server later, you can use our software's `SETALIAS` program.
   - If you enter an APPC LU Alias, it must be defined already at the SNA Server, and must be unique in the network. The LU Alias you enter won't take effect until after the computer is restarted (or the SnaBase service is stopped and restarted).

– Check the third check box to start the endpoint on installation. If you leave the box cleared, the endpoint starts when you restart the computer. No window is shown while the endpoint is running, since it runs as a service.

  A Windows NT, Windows 2000, Windows XP, or Windows Server 2003 service is controlled from the Services dialog box inside the Control Panel. If you want to restart a service without restarting Windows, use the Services dialog box. For example, to start `SnaBase`, go to the Services dialog box, select the `SnaBase` line, and click **Start** (or **Play**). The status changes to "started" when the service is successfully started.

  You can also manually start the endpoint after installation. See "Starting the Endpoint" on page 14 for instructions.

6. Finally, you are asked whether you want to install application monitoring support. This option is ***only*** recommended if you're planning to use this endpoint for NetIQ End2End application monitoring. It's not recommended if you're installing the endpoint on a server. The `README` file contains a list of significant operating restrictions. Click **Next** to accept the default option, which does not install the extra support. The endpoint installation copies the necessary files to your hard disk.

Note
Application monitoring support should ***not*** be installed on a server unless it has been thoroughly tested beforehand. Interaction problems may arise; proceed with caution. Consult the endpoint `README` file for more information.

To prevent the endpoint from running automatically on startup, see the section titled "Disabling Automatic Startup" on page 17. If you want to restore the setting later, you must do so manually.

When you've completed installation, refer to "Configuring Windows Endpoints" on page 10 to make sure your endpoint is ready for testing and monitoring.

What Happens During Installation

Here's what happens during the installation steps. Let's say you install the endpoint into the directory `\Program Files\NetIQ\Endpoint`. A directory is created with the following contents:

- The executable programs
- The `README` file
- The directory `Cmpfiles`. This directory contains files with the `.CMP` file extension. These are files containing data of different types, such as typical text or binary data. These files are used by the endpoint as data on `SEND` commands. The different data types can be used to vary the data compression performance of your network hardware and software.

- The file `endpoint.ini`

  See "Endpoint Initialization File" on page 33 for information about tailoring this file for individual endpoints.

- The directory `Updates`

  This directory contains files to support an endpoint upgrade function that works with our End2End product. The directory contains a file called `update.iss`. This file is used by subsequent updates to determine the proper responses for the installation dialogs. See "Customizing endpoint.ini for Windows Endpoints" on page 37 and "UPDATE_SERVER" on page 37 for more information.

The endpoint is installed as a service, which means there's nothing visible while it's running. During installation, the endpoint is configured to automatically start when the system reboots. A service can be controlled from the Services dialog box inside the Control Panel; this process is described in "Running Windows Endpoints" on page 14.

Should you have reason to install an older endpoint, you should delete any safestore files, taking the following steps:

1. Stop the endpoint.
2. Delete the safestore files from the endpoint directory (or from the directory specified by the `SAFESTORE_DIRECTORY` keyword in `endpoint.ini`). Safestore files have an extension of `.q*`; you may delete them using the command delete `*.q*`.
3. Uninstall the current endpoint.
4. Install the desired endpoint.

Unattended Installation

Unattended installation (also called silent installation) is available for the endpoints for Windows. You install an endpoint once, by hand, while the install facility saves your input in an answer file. You can then install that same endpoint silently on other computers, that is, without providing input other than the answer file.

First, run `gsendw32.exe`. An answer file called `update.iss` is created in the `\Updates` subdirectory of the directory where you installed the endpoint.

To perform a silent installation, specify the "`-s`" option on `SETUP`. Make sure the answers documented in the answer file `update.iss` are appropriate for the silent installation. If the `update.iss` file is not in the same directory as `setup.exe`, then specify the path and filename with the "`-f1`" option. For example, here's how to install using the `update.iss` file in the `\Program Files\NetIQ\Endpoint` directory on our `n:` LAN drive:

```
SETUP -s -f1n:\Program Files\NetIQ\Endpoint\update.iss
```

If you don't specify the path and filename with `-f1, the default filename is setup.iss.` Don't mix the `.iss` files among different Windows operating systems because their endpoint installations require slightly different input.

It's common to use unattended install from a LAN drive. Be sure you've copied all of the files for each type of endpoint into a single directory (rather than into separate diskette images), and you've created your initial `update.iss` file from that directory. Unattended install does not keep track of diskette label information, and will need user input if you install from separate disk images. You probably don't want your unattended install to ask you for `n:\disk1\, n:\disk2\`, and so on.

If you're planning to use APPC with the endpoint for Windows NT/2000/XP, do NOT enter an LU alias in your initial installation that would be propagated to all the other Windows computers. All the APPC LU aliases MUST be unique (like IP addresses or MAC addresses). So when doing the initial installation, leave the check box asking about LU alias unchecked. Go back later and create LU aliases using the `SETALIAS` program.

Installing the Windows Endpoint with SMS

See "Distributing Endpoints Using SMS" on page 41 for information on automatically installing (and uninstalling) endpoints, using Microsoft's Systems Management Server (SMS).

Removing the Endpoint Package (Uninstall)

To remove the endpoint package from your hard disk, follow these steps:

1. On the Start menu, click **Settings** and then **Control Panel**.
2. Click on **Add/Remove Programs**. The Add/Remove Programs Properties dialog box is shown.
3. Highlight **NetIQ Endpoint** and press **Add/Remove**. The uninstallation program begins. After the program is completed, the endpoint should be uninstalled.

Removing the Endpoint Manually

If the uninstallation program is unable to uninstall the endpoint, you will need to manually uninstall it. For detailed instructions on manually removing the endpoints, see the Performance Endpoints FAQ page in the Knowledge Base on our Web site at www.netiq.com/support/pe/default.asp.

Configuring Windows Endpoints

The endpoint program uses the network application programming interfaces, such as Sockets and APPC, for all of its communications. The endpoint dynamically configures its own programs, so you do not have to update the configuration files for your communications software. However, your communications software must be configured and running correctly. The following steps guide you through this verification process.

1. Determine the network addresses of the computers to be used in tests.
2. Select a service quality.
3. Verify the network connections.

The following sections describe how to accomplish these steps for Windows NT, Windows 2000, or Windows XP:

- "Windows NT and Windows 2000 Configuration for APPC" on page 10
- "Windows NT, Windows 2000, or Windows XP Configuration for IPX and SPX" on page 12
- "Windows NT, Windows 2000, or Windows XP Configuration for TCP/IP" on page 13

Windows NT and Windows 2000 Configuration for APPC

APPC has not been tested on Windows XP and may not be supported. On Windows NT or Windows 2000, the endpoint supports three APPC stacks:

- IBM Personal Communications AS/400 version 4.3 (for Windows NT)
- IBM Personal Communications version 5.0 (for Windows 2000)
- IBM Communications Server version 6.0 (for Windows NT and Windows 2000)
- Microsoft Windows SNA Server version 4.0 (for Windows NT 4.0)

Detailed directions for configuring these APPC protocol stacks can be found in the endpoint documentation section of our support Web site, located at www.netiq.com/support/.

IBM has created a thorough (but aging) "redbook" to assist in setting up APPC across a variety of platforms. This guide is called the *MultiPlatform APPC Configuration Guide* and can be viewed or downloaded from the Web at: www.redbooks.ibm.com/pubs/pdfs/redbooks/gg244485.pdf.

APPC TP Name

APPC applications use an *LU name* to decide which computer to connect to in a network. They use a *TP name* to decide which application program to connect to within a computer.

Our software uses the string **GANYMEDE.CHARIOT.ENDPOINT** as its TP name. This TP name is used when communicating with endpoints via an APPC connection.

Testing the APPC Connection

Now that you know the LUs and modes you are using, you can run a quick check using a program named APING.

APING is a small application packaged with most APPC stacks. It is similar to Ping in TCP/IP; it is an echo program that sends a block of data to another computer. That computer receives the data and sends it back. APING verifies that APPC is correctly installed at a pair of computers, that they are connected to the network, and that it is possible to get an APPC session using the mode you have selected.

To run APING, go to the IBM Communications Server or IBM Personal Communications Programs folder:

1. Select **Utilities**.
2. Select **APPC** and **CPI C Utilities**.
3. Select **Check Connection APING**.

Enter the LU name of the partner you want to connect with. You might want to try entering your own local LU name the first time, just to see how it works. Click **Start**, or click **Start** on the Action menu. It uses the mode name #INTER, by default. (In our software, the mode name is known as the "*service quality*.") If APING works, APING shows a table of timing information. This endpoint should be ready for APPC testing. Continue testing connections to the other endpoints you will use.

If you get any other APPC return code, you have a configuration problem somewhere. You should correct this before starting to run our software.

Make sure that you can run APING successfully from the Chariot or Qcheck Console or the End2Endserver to each computer serving as Endpoint 1, and between each pair of endpoints involved in a test, before starting your testing with APPC.

If you create a connection pair with the same Windows NT or Windows 2000 computer running APPC configured as both Endpoint 1 and Endpoint 2 (that is, a loopback connection), the endpoint returns message **CHR0182** to indicate an error.

Windows NT, Windows 2000, or Windows XP Configuration for IPX and SPX

To use the IPX or SPX protocol in tests, IPX addresses must be supplied as the network address when adding a connection pair. IPX addresses consist of a 4-byte network number (8 hexadecimal digits) followed by a 6-byte node ID (12 hex digits). A colon separates the network number and node ID. The 6-byte node ID (also known as the *device number*) is usually the same as the MAC address of the LAN adapter you're using.

In Chariot, it's tedious to enter IPX addresses when adding new connection pairs. When using the IPX or SPX protocol in your tests, our software can maintain an easy-to-remember alias in the Edit Pair dialog. You can set up the mapping once, and use the alias names ever after. The underlying file, named `spxdir.dat`, is like the `HOSTS` file used in TCP/IP, or the LU alias definitions offered with APPC.

For Win32 operating systems, endpoints make WinSock version 1.1 Sockets-compatible calls when using the IPX or SPX network protocol.

Determining Your IPX Network Address

To determine a Windows computer's local IPX address, enter the following at a command prompt:

```
IPXROUTE CONFIG
```

If your IPX software support is configured correctly, your output will look like the following (this output is taken from Windows NT 4.0):

```
NWLink IPX Routing and Source Routing Control Program v2.00
net 1: network number 00000002, frame type 802.2, device AMDPCN1
(0207011a3082)
```

The 8-digit network number is shown first; here, it's `00000002`. The 12-digit node ID is shown in parentheses at the end; here it's `0207011a3082`, which is our Ethernet MAC address. Thus, the IPX address to be used in tests is `00000002:0207011a3082`.

Another method: if you already know the IP address of a computer—and thus can Ping to that computer—it's easy find its MAC address. First, Ping to the target computer from a computer on the same network segment, using its IP address. Then, enter the following command:

```
arp -a
```

A list of recently cached IP addresses is shown, along with their MAC addresses if they are LAN-attached. The `arp` command only reports the physical address of computers it can reach without crossing a router. It also won't give you the physical address of the local computer.

Stopping Connections Doing SPX Loopback

A Chariot Console user can observe that stopping can take between 20 and 50 seconds when running connections using SPX on Windows NT, doing loopback (that is, both endpoints have the same address). If the endpoint is on a `Receive` call, the protocol stack can pause for almost a minute before returning.

Windows NT, Windows 2000, or Windows XP Configuration for TCP/IP

The RTP, TCP, and UDP protocols use TCP/IP software for network communications. TCP/IP offers two forms of network addresses: IP addresses and domain names. An IP address is a 32-bit numeric address. It is represented in dotted notation as a set of four numbers separated by periods, such as `199.72.46.202`. An alternative, domain names are in a format that is easier to recognize and remember, such as www.netiq.com. To use domain names, you need either a Domain Name Server (DNS) set up in your network or an `/etc/hosts` file on each computer.

Determining Your IP Network Address

To determine a Windows NT, Windows 2000, or Windows XP computer's local IP address, enter the following command:

```
IPCONFIG
```

If your TCP/IP stack is configured correctly, your output will look like the following (this output is taken from Windows NT 4.0):

```
Windows NT IP Configuration
Ethernet adapter AMDPCN1:
IP Address. . . . . . . . . : 10.10.44.3
Subnet Mask . . . . . . . . : 255.255.255.0
Default Gateway . . . . . . : 10.10.44.254
```

Its local IP address is shown in the first row; here it's `10.10.44.3`.

You can also find your IP address using the graphical user interface. Select the **Control Panel** folder, and double-click on the **Network** icon. The installed network components are shown. Double-click **TCP/IP Protocol** in the list to get to the **TCP/IP Configuration**. Your IP address and subnet mask are shown.

To determine a Windows NT, Windows 2000, or Windows XP computer's local hostname, enter the following command:

```
HOSTNAME
```

The current hostname is shown in the first row.

From the graphical user interface, return to the TCP/IP Protocol configuration. Select **DNS** (Domain Name System) to see or change your domain name. If the DNS Configuration is empty, avoid using domain names as network addresses; use numeric IP addresses instead.

Testing the TCP/IP Connection

Ping is a simple utility program, included in all TCP/IP implementations. To check the connection from one computer to another, enter the following at an MS-DOS command prompt:

```
ping xx.xx.xx.xx
```

Replace the x's with the IP address of the target computer. If Ping returns a message that says "`Reply from xx.xx.xx.xx ...`," the Ping worked. If it says "`Request timed out`," the Ping failed, and you have a configuration problem.

Make sure that you can run Ping successfully from the Chariot or Qcheck Console or the End2End server to each computer serving as Endpoint 1, and between each pair of endpoints involved in a test, before starting your testing with TCP/IP.

Running Windows Endpoints

The following topics describe starting and stopping an endpoint in the Windows NT, Windows 2000, Windows XP, or Windows Server 2003 operating systems, as well as some of the messages and information that become available during testing with this endpoint. The Windows endpoint is controlled from the Services dialog box. For Windows NT 4.0 or click **Settings**, then **Control Panel** on the Start menu, then double-click **Services**. For Windows 2000/XP/2003, click **Settings**, then **Control Panel** on the Start menu, double-click **Administrative Tools**, and then double-click **Services**. The Services dialog box lets you start or stop the endpoint, listed as "NetIQ Endpoint."

Only a user ID with Administrator authority is permitted to start or stop Windows NT, Windows 2000, Windows XP, or Windows Server 2003 services.

Starting the Endpoint

By default, the endpoint program is configured to start automatically, which means that you will not see a window for the program when it is running. Because the endpoint runs as a service, you do not have to be logged into your workstation for the endpoint to run.

If you stop the endpoint service, you can restart it without restarting Windows. There are two ways to restart the endpoint service:

1. At a command prompt, enter:

   ```
   net start netiqendpoint
   ```

2. In the Services dialog box, select **NetIQ Endpoint** and click **Start** (or **Play**). The status changes to "started" when the endpoint is successfully started.

Note
A single running copy of the endpoint service handles one or multiple concurrent tests.

Stopping a Windows Endpoint

There are two ways to stop the endpoint service:

- At a command prompt, enter the following:

```
net stop netiqendpoint
```

- In the Services dialog box, click **NetIQ Endpoint** and click **Stop**. The status is blank when the endpoint program has stopped.

Disable Your Screen Saver

Screen savers in Windows NT and Windows 2000 can significantly lower the throughput that's measured by an endpoint. We recommend disabling your screen saver at endpoint computers while running tests.

The SetAddr Utility

Endpoints for Windows operating systems now ship with a utility that helps you quickly create virtual IP addresses on Windows NT, Windows 2000, Windows XP (32-bit and 64-bit), and Windows Server 2003 endpoint computers. Virtual addresses are chiefly useful when you're testing hundreds or even thousands of endpoint pairs using only a few computers as endpoints. To all intents and purposes, the traffic on the network is identical, whether you're using "real" or virtual addresses.

For more information about creating virtual addresses, consult "Configuring Virtual Addresses on Endpoint Computers" in the *User Guide* for Chariot.

When you install a Windows endpoint, `Setaddr.exe` for 32-bit Windows is automatically installed in the same directory. For 64-bit Windows, a 64-bit version of `Setaddr.exe` is installed. The two versions of SetAddr cannot be used across operating systems with different architectures.

The usage is as follows:

```
setaddr [-dr] -a N -f Addr -t Addr -i Addr -s Addr
| -l[a]
| -da
| -ds -f Addr -s Addr
```

(where "`N`" indicates the adapter number of the NIC card you're assigning virtual addresses to, and "`Addr`" indicates the virtual addresses or subnet mask you're assigning to it).

Options:

```
-l     List all network adapters
-la    List all network adapters and their IP addresses
-a     Adapter to modify (number given by -l options)
-dr    Delete a range of addresses
-da    Delete all addresses
-ds    Delete a single address
-f     From address
-t     To address
-i     Increment by
-s     Subnet Mask
```

The `-d` flags cannot be used to delete a computer's primary IP address.

The `-i` flag lets you determine how the range of addresses will be created. This is an optional field; by default, SetAddr increments the range by one in the final byte only. This "`increment by`" value is represented as "`0.0.0.1`". Enter a value (0-255) for each byte of the 4-byte IP address. A value of `1` specifies that the address values in that byte will be incremented by one when SetAddr creates the range. For example, enter

```
setaddr -f 10.40.1.1 -t 10.40.4.250 -i 0.0.1.1 -s 255.255.0.0
```

SetAddr creates 1000 virtual addresses.

Known Limitations:

- IPv4 only.
- Windows NT, Windows 2000, Windows XP, and Windows Server 2003 computers only.
- SetAddr only works on computers with fixed IP addresses. DHCP-enabled adapters can't be used.
- You must restart the computer to whose NIC you've assigned virtual IP addresses before you begin testing with that computer. SetAddr modifies some Windows Registry keys, and restarting is required for the changes to take effect.
- The number of virtual addresses you can assign to a single adapter depends on the protocol stack and the size of the Windows Registry. We benchmarked measurements using computers running up to 2500 virtual addresses, which is a recommended limit.
- No checking is done to ensure that thousands of addresses are not being created. Be careful! More TCP/IP stack resources are required to manage virtual addresses.
- You may only add Class A, B, and C virtual IP addresses. Loopback addresses and Class D and E IP addresses are invalid. Valid address ranges, then, are `1.x.x.x` to `233.x.x.x`, excluding `127.x.x.x`.

- When more than 2250 virtual address are defined on Windows 2000 computers, all the LAN adaptor icons disappear from the Network and Dial-up Connections dialog box in My Network Places. You can still see the adaptors by invoking `ipconfig` or `setaddr` from the command line, and the addresses are still reachable. Removing some virtual addresses so that fewer than 2250 were specified and restarting the computer solved the problem.

Disabling Automatic Startup

To disable the automatic starting of the endpoint, take the following steps in Windows 2000:

1. On the Start menu, click **Settings**, then **Control Panel**, then **Administrative Tools**, then **Services**. The Services dialog box appears.
2. Double-click **NetIQ Endpoint**.
3. On the Startup type menu, click **Manual**.
4. Click **OK** to save the new setting and exit the dialog box. The endpoint will no longer start automatically when you restart the computer. However, you can manually start the endpoint.

Take the following steps in Windows NT:

1. On the Start menu, click **Settings** and then **Control Panel**. The Control Panel appears.
2. Double-click the **Services** icon.
3. Highlight **NetIQ Endpoint** and click **Startup**.
4. Click **Manual**.
5. Click **OK** and then **Close**. The endpoint will no longer start automatically when you restart the computer. However, you can manually start the endpoint.

How to Tell If a Windows Endpoint Is Active

The status field in the Services dialog box shows whether the NetIQ Endpoint service has started.

Similarly, the Windows Performance Monitor program can be used to look at various aspects of the endpoint. Start Performance Monitor by double-clicking its icon in the Administrative tools group. Click **Add to Chart** on the Edit menu. Select the **Process** object and the **Endpoint** instance. Then add the counters you are interested in, such as thread count or % of processor time. In the Steady state (that is, no tests are active), Thread Count will show about 6 threads active for the endpoint; the answer depends on the number of protocols in use.

Logging and Messages

While most error messages encountered on an endpoint are returned to the Chariot or Qcheck Console or the End2End server, some may be logged to disk. Errors are saved in a file named `ENDPOINT.LOG`, in the directory where you installed the endpoint. To view an error log, use the command-line program named `FMTLOG.EXE`. The program `FMTLOG.EXE` reads from a binary log file, and writes its formatted output to `stdout`. Use the following `FMTLOG` command:

```
FMTLOG log_filename > output_file
```

This endpoint performs extensive internal cross-checking to catch unexpected conditions early. If an assertion failure occurs, the file `assert.err` is written to the directory where you installed the endpoint.

Application Monitoring Support with Check Point VPN Software

Near the end of the endpoint installation, you are asked if you want to install application monitoring support. This support includes a .DLL inserted into the protocol stack and is only recommended for End2End customers who plan to run application monitoring. Occasionally you may see minor interaction problems with other software; the `README` file for the Windows endpoint has a full list of known interaction issues.

We've found that the Check Point SecuRemote VPN client may not work after you install endpoint application monitoring support on Windows NT 4.0 (with Service Pack 4 or Service Pack 6a). It works well on Windows 2000. For Windows NT 4.0, a possible workaround is to install Microsoft's Remote Access Service (RAS) and the endpoint first, and then install the Check Point VPN Client. More information is available in the endpoint `README` and in the *User Guide* for End2End; refer to the "Troubleshooting" chapter.

Getting the Latest Fixes and Service Updates

We've found that communications software is often fragile. Its developers are constantly working to make it more robust, as the software gets used in an ever-wider set of situations.

We therefore recommend working with the very latest software for the underlying operating system and communications software. Here are the best sources we've found for the Windows NT, Windows 2000, Windows XP, or Windows Server 2003 software used by the endpoint program.

Updates and Information for Windows

Microsoft posts code and driver updates to the following Web site: www.microsoft.com/windows/downloads/.

For information about configuring TCP/IP to make it work better on Windows NT, consult the following Web site: www.microsoft.com/windows2000/techinfo/howitworks/communications/networkbasics/tcpip_implement.asp.

Updates for Microsoft SNA Server

Microsoft posts code and driver updates to the following Web site: http://support.microsoft.com/support/sna/sp.asp.

Updates for IBM SNA Software for Windows

For information on IBM's Personal Communications (PCOMM) family of software, see: www.software.ibm.com/network/pcomm/support/.

Chapter 2
Performance Endpoints

This guide contains information about the Performance Endpoints, which are available for more than 20 different operating systems.

All the information you need to install, configure, and run the endpoints in your network is included here and in the printed version of the *Performance Endpoints* guide. In addition to topics discussing issues common to all the endpoints, these guides also contain information about each operating system, organized in separate chapters.

Endpoint Requirements and Capabilities

The following topics describe the software and hardware requirements and the supported functions of the Performance Endpoints, version 5.0.

The latest version of the endpoint software can always be downloaded free from the Internet. A single installable file is available for each operating system. Endpoints are available for downloading at www.netiq.com/support/pe/pe.asp.

You cannot run endpoint software from a CD-ROM; you must install it on a computer.

Operating System and Protocol Stack Support

The following tables list the software with which we have tested the Performance Endpoints for each operating system.

Note
Versions listed are the **earliest**, not necessarily the only, versions supported.

Endpoint	OS version	TCP, UDP, RTP	IP Multicast version	IPX/SPX stack	APPC stack version
Cobalt RaQ/RaQ2 (MIPS)	Linux v. 2.0 for MIPS	included	kernel 2.0.32	no	no
Cobalt RaQ3 (x86)	kernel 2.0.32	included	kernel 2.0.32	no	no
Compaq Tru64 UNIX	Digital UNIX 4.0B or Compaq Tru64 Unix for Alpha	included	v4.0B	no	no
FreeBSD UNIX	BSD v3.1	included	v3.1	no	no
HP-UX	HP-UX v10.10	included	v10.10	no	no
IBM AIX	AIX v4.1.4	included	v4.1.4	no	no
IBM MVS	MVS/ESA SP v4R2.2	See “MVS TCP/IP Stacks”	no	no	IBM ACF/VTAM for MVS/ESA v3R4.2
IBM OS/2	OS/2 Warp 4, Warp Connect 3	Download TCP 4.1	Download TCP 4.1	Download Novell Netware Client v2.12	IBM CommServer for OS/2 v4.1
Linux (x86 and MIPS)	kernel 2.0.32	included	kernel 2.0.32	no	no
Linux IA-64	kernel 2.4.0test7-42	included	kernel 2.4.0test7-42	no	no
Microsoft Windows 3.1	Windows 3.1 or Windows for Workgroups 3.11	see “Microsoft Windows 3.1 TCP/IP Stacks”	Chameleon 7.0, as E2	no	no
Microsoft Windows 95	Windows 95	included	no	Download Novell Netware Client v3.21	IBM PComm v4.3 for Windows 95

Endpoint	OS version	TCP, UDP, RTP	IP Multicast version	IPX/SPX stack	APPC stack version
Microsoft Windows 95 with WinSock 2	Windows 95 with WinSock 2 installed	Download WinSock 2	included	included	IBM PComm v4.3 for Windows 95
Microsoft Windows 98	Windows 98	included	included	included	IBM PComm v4.3 for Windows 98
Microsoft Windows Millennium Edition (Me)	Windows Me	included	included	included	IBM PComm v4.3 for Windows 98
Microsoft Windows NT 4	Windows NT SP 4	included	SP3 (IGMPv1) SP4 (IGMPv2)	included	IBM PComm v4.3, or IBM CommServer v5.0 (for Windows NT), or Microsoft SNA Server v4.0s for Windows NT
Microsoft Windows NT 4 for Alpha	Windows NT4 SP 3	included	SP3 (IGMPv1) SP4 (IGMPv2)	included	Microsoft SNA Server for Alpha v4.0 with SP1 or v3.0 with SP2
Microsoft Windows 2000	Windows 2000	included	included	included	IBM PCOMM version 5.0, or IBM CommServer v6.0
Microsoft Windows XP	Windows XP (32-bit)	included	included	included	IBM PCOMM version 5.0, or IBM CommServer v6.0
Microsoft Windows XP (64-bit)	Windows XP (64-bit)	included	included	no	no
Novell NetWare	v3.12	included	v4.0	included	no

Endpoint	OS version	TCP, UDP, RTP	IP Multicast version	IPX/SPX stack	APPC stack version
SCO UnixWare	UnixWare v2.1	included	v7.0	no	no
SGI IRIX	IRIX v6.2 with patches	included	v6.2	no	no
Spirent Terametrics	kernel 2.2.11	included	kernel 2.2.11	no	no
Sun Solaris for SPARC	Solaris v2.4	included	v2.4	no	no
Sun Solaris for x86	Solaris v2.4	included	v2.4	no	no

Microsoft Windows 3.1 TCP/IP Stacks

The Microsoft Windows 3.1 Performance Endpoint software supports the following TCP/IP stacks:

- Microsoft 32-bit stack, shipped on the Windows NT 4.0 Server CD-ROM
- Frontier Technologies *SuperTCP* v2.2
- FTP Software *OnNet for Windows* v2.1
- NetManage *Chameleon NFS* v4.6.3 (IP Multicast support requires version 7.0 or later)
- Novell Client 3.1 for DOS and Windows 3.x v2.71
- Novell *Client for DOS/Win* (VLMs) v1.21
- WRQ TCP Connection for Windows v5.1

Because Windows 3.x lacks thread support, you cannot use the Windows 3.1 endpoint as Endpoint 1 in an IP Multicast test.

MVS TCP/IP Stacks

The MVS Performance Endpoint software (archived at endpoint version 4.4) supports the following TCP/IP stacks:

- TCP/IP versions 3.2 through 3.8, from IBM. Version 2.6 of OS/390 (TCP/IP version 3.5) and higher includes support for IP Multicast testing with Chariot.
- *SOLVE:TCPaccess* versions 4.1 and 5.2 stack from Sterling Software. A set of PTFs is required for operation with version 4.1.

Endpoint Capabilities

The following table indicates which endpoints have been tested with and are supported by NetIQ products. Shaded rows indicate endpoints that have been archived at previous versions. For more details on specific product capabilities, see the topics below.

NetIQ Product Endpoint	Qcheck	Chariot	End2End	Vivinet Suite
Compaq Tru64 UNIX	Yes	Yes	Yes	No
FreeBSD UNIX	Yes	Yes	Yes	No
HP-UX	Yes	Yes	Yes	No
IBM AIX	Yes	Yes	Yes	No
IBM MVS, Windows install	Yes	Yes	Yes	No
IBM OS/2	Yes	Yes	Yes	No
Linux for Cobalt RaQ/RaQ2 (MIPS)	Yes	Yes	Yes	No
Linux for Cobalt RaQ3 (x86)	Yes	Yes	Yes	Yes
Linux x86 (TAR)	Yes	Yes	Yes	Yes
Linux x86 (RPM)	Yes	Yes	Yes	Yes
Linux IA-64 (TurboLinux)	Yes	Yes	Yes	No
Microsoft Windows 95	Yes	Yes	Yes	No
Microsoft Windows 98	Yes	Yes	Yes	Assessor only
Microsoft Windows Me/NT/2000/XP	Yes	Yes	Yes	Yes
Microsoft Windows 98 (Web-Based)	Yes	Yes	No	Assessor only
Microsoft Windows Me/NT/2000/XP (Web-Based)	Yes	Yes	No	Assessor only
Microsoft Windows XP (64-Bit)	Yes	Yes	Yes	No

NetIQ Product	Qcheck	Chariot	End2End	Vivinet Suite
Endpoint				
Microsoft Windows 3.1	Yes	Yes	Yes	No
Novell NetWare	Yes	Yes	Yes	No
SCO UnixWare	Yes	Yes	Yes	No
SGI IRIX	Yes	Yes	Yes	No
Spirent Communications TeraMetrics	Yes	Yes	Yes	Assessor only
Sun Solaris (SPARC)	Yes	Yes	Yes	Yes
Sun Solaris Endpoint (x86)	Yes	Yes	Yes	Yes

Endpoints for Windows 2000 and Windows XP also support testing with IPv6. Refer to the following topic, "Performance Endpoint Support for Chariot Functions" below for more information.

Performance Endpoint Support for Chariot Functions

The following table describes the Performance Endpoint capabilities for the supported operating systems. Shaded rows indicate endpoints that have been archived at previous versions. These endpoints may not support functionality new in the latest versions of NetIQ Chariot.

Endpoint OS	IP QoS (DiffServ, GQOS, TOS)	Traceroute	CPU Util.	VoIP Test Module	IPv6 Test Module
Cobalt RaQ or RaQ2 (MIPS)	TOS	No	Yes	No	No
Cobalt RaQ3 (x86)	TOS	Yes	Yes	Yes	No
Compaq Tru64 UNIX	TOS	No	Yes	No	No
FreeBSD UNIX	TOS	No	Yes	No	No
HP-UX	TOS	Yes	Yes	No	No
IBM AIX	TOS	Yes	Yes	No	No
IBM MVS	No	No	No	No	No
IBM OS/2	TOS	No	Yes	No	No
Linux	TOS	Yes	Yes	Yes	Yes
Linux IA-64	TOS	Yes	Yes	No	No

Endpoint OS	IP QoS (DiffServ, GQOS, TOS)	Traceroute	CPU Util.	VoIP Test Module	IPv6 Test Module
Microsoft Windows 3.1	No	No	No	No	No
Microsoft Windows 95	No	No	Yes	No	No
Microsoft Windows 95 with WinSock 2	TOS (UDP, RTP)	Yes	Yes	No	No
Microsoft Windows 98	GQOS (RSVP), TOS (UDP, RTP)	Yes	Yes	Yes	No
Microsoft Windows Me	GQOS (RSVP)	Yes	Yes	Yes	No
Microsoft Windows NT 4	TOS (UDP, RTP)	Yes	Yes	Yes	No
Microsoft Windows NT 4 for Alpha	No	Yes	Yes	No	No
Microsoft Windows 2000	DiffServ, GQOS, TOS (via Registry)	Yes	Yes	Yes	No. See “IPv6 Test Module Support”
Microsoft Windows 98 (Web-Based)	Yes	No	Yes	Yes	No
Microsoft Windows Me/NT/2000/XP (Web-Based)	Yes	No	Yes	Yes	No
Microsoft Windows XP	DiffServ, GQOS, TOS (via Registry)	No	Yes	Yes	Yes. See “IPv6 Test Module Support”
Microsoft Windows XP (64-bit)	DiffServ, GQoS, TOS	No	No	Yes	No
Novell NetWare	No	No	No, v3.12; Yes, v4.0	No	No
SCO UnixWare	TOS (bits 3-5)	No	No	No	No
SGI IRIX	TOS	No	Yes	No	No
Spirent Terametrics	TOS	Yes	Yes	Yes	No

Endpoint OS	IP QoS (DiffServ, GQOS, TOS)	Traceroute	CPU Util.	VoIP Test Module	IPv6 Test Module
Sun Solaris for SPARC	TOS	Yes	Yes	Yes	No
Sun Solaris for x86	TOS	Yes	Yes	Yes	No

IPv6 Test Module Support

Currently, testing with version 6 of the Internet Protocol (IPv6) is only supported on endpoints for Windows XP (32-bit only) and Red Hat Linux, versions 8.0 and higher. You must first install IPv6 support on these endpoints before you begin testing. You must also purchase the separately licensed IPv6 Test Module for Chariot.

In addition, Windows 2000 provides unofficial support for IPv6, but it requires a patch called the "Microsoft IPv6 Technology Preview for Windows 2000 Network Protocol Stack." You can download it from http://msdn.microsoft.com/Downloads/sdks/platform/tpipv6/readme.asp.

Performance Endpoint Support for End2End Functions

The following table shows which endpoint platforms are supported for End2End functions, particularly for the different types of monitoring End2End performs. Shaded rows indicate endpoints that have been archived at previous versions. These endpoints may not support functionality new in the latest version of NetIQ End2End.

Endpoint OS	System Mon.	Application Mon.	Network Mon.	Service Mon.	Traceroute	Auto-Upgrade
Cobalt RaQ or RaQ2 (MIPS)	No	No	Yes	No	No	No
Cobalt RaQ3 (x86)	No	No	Yes	No	Yes	No
Compaq Tru64 UNIX	No	No	Yes	No	No	No
FreeBSD UNIX	No	No	Yes	No	No	No
HP-UX	Yes	No	Yes	No	Yes	No
IBM AIX	Yes	No	Yes	No	Yes	No
IBM MVS	No	No	Yes	No	No	No
IBM OS/2	No	No	Yes	No	No	No
Linux	No	No	Yes	No	Yes	No
Linux IA-64	No	No	No	No	Yes	No

Endpoint OS	System Mon.	Application Mon.	Network Mon.	Service Mon.	Traceroute	Auto-Upgrade
Microsoft Windows 3.1	No	No	Yes	No	No	No
Microsoft Windows 95	Yes	Yes	Yes	No	No	Yes
Microsoft Windows 95 with WinSock 2	Yes	Yes	Yes	No	Yes	Yes
Microsoft Windows 98	Yes	Yes	Yes	Yes	Yes	Yes
Microsoft Windows Me	Yes	Yes	Yes	Yes	Yes	Yes
Microsoft Windows NT 4	Yes	Yes	Yes	Yes	Yes	Yes
Microsoft Windows NT 4 for Alpha	Yes	No	Yes	No	Yes	No
Microsoft Windows 2000	Yes	Yes	Yes	Yes	Yes	Yes
Microsoft Windows 98 (Web-Based)	No	No	No	No	No	No
Microsoft Windows Me/NT/2000/XP (Web-Based)	No	No	No	No	No	No
Microsoft Windows XP	Yes	Yes	Yes	Yes	Yes	Yes
Microsoft Windows XP (64-bit)	No	No	Yes	No	Yes	No
Novell NetWare	No	No	Yes	No	No	No
SCO UnixWare	No	No	Yes	No	No	No
SGI IRIX	No	No	Yes	No	No	No
Spirent Terametrics	No	No	Yes	No	Yes	No
Sun Solaris for SPARC	Yes	No	Yes	No	Yes	No
Sun Solaris for x86	Yes	No	Yes	No	Yes	No

Endpoint Computer Resource Guidelines

Determining the computer requirements for a given endpoint can be challenging. There are many variables involved, such as processor speed, operating system, protocol stack, memory, disk space, and the underlying network.

To determine your computer requirements, you must first define how you plan to use Chariot or End2End. The type of information you need depends upon your usage. The following topics provide recommended endpoint computer specifications according to different testing scenarios.

Generating Maximum Throughput

The main factors in getting the most throughput from a computer are CPU speed and memory. You need a CPU that is fast enough to match your network capacity, and with enough memory to hold the code and data used for the test. For best throughput, we recommend using a 32-bit (or better) operating system. The memory you need is based on your operating system. Make sure that you have enough memory at the endpoints so that no swapping takes place while running a test. The following table shows some guidelines in determining the best CPU for different network speeds.

Throughput	Recommended computer
less than 100 Mbps	PCI-based computer with a 32-bit operating system
100 to 200 Mbps	Pentium 166 or greater (consider multiple concurrent pairs)
200 to 500 Mbps	Pentium II or greater (consider multiprocessors)
over 500 Mbps	latest Pentium III or equivalent, with the latest NICs (consider multiprocessors)

The following observations may help guide your throughput testing.

- Windows NT, Windows 2000, Windows XP, and Linux yield the highest throughput. If you test on one of the Windows OSs with the Chariot benchmark script called `High_Performance_Throughput`, the endpoints can make use of Microsoft's WinSock 2 overlapped I/O to achieve much greater throughput on high-speed networks (100 MB and faster). In a test of Gigabit Ethernet throughput using Windows 2000 Server and two Pentium III computers, each having two 933-MHz processors, 1 Gigabyte of RAM, and a single Gigabit NIC, we generated 943 Mbps with six pairs.
- We have also observed some improvements in throughput measurements after changing the `TcpWindowSize` setting in the Windows NT Registry to `65536`. You can set this parameter in the following Registry key:

  ```
  HKEY_LOCAL_MACHINES\SYSTEM\CurrentControlSet\Services\Tcpip
  \Parameters
  ```

  Set the key as a type `REG_DWORD`.

  Refer to the Windows NT Resource Kit for more information.

Calculating Memory Requirements

Endpoints are designed to run in any computer that has sufficient memory to run the operating system well. If you plan to use multiple pairs on a single computer, you may want to calculate the number of pairs that will run without causing the operating system to swap either code or data.

The following table can be used to plan for multiple pairs. The Base RAM column indicates the amount of memory that is allocated by the endpoint before running any pairs. If the endpoint is not being used, this amount may go toward zero if the operating system supports swapping. The protocol columns indicate the amount of memory required for a pair of that protocol.

Operating System	Base RAM (in KB)	TCP KB/pair	RTP or UDP KB/pair	SPX KB/pair	IPX KB/pair	APPC KB/pair
MVS	666	25-48	24-52	n/a	n/a	n/a
NetWare	1100	80-110	320-340	70-100	260-280	n/a
OS/2	1096	50-65	150-170	315-340	150-170	65-90
UNIX (AIX)	1176	132-284	146-296	n/a	n/a	n/a
Windows 3.1	550	72-600	72-600	n/a	n/a	n/a
Windows 95/98/Me	1100	40-65	100-145	40-65	55-75	n/a
Windows NT/2000/XP	2076	35-60	160-180	35-60	160-180	n/a

These RAM usage numbers represent sending with the variable `send_datatype` set to `ZEROS`. Other `send_datatypes` require memory buffers roughly equivalent to the disk space of the .`cmp` file being used. Add 2 KBytes when using `send_datatype = NOCOMPRESS`. See the *Application Scripts* guide for more information on script variables.

Endpoint Pair Capacity

The following table shows some example pair capacities we have tested on various computers. These pairs ran on a 10 Mbps Ethernet LAN. The values in the pairs columns represent the number of pairs this computer supported as Endpoint 2 for a single test. We used the default values for all tests, with two exceptions: for datagram testing, we lengthened the timeout values, as well as the `initial_delay` in test scripts.

This table does not represent the full capacities of these operating systems and stacks, just some representative tests we have run in our test lab.

Operating System	Installed RAM	TCP pairs	RTP or UDP pairs	SPX pairs	IPX pairs	APPC pairs
AIX 4.1	64 MB	200	180	n/a	n/a	n/a
NetWare 4.12	64 MB	500	200	100	100	n/a
OS/2 4.0	32 MB	500	200	20	20	500
Windows 3.1	8 MB	1	1	n/a	n/a	n/a
Windows 95/98/Me	16 MB	18	100	40	175	n/a
Windows NT/2000/XP	32 MB	500	100	300	100	200

Notes

- On Windows NT and Windows 2000, APPC pairs were run using Microsoft SNA Server.
- On Windows 95, Windows 98, and Windows Me, SPX and IPX pairs were run using Novell Client32 for SPX and IPX.
- On OS/2 4.0, IPX and SPX pairs were run using Novell Client for OS/2.

Chariot now supports larger tests under certain conditions using TCP. See "Internet-Scale Testing" in the *User Guide* for Chariot for more information about requirements for large tests.

Endpoint Versions

With each new release of NetIQ Chariot and End2End, the endpoints are updated to support new functionality. However, because some endpoint operating systems are rarely used or provide limited support for Chariot and End2End features, such as End2End service monitoring, endpoints for a few operating systems have been archived. These endpoints are still made available on the Performance Endpoints CD-ROM and on the NetIQ Web site; however, they may not support the latest capabilities of Chariot and End2End. The Endpoint `README` file, included in the root directory of the endpoint CD-ROM, provides a list of all available endpoints and indicates their versions if they are different from the current endpoint level.

Chapter 3
Endpoint Initialization File

An endpoint initialization file is installed with each Performance Endpoint. With this file, you can do the following:

- Restrict the use of this endpoint to specific Chariot or Qcheck Consoles or End2End servers.
- Control which access attempts are logged in an audit file.
- Change the filename of the audit file.
- Enable only particular protocols on this endpoint for setup connections.
- Change the filename of the End2End safestore file.
- Change the location of the endpoint software used for automatic updating.

On most operating systems, this file is named `endpoint.ini` (on MVS, see data set `HLQ.SLQ.JCL(ENDPTINI)`, where "`HLQ`" and "`SLQ`" are the high-level and second-level qualifiers entered during MVS endpoint installation). This file has the same format and structure on all the operating systems.

Here are the default contents of the endpoint initialization file. You can change these keywords and their parameters to tailor individual endpoints for your needs.

Keyword	Parameters
ALLOW	ALL
SECURITY_AUDITING	NONE
AUDIT_FILENAME	ENDPOINT.AUD
ENABLE_PROTOCOL	ALL
SAFESTORE_DIRECTORY	(the directory where the endpoint is installed)
UPDATE_SERVER	endpointupdate.ganymede.com
END2END_SERVER	No default

Note
For the MVS endpoint, the default filename of `ENDPOINT.AUD` is `ENDPTAUD`.

This file is an editable text file. There is a separate copy for each operating system. You might want to make changes to it once, before endpoint installation, which are then incorporated into all the installs for different sets of computers. You can modify this text file before installation by copying the endpoint installation directory for an operating system to a hard drive (preferably a LAN drive), and then modifying the file before running the install from that drive.

We strongly recommend that you make any changes to your `endpoint.ini` files once, before you install any endpoints, as opposed to installing the endpoints and then going back to each of them and separately modifying each one. If you're using Windows (32-bit or 64-bit) endpoints, we've included a utility to help you edit the `endpoint.ini` files before installing the endpoints, should you wish to prepare the endpoints for future automatic upgrades. See "Customizing endpoint.ini for Windows Endpoints" on page 37 for more information.

ALLOW

This keyword determines which Chariot or Qcheck Consoles or End2End servers can run tests using this endpoint.

To allow any user to run tests on this endpoint, use the `ALL` parameter, which is the installation default:

```
ALLOW ALL
```

However, **the default "ALLOW ALL" is NOT RECOMMENDED**. Although "`ALLOW ALL`" makes it easy to install an endpoint and see that it's running, it also lets any user who can reach the endpoint potentially use that endpoint as a traffic generator. For example, End2End administrators who deploy endpoints widely in their networks probably don't expect to have those endpoints used in network stress testing.

To allow only specific users to run tests with this endpoint, remove the "`ALLOW ALL`" line and identify one or more specific Chariot or Qcheck Consoles or End2End servers by their network addresses. You can specify more than one address per protocol. For example,

```
ALLOW TCP 192.86.77.120
ALLOW TCP 192.86.77.121
ALLOW APPC netiq.johnq
```

Specify a connection-oriented protocol (that is, APPC, TCP, or SPX) as the first parameter and provide its corresponding network address as the second parameter. Endpoints only listen for incoming tests on connection-oriented protocols, like TCP. Datagram tests are set up and results are returned using their "sister" connection-oriented protocol; thus, UDP tests are set up using TCP, and IPX tests are set up using SPX.

The network address cannot be an alias or hostname; that is, in APPC it must be a fully qualified LU name, in TCP/IP it must be an IP address in dotted notation, and in IPX/SPX it must be an IPX address with hex network address and node address.

Endpoints do not respond to End2End endpoint discovery requests unless the IP address of the End2End server is specifically allowed (or unless "ALLOW ALL" is specified). This prevents the user of an End2End server from finding endpoints to which it should not have access.

You cannot use the ALLOW parameter to restrict access from one endpoint to another endpoint. The ALLOW parameter can only be used to permit (or prevent) access from specific Chariot or Qcheck Consoles or End2End servers to the endpoint at which the parameter is defined.

If, for some reason, you need to restrict your endpoint to access only your own computer, specify your own IP network address rather than 127.0.0.1. Specifying 127.0.0.1 (the equivalent of localhost) allows any other user who specifies "localhost" as Endpoint 1 to access your computer as Endpoint 2.

SECURITY_AUDITING

This keyword determines which access attempts the endpoint keeps track of in its audit file. Here are the possible parameters:

NONE	Nothing is written to the audit file.
PASSED	Only access attempts that passed the ALLOW address check are logged.
REJECTED	Only access attempts that failed the ALLOW address check are logged.
ALL	Both passed and rejected access attempts are logged.

If a test initialization fails for a reason other than address checking, no entry is made in the audit file.

AUDIT_FILENAME

This keyword specifies the filespec for the audit file. See SECURITY_AUDITING on page 35 to understand the types of events logged in its audit file. The default filename, in endpoint.ini, is endpoint.aud. If no drive or path is specified, the audit file uses the drive and path of the endpoint program.

This file contains at most two lines for each endpoint pair that is started on this endpoint. These two lines represent the start of an endpoint instance and the end of that instance.

Each line written to the audit file consists of a set of information about the endpoint instance and what it has been asked to do. The information is written in comma-delimited form, so you can load the audit file into a spreadsheet or database. When the audit file is created, an initial header line explains the contents of the subsequent entries.

The following table shows the fields of each entry in the audit file:

Time	The date and time when the entry was created, in the local time zone.
Action	Whether this entry indicates that an endpoint instance was “Started” or “Ended.”
Endpoint	Whether the endpoint is in the role of Endpoint 1 or Endpoint 2.
Protocol of Chariot Console or End2End server	The network protocol used to contact Endpoint 1.
Network Address of Chariot Console or End2End server	The network address as seen by Endpoint 1. If you encounter problems setting up your `ALLOW` entries, this is the value to use for the protocol address.
Security Result	Whether this `SECURITY_AUDITING` “passed” or was “rejected.” If this is an entry for an “Ended” action, this field is reported as “n/a.”
Endpoint Partner Protocol	The network protocol used to run the test with our partner endpoint.
Endpoint Partner Address	The network address of our partner endpoint.

ENABLE_PROTOCOL

This keyword lets you control which connection-oriented protocols this endpoint uses to listen for setup connections. This does not affect the network protocols, which can be used to run tests. Here are the possible parameters:

```
ALL
APPC
SPX
TCP
```

In general, you should use the `ALL` setting (the default). Specify protocols explicitly to reduce the overhead of listening on the other protocols or if you’re encountering errors when listening on the other protocols.

See the discussion of the ALLOW keyword on page 34 for information about support of the datagram protocols, IPX, RTP, and UDP.

SAFESTORE_DIRECTORY

Use the `SAFESTORE_DIRECTORY` keyword to change the filename of the End2End safestore files, which hold the endpoint’s schedule and any results that have not yet been sent to the End2End server. This keyword has no effect on Chariot users.

For example, the following line causes the endpoint to write its safestore files to the directory `d:\NetIQ\Endpoint`:

```
SAFESTORE_DIRECTORY D:\NETIQ\ENDPOINT
```

UPDATE_SERVER

This keyword lets you specify the Uniform Resource Locator (URL) of the Web server containing the upgraded Performance Endpoint software. This location is used when the endpoint receives a request from an End2End server to download and install upgraded endpoint software. The URL is the "scheme-specific" part of a valid Uniform Resource Locator, that is, the part after "http://" (see RFC 1738). The default for this keyword is:

```
UPDATE_SERVER ENDPOINTUPDATE.GANYMEDE.COM
```

You can also configure your endpoints to automatically download upgrades from a local server. If you don't want to perform a separate configuration of each endpoint's `endpoint.ini` file to change the location of the server, you can modify an executable file included in the endpoint packaging for all Win32 endpoints. When the endpoints are installed, they will automatically use the customized `endpoint.ini` file you edited. See "Customizing endpoint.ini for Windows Endpoints" on page 37 for more information.

END2END_SERVER

This keyword allows you to specify one or more End2End servers to which the endpoint will initiate a connection. The server is polled every 15 minutes until contacted.

These endpoints are subject to the rules governing inbound endpoints, as defined in the End2End server's Global Endpoint Defaults.

TCP is the only protocol supported for polling at this time. Here's the syntax:

```
END2END_SERVER protocol name | address
```

Here's an example of the statement using an IP address. Multiple iterations are allowed:

```
END2END_SERVER TCP 192.86.77.120
```

For more information about inbound endpoints, see the *User Guide* for End2End.

Customizing endpoint.ini for Windows Endpoints

Endpoints for Windows can be set up to automatically upgrade themselves. These endpoint computers must be able to contact a Web server where the upgraded software is stored. Although the default Web server for endpoint upgrades is accessible on the Internet at http://endpointupdate.ganymede.com/, in `endpoint.ini` you can configure the location and directory of any Web server from which the upgraded software can be accessed. See the section titled "UPDATE_SERVER" on page 37 for more information on the endpoint upgrade procedure.

We've included a utility, `gsendw32.exe` for Windows and `gsendw64.exe` for 64-bit Windows, to help you configure `endpoint.ini` for Windows endpoint upgrades. To modify `endpoint.ini` before installing an endpoint for Windows, unzip the self-extracting `gsendw32.exe` (or `gsendw64.exe`) installation file and make the changes before starting the endpoint installation.

When you click **Extract**, the endpoint files are extracted (including `endpoint.ini`) to the directory you selected.

If you are installing over an existing endpoint, the `endpoint.ini` file is not installed by default. To install the new file, use the following option for `SETUP`:

```
SETUP replace_ini
```

This option reinstalls the endpoint. If an endpoint encounters an error while processing the `endpoint.ini` file, no one is allowed to use the endpoint in tests. That means no Chariot Console, End2End server, or Qcheck program will be able to run tests with the endpoint.

Configuring Endpoints for Large-Scale Customization

To customize features such as automatic upgrades, you must edit the `endpoint.ini` file for each endpoint. For obvious reasons, you may not want to undertake such a potentially lengthy procedure. You can extract the files located in `gsendw32.exe` if you need to perform a large-scale customization of `endpoint.ini`. In addition to WinZip 7.0, you'll need the WinZip command-line support add-on and WinZip Self-Extractor. Here's how to use it:

1. Open the file `gsendw32.exe` using WinZip.
2. Extract the files to a temporary directory.
3. Edit or replace the `endpoint.ini` that is now in the temporary directory.
4. Using WinZip, create a new archive that contains all the files in the temporary directory.
5. Using the WinZip Self-Extractor, create a self-extracting executable; for the command line to run, enter the following:

   ```
   SETUP.EXE replace_ini
   ```

Now, anyone who executes the new executable you've created will automatically have the endpoint installed using the `endpoint.ini` file that you've customized.

To create a file that silently self-installs with a custom `endpoint.ini`, take the following steps:

1. Open the file `gsendw32.exe` using WinZip.
2. Extract the files to a temporary directory.
3. Edit or replace the `endpoint.ini` that is now in the temporary directory.
4. Create a custom response file (say, `customer.iss`); enter

```
i. SETUP -noinst -r -f1.\customer.iss
```

5. Using WinZip, create a new archive that contains all the files in the temporary directory.
6. Using the WinZip Self-Extractor, create a self-extracting executable; for the command line to run, enter the following:

```
SETUP.EXE replace_ini -s -f1.\CUSTOMER.ISS
```

Now, anyone who executes the file you've created will automatically have the endpoint installed using `customer.iss` as the response file, and the `endpoint.ini` file installed will also be the customized version you created.

See "UPDATE_SERVER" on page 37 for information on endpoint upgrades.

Chapter 4
Distributing Endpoints Using SMS

Endpoints can be installed and uninstalled on Windows computers automatically using Microsoft's Systems Management Server (SMS). This discussion assumes you are already familiar with package distribution via SMS.

- The SMS Server software must be installed and running properly on a Windows NT server.
- The SMS Client software must be installed and running properly on the Windows computers (that is, Windows 3.1x, plus all Win32 operating systems) where you want to remotely install endpoints. A folder titled "SMS Client" is present when the software has been installed correctly.

Our testing indicates that Version 1.2 of SMS (with Service Pack 2) or later is required.

Installing Endpoints Using SMS

Follow these steps to install endpoints with SMS version 1.2.

1. If you are installing endpoints on Windows, you need to unzip the `gsendw32.exe` file from the CD.
2. Once the files are extracted and saved to the directory you selected, create a response file for each distinct set of client computers.

You need to create a response file (typically named `setup.iss`) for each unique installation. Each different operating system or target path is a unique installation. For example, you may have a set of Windows NT x86 computers where you want to install the endpoint in a directory named for our software (that is, `d:\Program Files\NetIQ\Endpoint`) and another set where you want to install to a directory named `c:\Programs\Endpoint`. In this case, you would create two separate response files, one for each distinct set of installations.

To create a response file for a set of computers, go to one of the computers in the set and change the current working directory to the one where you extracted and saved the installation files for that computer. Enter a command like the following:

```
setup -noinst -r -f1d:\yourdirectory\setup.iss
```

It is important to run `SETUP` from that directory, because the version of `setup.exe` in your Windows directory will not work.

Here are the parameters for the `SETUP` command:

`-noinst`	No install: create the `setup.iss` file, but don't really install the endpoint right now. This is a NetIQ-specific option and must appear before any setup-defined options, like "`-r`."
`-r`	Records the installation actions in an `.iss` file.
`-f1`	Gives the path name for the output response file.

1. Copy the endpoint installation files from the directory to a hard disk, along with the `setup.iss` file.
2. For each distinct set of client computers, create a directory on a hard disk available to the SMS Server. Into each directory, copy the corresponding endpoint installation files. In addition, copy the new `setup.iss` file you just created to the matching directory.

   For example, create directories on the SMS Server's hard disk named `\Endpoint_WNT1` and `\Endpoint_WNT2` for the two sets of client computers discussed in the preceding step. Copy all the unzipped installation files to each of these directories. Finally, copy the `setup.iss` file for the first set of client computers into directory `\Endpoint_WNT1`; copy the other `setup.iss` file into the second directory.
3. Inside the SMS program at the SMS Server, select **File**, then **New**. Click **Import**. Navigate to the drive and path where you've copied the endpoint installation files and their `setup.iss` file. Choose the corresponding `.pdf` file, which should be shown in the file list.

   A dialog box should appear showing the correct package installation information.
4. Click **Workstations**. In the dialog box that follows, move to the same drive and path you selected in step 3 by clicking the **"..."** symbol under "Source Directory." Then choose "Automated Installation" and click **Properties**. You should see the command line string necessary to install the endpoint, similar to the string you entered to create the `setup.iss` file.
5. Click **OK**, **Close**, and then **OK** to finish creating the SMS package. Repeat these steps for each distinct set of client computers.
6. Configure the packages at the SMS Server for your schedules and sites.
7. Decide when you want the endpoints installed, and on which computers. Configure these schedules and sites in SMS as you would with other SMS packages. See the SMS documentation for assistance.

Our software supports SMS Inventory Information, which has been encoded in the `.pdf` files.

Uninstalling Endpoints Using SMS

Follow these steps to remove endpoint packages, using SMS version 1.2:

1. At the SMS Server, select a package to delete and update the name of the `Delsl?.isu` file.
2. Inside the SMS program at the SMS Server, select **File**, then **Open** the endpoint package you want to uninstall.
3. Click **Workstations**. In the dialog box that follows, move to the drive and path for the package by clicking the "..." symbol under "Source Directory." Then choose **Automated Uninstallation** and click **Properties**. It should show the command line string necessary to uninstall the endpoint, similar to the string you entered to create the `setup.iss` file. You should see a sequence that looks like " `fDelsl?.isu`" in the middle of the string. The "?" here is a number, representing the latest installation on the client computer. For example, if the endpoint has been installed twice, the client computer will have a file named "`Delsl2.isu`" in the directory where you installed the endpoint. This filename at the SMS Server must exactly match the filename at the SMS Client where the endpoint is being uninstalled.
4. Click **OK**, **Close**, and then **OK** to finish the update of the SMS package. Repeat these steps for each distinct set of client computers.
5. Configure the packages at the SMS Server for your schedules and sites.
6. Decide when you want the endpoints uninstalled, and on which computers. Configure these schedules and sites in SMS as you would with other SMS packages. See the SMS documentation for assistance.

Index

L

M

P

R

S

T

U

V

W