



IxChariot® Performance Endpoints



Release 6.50

913-0013-05 Rev. A

May 2007





Copyright © 2007 Ixia. All rights reserved.

This publication may not be copied, in whole or in part, without Ixia's consent.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 and FAR 52.227-19.

Ixia, the Ixia logo, and all Ixia brand names and product names in this document are either trademarks or registered trademarks of Ixia in the United States and/or other countries. All other trademarks belong to their respective owners.

The information herein is furnished for informational use only, is subject to change by Ixia without notice, and should not be construed as a commitment by Ixia. Ixia assumes no responsibility or liability for any errors or inaccuracies contained in this publication.

Corporate Headquarters	Ixia Worldwide Headquarters 26601 W. Agoura Rd. Calabasas, CA 91302 USA +1 877 FOR IXIA (877 367 4942) +1 818 871 1800 (International) (FAX) +1 818 871 1805 sales@ixiacom.com	Web site: www.ixiacom.com General: info@ixiacom.com Investor Relations: ir@ixiacom.com Training: training@ixiacom.com Support: support@ixiacom.com +1 877 367 4942
EMEA	Ixia Europe Limited Globeside Business Park Building One, Unit A Marlow, SL7 1GJ United Kingdom +44 1869 356370 (FAX) +44 1869 356371 ixiaeurope@ixiacom.com	Support: eurosupport@ixiacom.com +44 1869 356370 (Option 5)
Asia Pacific	Asia Pacific Representative Office New Shanghai International Tower, Suite 26E 360 Pudong Nan Rd Shanghai 200120 China +86 21 50543439 ixiachina@ixiacom.com	Support: support@ixiacom.com +1 818 871 1800 (Option 1)
Japan	Ixia KK Aioi Sampo Shinjuku Building, 16th Floor 3-25-3 Yoyogi Shibuya-Ku Tokyo 151-0053 Japan +81 3 5365 4690 (FAX) +81 3 3299 6263 ixiajapan@ixiacom.com	Support: support@ixiacom.com +1 818 871 1800 (Option 1)
India	Ixia India No. 508, 6th Main 6th Cross ST Bed, Koramangala 4th Block Bangalore 560 034 India +91 80 25633570 (FAX) +91 80 25633487 ixiaindia@ixiacom.com	Support: support-india@ixiacom.com +91 80 32918500

913-0013-05 Rev. A
May 25, 2007

Table of Contents

Chapter 1 IxChariot Performance Endpoints Overview

What Is a Performance Endpoint?	1-1
Endpoint Versions	1-1
Displaying Endpoint Configuration Settings	1-2
What's New in Version 6.50?	1-2

Chapter 2 Performance Endpoint Specifications

Operating System and Protocol Stack Support	2-2
Performance Endpoint Support for IxChariot Functions . . .	2-3
IPv6 Test Module Support	2-4
MSS Option Support	2-4
Performance Endpoint Support for QoS	2-4
Endpoint Computer Resource Guidelines	2-5
Generating Maximum Throughput	2-6
Calculating Memory Requirements	2-6
Endpoint Pair Capacity	2-7

Endpoint Capabilities	2-10
---------------------------------	------

Chapter 3 Endpoint Initialization File

Initialization File Description	3-1
endpoint.ini Example	3-2
Keyword Default Values	3-2
Keywords	3-3
ALLOW	3-3
SECURITY_AUDITING	3-4
AUDIT_FILENAME	3-5
ENABLE_PROTOCOL	3-6
USE_ENCRYPTION	3-6
SOCKET_SEND_BUFFER_SIZE	3-7
SOCKET_RECEIVE_BUFFER_SIZE	3-8
FORCE_CLOCKSYNC	3-8
MAX_PAYLOAD_DISK_USAGE	3-9
MAX_PAYLOAD_MEMORY_USAGE	3-9
PAYLOAD_MEMORY_LIMIT_USAGE	3-9
MANAGEMENT_PORT	3-10
Configuring Endpoints for Large-Scale Customization	3-10

Chapter 4 Ixia Load Module

Installing the Ixia Performance Endpoint	4-1
Updating the Ixia Performance Endpoint	4-1
About Stack Manager	4-1
Logging and Messages	4-2
Message CHR0181	4-2
Starting and Stopping Ixia Endpoints	4-2
Stopping the Endpoint	4-3
Restarting the Endpoint	4-3

Supported Load Modules	4-4
------------------------------	-----

Chapter 5 HP-UX

Installation Requirements for HP-UX Endpoints	5-1
Endpoint Installation for HP-UX.	5-2
Performance Endpoint File Name	5-2
Installation Procedures	5-2
Unattended Installation for HP-UX	5-5
What Happens During Installation	5-5
Removing the Endpoint Package (Uninstall)	5-6
Configuring HP-UX Endpoints	5-6
Configuration for TCP/IP	5-6
Determining Your IP Network Address	5-7
Testing the TCP/IP Connection	5-7
Sockets Port Number	5-7
Running HP-UX Endpoints	5-8
Starting an HP-UX Endpoint	5-8
Stopping an HP-UX Endpoint	5-8
Cleanup after Unexpected Errors	5-9
How to Tell If an HP-UX Endpoint Is Active	5-9
Disabling Automatic Startup	5-9
Messages CHR0174, CHR0204, CHR0210, or CHR0245	5-9
Logging and Messages	5-9
CORE and CMA_DUMP.LOG Files	5-10
Message CHR0181	5-10
Updates for HP-UX	5-10

Chapter 6 IBM AIX

Installation Requirements for AIX Endpoints	6-1
---------------------------------------------------	-----

Endpoint Installation for AIX.	6-2
Performance Endpoint File Name.	6-2
Installation Procedures	6-2
Unattended Installation for AIX	6-4
What Happens During Installation	6-4
Removing the Endpoint Package (Uninstall).	6-5
Configuring AIX Endpoints.	6-6
Configuration for TCP/IP.	6-6
Determining Your IP Network Address	6-6
Testing the TCP/IP Connection.	6-6
Sockets Port Number	6-7
Maximum Value for the MSS Option.	6-7
Running AIX Endpoints	6-7
Starting an AIX Endpoint.	6-7
Stopping an AIX Endpoint.	6-8
Cleanup after Unexpected Errors	6-8
How to Tell If an AIX Endpoint Is Active	6-8
Disabling Automatic Startup	6-9
Logging and Messages	6-10
Message CHR0181.	6-10
Updates for AIX	6-10

Chapter 7 Linux 32-Bit (x86)

Linux 32-bit x86 Performance Endpoint File Names	7-1
Protocols Supported	7-2
Installation Requirements	7-2
Installing 32-bit Linux x86 Endpoints	7-3
TAR-Based Endpoint Installation for 32-bit Linux	7-3
RPM-Based Endpoint Installation for 32-bit Linux.	7-5

What Happens During Installation	7-7
Removing 32-bit Linux x86 Endpoints	7-8
Removing the TAR-Based Endpoint Package (Uninstall)	7-8
Removing the RPM Endpoint Package (Uninstall)	7-8
Configuring 32-bit Linux x86 Performance Endpoints	7-9
Configuration for TCP/IP	7-9
Determining Your IP Network Address for TAR and RPM Linux . . .	7-9
Sockets Port Number	7-9
Testing the TCP Connection	7-10
Autostarting the Endpoint	7-10
Running Linux Endpoints	7-11
Starting a Linux Endpoint.	7-11
Stopping a Linux Endpoint.	7-12
Cleanup after Unexpected Errors	7-12
How to Tell If a Linux Endpoint Is Active	7-12
Disabling Automatic Startup	7-12
Logging and Messages	7-13
Message CHR0181	7-13
Increasing the Number of Concurrent Connections	7-13

Chapter 8 Linux 64-Bit (x86-64)

Supported Distributions	8-1
Installation Requirements	8-1
Installing Linux 64-Bit Performance Endpoints	8-2
TAR-Based Installation of the x86-64 Linux Endpoint	8-2
RPM-Based Installation for the x86-64 Linux Endpoint	8-4
What We Do During Installation.	8-5
Removing 64-Bit Linux Endpoints	8-6
Removing the TAR-Based Endpoint Package (Uninstall)	8-6
Removing the RPM-Based Endpoint Package (Uninstall).	8-7

Configuring 64-Bit Linux Endpoints	8-7
Configuration for TCP/IP	8-7
Determining Your IP Network Address	8-7
Sockets Port Number	8-8
Testing the TCP Connection	8-8
Running 64-Bit Linux Endpoints.	8-8
Autostarting the Endpoint	8-8
Starting a 64-Bit Linux Endpoint	8-9
Stopping a 64-Bit Linux Endpoint	8-9
Cleanup after Unexpected Errors	8-10
How to Tell If a 64-Bit Linux Endpoint Is Active.	8-10
Disabling Automatic Startup	8-10
Increasing the Number of Concurrent Connections	8-10
Logging and Messages	8-10
Message CHR0181.	8-11

Chapter 9 Linux on ARM Processors

Linux on ARM Performance Endpoints	9-1
About Endianness.	9-2
Installing the Linux 32-bit on ARM Endpoint	9-2
Requirements	9-2
No Log Files are Created	9-2
TAR-Based Installation for Linux 32-Bit on ARM Endpoints	9-2
What We Do During Installation	9-3
Uninstalling	9-3
Configuring the Linux 32-Bit on ARM Endpoint	9-3
Supported Protocols	9-3
Configuration for TCP/IP.	9-4
Determining Your IP Network Address	9-4
Sockets Port Number	9-4
Testing the TCP Connection	9-4
Running the Linux 32-Bit on ARM Endpoint	9-5

Starting the Linux 32-Bit on ARM Endpoint	9-5
Stopping the Linux 32-Bit on ARM Endpoint	9-5
Clean-up After Unexpected Errors	9-6
How to Tell if the Linux 32-Bit on ARM Endpoint is Active	9-6

Chapter 10 Linux on Lexra

Installing the Linux 32-bit on Lexra Endpoint	10-1
Requirements	10-1
No log files are created	10-1
TAR-Based Installation for Linux 32-bit on Lexra Endpoints	10-2
What We Do During Installation	10-2
Configuring the Linux 32-bit on Lexra Endpoint	10-2
Configuration for TCP/IP	10-3
Determining Your IP Network Address	10-3
Sockets Port Number	10-3
Testing the TCP Connection	10-3
Running the Linux 32-bit on Lexra Endpoint	10-4
Starting the Linux 32-bit on Lexra Endpoint	10-4
Stopping the Linux 32-bit on Lexra Endpoint	10-4
Cleanup after Unexpected Errors	10-5
How to Tell If the Linux 32-bit on Lexra Endpoint Is Active	10-5

Chapter 11 Linux on OpenWrt (MIPS Platforms)

Performance Endpoint Overview	11-1
File Names	11-1
Linux kernel 2.4.30	11-1
Little Endian	11-2
Installing the Linux 32-bit on OpenWrt Endpoint	11-2
Requirements	11-2
No Log Files are Created	11-2
Linux 32-Bit on OpenWrt Endpoints Installation Instructions	11-2
What We Do During Installation	11-2

Removing the Linux 32-bit on OpenWrt Endpoint	11-3
Removing TAR-Based Installations	11-3
Removing IPKG-Based Installations	11-3
TCP/IP Sockets Interface Support	11-3
Determining Your IP Network Address	11-3
Supported Protocols	11-4
Sockets Port Number	11-4
Running the Linux 32-bit on OpenWrt Endpoint	11-4
Starting the Linux 32-bit on OpenWrt Endpoint	11-4
Stopping the Linux 32-bit on OpenWrt Endpoint	11-5
Clean-up After Unexpected Errors	11-5
How to Tell if the Linux 32-bit on OpenWrt Endpoint is Active	11-5

Chapter 12 Linux on PowerPC

Installing the Linux on PowerPC Performance Endpoint . .	12-1
Requirements	12-1
No Log Files are Created	12-1
TAR-Based Installation for Linux on PowerPC Endpoints	12-2
Contents of the tar file	12-2
Uninstalling	12-3
Configuring the Linux on PowerPC Endpoint	12-3
Supported Protocols	12-3
Determining Your IP Network Address	12-3
Sockets Port Number	12-3
Testing the TCP Connection	12-3
Running the Linux on PowerPC Endpoint	12-4
Starting the Linux on PowerPC Endpoint	12-4
Stopping the Linux on PowerPC Endpoint	12-4
Cleanup After Unexpected Errors	12-5
How to Tell if the Linux on PowerPC Endpoint is Active	12-5

Chapter 13 Mac OS X

Platforms Supported	13-1
Installing the Mac OS Performance Endpoint	13-1
Installation Procedure	13-2
Removing the Endpoint (Uninstall)	13-2
What Happens During Installation	13-2
Downgrading to an older version of the Endpoint	13-2
Configuring Mac OS X Endpoints	13-3
Configuration for TCP/IP	13-3
Determining Your IP Network Address	13-3
Testing the TCP Connection	13-3
Sockets Port Number	13-4
Running Mac OS X Endpoints	13-4
Stopping a Mac OS X Endpoint	13-5
Cleanup after Unexpected Errors	13-5
How to Tell If a Mac OS X Endpoint Is Active	13-5
Disabling Automatic Startup	13-5
Logging and Messages	13-5
Updates for Mac OS X	13-6

Chapter 14 Microsoft Windows 32-Bit

Windows Operating Systems Supported	14-1
Installation Requirements for the 32-bit Windows Endpoint	14-2
Installing the Endpoint	14-3
Performance Endpoint Filenames	14-4
User and System Permission Requirements	14-4
Before Installing an Older Endpoint	14-4
Installing from CD-ROM	14-4

Installing from a Downloaded Executable	14-6
Silent Mode Installation	14-8
Installing the Windows Endpoint with SMS	14-8
What Happens During Installation	14-8
Uninstalling the Endpoint	14-9
Removing the Endpoint Package (Uninstall)	14-9
Silent Mode Uninstall	14-9
Removing the Endpoint Manually	14-9
Configuring Windows Endpoints	14-10
Sockets Port Number	14-10
Windows Configuration for IPX and SPX	14-10
Windows Configuration for TCP/IP	14-11
Running Windows Endpoints	14-13
Starting the Endpoint	14-13
Stopping a Windows Endpoint	14-13
Disable Your Screen Saver	14-13
The SetAddr Utility	14-13
Disabling Automatic Startup	14-15
How to Tell If a Windows Endpoint Is Active	14-16
Logging and Messages	14-16
Getting the Latest Fixes and Service Updates	14-16
Updates and Information for Windows	14-16
Updates for Microsoft SNA Server	14-16

Chapter 15 Microsoft Windows CE

Available Performance Endpoints for Windows CE	15-1
Installation Requirements	15-2
Network Protocol Stacks	15-2
Endpoint Installation for Windows CE	15-3

Installing the <i>pewcearm</i> Performance Endpoint	15-3
Installing the <i>pewcearm_cl</i> Performance Endpoint	15-3
Installing the <i>pewcearm_disk</i> Performance Endpoint	15-3
Installing the <i>pewcex86</i> Performance Endpoint	15-4
Removing the Endpoint Package (Uninstall)	15-5
Windows CE Configuration for TCP/IP	15-5
Determining Your IP Network Address	15-5
Testing the TCP Connection	15-5
Sockets Port Number	15-6
Running Windows CE Endpoints	15-6
Intel Strong Arm and XScale Processor Based Operation	15-6
Intel x86 Processor Based Operation	15-7
Checking the Endpoint Version	15-7
Logging and Messages	15-8
Limitations of the Windows CE Endpoint	15-8

Chapter 16 Microsoft Windows 64-Bit

Operating Systems and Processors Supported	16-1
Installation Requirements for the Windows 64-Bit Endpoint	16-2
Supported Protocols	16-2
Microsoft Windows 64-Bit Performance Endpoint Installation	16-3
Performance Endpoint Filenames	16-3
User and System Permission Requirements	16-3
Interactive Installation	16-3
Silent Mode Installation	16-5
What We Do During Installation	16-5

Uninstalling the Endpoint	16-5
Removing the Endpoint Package (Uninstall).	16-5
Silent Mode Uninstall	16-6
Removing the Endpoint Manually	16-6
Configuring Windows 64-bit Performance Endpoints	16-6
64-bit Windows Configuration for TCP/IP	16-6
Determining Your IP Network Address	16-7
Trying Out the TCP/IP Connection	16-7
Sockets Port Number	16-8
Running Microsoft Windows 64-Bit Performance	
Endpoints	16-8
Starting a Windows 64-bit Performance Endpoint	16-8
Stopping a Windows 64-bit Performance Endpoint.	16-9
Disable Your Screen Saver	16-9
Disable NIC Power Save Mode	16-9
Disabling Automatic Startup	16-9
How to Tell If a Windows 64-bit Performance Endpoint Is Active . .	16-9
The SetAddr Utility for 64-bit Windows	16-9
Logging and Messages	16-12
Getting the Latest Fixes and Service Updates.	16-12

Chapter 17 Sun Solaris

Platforms Supported	17-1
Installation Requirements for Solaris Endpoints	17-1
Endpoint Installation for Sun Solaris	17-2
Performance Endpoint File Name.	17-2
Installation Procedures	17-2
Installation Defaults File for Solaris.	17-5
Unattended Installation for Solaris	17-5
What Happens During Installation	17-6

Removing the Endpoint Package (Uninstall)	17-7
Configuring Solaris Endpoints	17-7
Configuration for TCP/IP	17-7
Determining Your IP Network Address	17-7
Testing the TCP/IP Connection	17-8
Sockets Port Number	17-8
Running Solaris Endpoints	17-8
Starting a Solaris Endpoint	17-9
Stopping a Solaris Endpoint	17-9
Cleanup after Unexpected Errors	17-9
How to Tell If a Solaris Endpoint Is Active	17-10
Disabling Automatic Startup	17-10
Logging and Messages	17-10
Known Problems	17-10
Message CHR0181	17-11
Updates for Sun Solaris	17-11

Chapter 18 Web-Based Performance Endpoint

Web-Based Performance Endpoint Overview	18-1
Running the Web-Based Endpoint	18-2
Starting the Endpoint	18-3
Restarting the Endpoint	18-3
Error Handling	18-3
Compatibility with Other Endpoints	18-3
Stopping the Web-Based Endpoint	18-4

Appendix A Archived Endpoint Specifications

Operating System and Protocol Stack Support A-1

Performance Endpoint Support for IxChariot Functions . . . A-3

Endpoint Computer Resource Guidelines A-4

 Calculating Memory Requirements A-4

 Endpoint Pair Capacity A-5

1

IxChariot Performance Endpoints Overview

This guide contains information about the IxChariot 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 in this guide. In addition to topics discussing issues common to all the endpoints, this guide also contains information about each endpoint, organized in separate chapters.

This chapter includes the following topics:

- [What Is a Performance Endpoint?](#) on page 1-1
- [Endpoint Versions](#) on page 1-1
- [Displaying Endpoint Configuration Settings](#) on page 1-2
- [What's New in Version 6.50?](#) on page 1-2

What Is a Performance Endpoint?

IxChariot executes tests using *endpoint* computers. Each computer used as an endpoint requires *Performance Endpoint* software. These programs operate in the background, carrying out the instructions provided by IxChariot test scripts. Endpoints collect performance statistics while executing test scripts and send the statistics to the IxChariot Console, which produces reports reflecting the response time, transaction rate, connectivity, and throughput in your system under test. (Refer to the *IxChariot Getting Started Guide* for an overview of IxChariot operations.)

Once installed, performance endpoints rarely require any interaction with users.

Endpoint Versions

With each new release of IxChariot, the endpoints are updated to support new functionality. However, because some endpoint operating systems are rarely used

or provide limited support for IxChariot features, endpoints for a few operating systems have been archived. These endpoints are still made available on the Performance Endpoints CD-ROM and on the Ixia Web site; however, they may not support the latest capabilities of IxChariot. 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.

Refer to Appendix A, *Archived Endpoint Specifications* for a description of the archived endpoints.

Displaying Endpoint Configuration Settings

If you right-click a pair in the IxChariot Test window and select “Show endpoint configuration...” from the drop-down menu, IxChariot displays an Endpoint Configuration window for that pair. This window displays all of the configuration settings for each of the endpoints in the selected pair.

What’s New in Version 6.50?

Following is a list of the new, enhanced, or archived endpoints for release 6.50:

- Windows CE 6.0 QoS Support:

IxChariot and the Windows CE Performance Endpoint now provide support for Windows CE 6.0 QoS. Refer to *Performance Endpoint Support for QoS* on page 2-4 for a summary of the QoS support provided to all the Performance Endpoints.

- Configurable Management Port:

In releases prior to 6.50, the TCP management port was not configurable: it was fixed at 10115. Starting with release 6.50, you can specify the management port number. To configure the management port, you enter a parameter in the .ini file for each affected endpoint, and you set the value in the IxChariot Console. Refer to *Keywords* on page 3-3 for detailed information.

- Force clock synchronization keyword:

Starting in release 6.50, you can configure your endpoint.ini files to force the endpoints to synchronize their clocks before each test run. Refer to *Keywords* on page 3-3 for detailed information.

- Sockets buffer size keywords:

Starting in release 6.50, you can configure your endpoint.ini files to specify how much send buffer space and receive buffer space the operating system should allocate for the sockets datagram service. (These keywords are applicable to video, multicast video, and IPTV tests only.) Refer to *Keywords* on page 3-3 for detailed information.

2

Performance Endpoint Specifications

This chapter describes the resource requirements and the supported functions of the IxChariot Performance Endpoints. It contains the following topics:

- *Operating System and Protocol Stack Support* on page 2-2
- *Performance Endpoint Support for IxChariot Functions* on page 2-3
- *Endpoint Computer Resource Guidelines* on page 2-5
- *Endpoint Capabilities* on page 2-10

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 <http://www.ixiacom.com/support/ixchariot>.

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

Operating System and Protocol Stack Support

Table 2-1 identifies the supported operating systems and protocol stack software for each currently-active endpoint. The table lists 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.

Table 2-1.Active Endpoints - Operating System Compatibility

Endpoint	OS version	TCP, UDP, RTP	IP Multicast version	IPX/SPX stack
Apple Macintosh (G4 and G5 processors)	OS X 10.3	included	included	no
HP-UX	HP-UX v11.0	included	v11.0	no
IBM AIX	AIX v4.1.4	included	v4.1.4	no
Ixia Load Module	Linux - automatically downloaded	included	included	no
Linux 32-bit (x86)	kernel 2.4.20	included	kernel 2.0.32	no
Linux 64-bit (x86-64)	kernel 2.4.20	included	kernel 2.4.0test7-42	no
Linux on ARM	kernel 2.4.20	included	kernel 2.4.20	no
Linux on Lexra	kernel 2.4.18	TCP and UDP only	kernel 2.4.18	no
Linux on OpenWrt	kernel 2.4.30	included	kernel 2.4.30	no
Linux on PowerPC	kernel 2.4.19	included	kernel 2.4.19	no
Microsoft Windows NT	Windows NT, SP4	included	SP3 (IGMPv1) SP4 (IGMPv2)	included
Microsoft Windows 2000	Windows 2000	included	included	included
Microsoft Windows CE	Windows CE 4.2, 4.3, 5.0	included	included	no
Microsoft Windows Vista (32-bit and 64-bit editions)	Windows Vista	included	included	no
Microsoft Windows XP	Windows XP (32-bit)	included	included	included
Microsoft Windows XP 64-bit Edition	Windows XP x64 Edition	included	included	no
Microsoft Windows Server 2003 (32-bit edition)	Windows Server 2003	included	included	included
Microsoft Windows Server 2003 (64-bit edition)	Windows Server 2003	included	included	no

Table 2-1.Active Endpoints - Operating System Compatibility (Continued)

Endpoint	OS version	TCP, UDP, RTP	IP Multicast version	IPX/SPX stack
Sun Solaris for SPARC	Solaris v2.4	included	v2.4	no
Sun Solaris for x86	Solaris v2.4	included	v2.4	no

Performance Endpoint Support for IxChariot Functions

The following table describes the basic Performance Endpoint capabilities for the supported operating systems. (Refer to [Performance Endpoint Support for QoS](#) on page 2-4 for information about QoS capabilities.)

Table 2-2.Performance Endpoint Capabilities per OS

Endpoint OS	Trace-route	CPU Utiliti-zation	VoIP Tests	Video Pair Tests	IPVT Tests	IPv6 Tests	802.11 Statistics
Apple Macintosh OS X (G4 and G5 processors)	No	Yes	Yes	Yes	No	No	No
HP-UX	Yes	Yes	No	Yes	No	No	No
IBM AIX	Yes	Yes	No	Yes	No	No	No
Ixia Load Module	Yes ^a	Yes	Yes	Yes	Yes	Yes	No
Linux 32-bit (x86)	Yes	Yes	Yes	Yes	Yes	Yes ^b	No
Linux 64-Bit x86-64)	Yes	Yes	Yes	Yes	Yes	Yes ^b	No
Linux on ARM	Yes	Yes	Yes	Yes	Yes	No	No
Linux on Lexra	Yes	Yes	Yes	Yes	Yes	No	No
Linux on OpenWrt	Yes	Yes	Yes	Yes	Yes	No	No
Linux on PowerPC	Yes	Yes	Yes	Yes	No	No	No
Microsoft Windows NT 4	Yes	Yes	Yes	Yes	No	No	No
Microsoft Windows 2000	Yes	Yes	Yes	Yes	No	Yes ^b	Yes
Microsoft Windows Server 2003	Yes	Yes	Yes	Yes	No	Yes	Yes
Microsoft Windows XP	Yes	Yes	Yes	Yes	No	Yes ^b	Yes
Microsoft Windows XP 64-bit Edition	Yes	No	Yes	Yes	No	Yes	No
Microsoft Windows Vista	Yes	Yes	Yes	Yes	No	Yes ^b	Yes
Microsoft Windows CE	No	No ^c	Yes	Yes	No	No	Yes ^d

Table 2-2. Performance Endpoint Capabilities per OS (Continued)

Endpoint OS	Trace-route	CPU Utilization	VoIP Tests	Video Pair Tests	IPVT Tests	IPv6 Tests	802.11 Statistics
Sun Solaris for SPARC	Yes	Yes	Yes	Yes	No	No	No
Sun Solaris for x86	Yes	Yes	Yes	Yes	No	No	No

a. Traceroute is support unless you use the endpoint address as the management address.

b. See “[IPv6 Test Module Support](#)” on page 2-4.

c. Support for CPU Utilization on Windows CE is device-dependent. For more information, see <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wcemain4/html/cerefGetIdleTime.asp>.

d. Windows CE version 4.20 or later.

IPv6 Test Module Support

Currently, testing with version 6 of the Internet Protocol (IPv6) is supported on endpoints for Ixia Performance Endpoints, Windows 2003, Windows Vista, Windows XP (32-bit and 64-bit), the Linux x86 endpoint, and the Linux 64-Bit endpoint. You may need to configure IPv6 support on these operating systems before you begin testing. Refer to “IPv6 Configuration and Testing” in the IxChariot User Guide for detailed information.

Linux kernel 2.4.20 is required for IPv6 support.

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,” which you can download from the Microsoft web site.

MSS Option Support

The Maximum Segment Size (MSS) is defined as the maximum number of bytes in the TCP payload of an IP packet. The following Ixia Performance Endpoints support the use of the MSS Option in testing:

- IBM AIX
- Linux 32-bit (x86)
- Linux 64-Bit (x86-64)
- Linux on Lexra
- Linux on OpenWrt
- Linux on ARM
- Linux on PowerPC

Refer to “Setting the Transmit MSS Option” in the *IxChariot Scripts Development and Editing Guide* for additional information.

Performance Endpoint Support for QoS

The following table indicates the QoS support that the Performance Endpoints provide for the supported operating systems.

Table 2-3. Performance Endpoint QoS Support

Operating System	IP TOS	DiffServ	GQoS	qWave	WMM
Apple Macintosh	Yes	No	No	No	No
IoT Load Module	Yes	Yes	No	No	No
Linux (all)	Yes	Yes ^a	No	No	No
UNIX (all)	Yes	Yes	No	No	No
Microsoft Windows NT	Yes	No	No	No	No
Microsoft Windows 2000	Yes	Yes	Yes	No	No
Microsoft Windows Server 2003 (32-bit and 64-bit editions)	Yes	Yes	Yes	No	No
Microsoft Windows XP (32-bit and 64-bit editions)	Yes	Yes	Yes	No	No
Microsoft Windows Vista (32-bit and 64-bit editions)	No	No	Yes	Yes	No
Microsoft Windows CE 5.0 (and earlier)	No	No	No	No	No
Microsoft Windows CE 6.0 (and later)	No	No	No	No	Yes
Microsoft Mobile 5.0 AKU3	No	No	No	No	Yes

a.integrated into 2.4 kernel.

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 IxChariot. The type of information you need depends upon your usage. The following topics provide recommended endpoint computer specifications according to different testing scenarios.

This section includes the following topics:

- [Generating Maximum Throughput](#) on page 2-6
- [Calculating Memory Requirements](#) on page 2-6
- [Endpoint Pair Capacity](#) on page 2-7

- [IxOS Endpoint Pair Capacities](#) on page 2-8

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.

Table 2-4. Guidelines for Selecting CPUs

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	Pentium III or greater, with the latest NICs (consider multiprocessors)

Windows 2000/2003, Windows XP, and Linux yield the highest throughput. If you test on one of the Windows OSs with the IxChariot 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.

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 ("n/a" indicates that the protocol is not supported by the endpoint).

Table 2-5. Calculating Memory Requirements

Operating System	Base RAM (in KB)	TCP KB/pair	UDP KB/pair	RTP KB/pair	SPX KB/pair	IPX KB/pair
Apple Macintosh OS_X (G4 and G5 processors)	2540	142 - 276	177-312	158-308	n/a	n/a
HP-UX	844	140-150	257-292	158-207	n/a	n/a

Table 2-5. Calculating Memory Requirements (Continued)

UNIX (AIX)	1176	132-284	146-296	146-296	n/a	n/a
Ixia Load Module	1320	57-74	89-105	65-84	n/a	n/a
Ixia Load Module, with Scalable Mode enabled ^a	1320	22-27	n/a	n/a	n/a	n/a
Linux 32-bit (x86)	1100	140-240	170-280	160-280	n/a	n/a
Linux 64-Bit (x86-64)	1260	150-260	200-300	150-260	n/a	n/a
Linux on ARM	308	55-67	92-108	67-85	n/a	n/a
Linux on Lexra	744	63-140	99-177	81-163	n/a	n/a
Linux on OpenWrt	316	76-96	56-64	68-84	n/a	n/a
Linux on PowerPC	364	56-68	80-110	56-62	n/a	n/a
Sun Solaris for SPARC	2200	58-85	103-128	110-150	n/a	n/a
Sun Solaris (x86)	4500	62-668	202-616	164-1028	n/a	n/a
Windows NT	2076	35-60	160-180	160-180	35-60	160-180
Windows CE	277	44-70	196-436	700-773	n/a	n/a
Windows 2000	3220	200-430	240-400	200-330	35-60	160-180
Windows XP	2800	205-350	240-420	193-320	35-60	160-180
Windows XP 64-bit Edition	3800	219-388	272-438	226-374	n/a	n/a
Windows Vista (32-bit)	3500	203-356	240-396	205-340	n/a	n/a
Windows Vista (64-bit)	3500	203-356	240-396	205-340	n/a	n/a
Windows Server 2003 32-bit Edition	3560	48-232	116	124	32-244	44-256
Windows Server 2003 64-bit Edition	4748	64-344	148	144	n/a	n/a

a. Refer to [IxOS Endpoint Pair Capacities](#) on page 2-8 for more information about scalable mode.

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 *IxChariot Application Scripts* guide for more information on script variables.

Endpoint Pair Capacity

The following table identifies the maximum number of pairs supported by each of the operating systems for which active Performance Endpoints are available. These pairs ran on a 10 Mbps Ethernet LAN. The values in the pairs columns represent the maximum 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.

Table 2-6. Endpoint Pair Capacity

Operating System	Installed RAM	TCP pairs	RTP or UDP pairs	SPX pairs	IPX pairs
Apple Macintosh OS X	512 MB	200	100	n/a	n/a
HP-UX	1 GB	200	150	n/a	n/a
IBM AIX 4.1	1 GB	200	180	n/a	n/a
IxOS	Refer to IxOS Endpoint Pair Capacities on page 2-8.				
Linux 32-bit (x86)	768 MB	300	180	n/a	n/a
Linux 64-Bit (x86-64)	768 MB	300	180	n/a	n/a
Linux on ARM	16 MB	15	15	n/a	n/a
Linux on Lexra	16 MB	20	5	n/a	n/a
Linux on OpenWrt	16 MB	106	55	n/a	n/a
Linux on PowerPC	32 MB	120	100	n/a	n/a
Sun Solaris for SPARC	512 MB	100	80	n/a	n/a
Sun Solaris for x86	768 MB	500	200	n/a	n/a
Windows CE	56 MB	85	35	n/a	n/a
Windows NT/2000/XP	32 MB	500	100	300	100
Windows Vista	1 GB	3000	1000	n/a	n/a
Windows XP x64 Edition	768 MB	175	120	n/a	n/a

IxOS Endpoint Pair Capacities

IxChariot provides a *scalable mode* for TCP tests that use Ixia ports. (Refer to “Large-Scale Tests in IxChariot” in the *IxChariot User Guide* for more information about scalable mode testing.) The following tables identify the maximum number of pairs supported by the IxOS Performance Endpoint for various load modules, for both non-scalable mode and scalable mode.

[Table 2-7](#) lists the maximum number of pairs supported by the IxOS Performance Endpoint running in non-scalable mode.

Table 2-7. Ixia Load Module Pair Capacity - Non-Scalable Mode

Ixia Load Module	Installed RAM	TCP Pairs	RTP or UDP Pairs	IPX / SPX Pairs
ALM1000T8	512 MB	500	500	n/a
TXS family ^a	256 MB	500	500	n/a
LM100TXS2	128 MB	150	150	n/a

Table 2-7. Ixia Load Module Pair Capacity - Non-Scalable Mode (Continued)

Ixia Load Module	Installed RAM	TCP Pairs	RTP or UDP Pairs	IPX / SPX Pairs
LM100TXS8	128 MB	150	150	n/a
Encryption Load Module	512 MB	500	300	n/a
ATM Load Module	256 MB	150	100	n/a
10G Ethernet LSM	512 MB	500	200	n/a

a. The TXS family includes the following load modules: LM1000TXS1, LM1000TXS4, LM1000STXS2, LM1000STX4, LM1000STXS4, OLM1000STX24, OLM1000STXS24, LM1000SFPS4, and LSM1000XMS12.

[Table 2-8](#) lists the load modules that are supported by scalable mode, and identifies the maximum number of pairs supported by the IxOS Performance Endpoint running in scalable mode.

Table 2-8. Ixia Load Module Pair Capacity - Scalable Mode

Ixia Load Module	Installed RAM	TCP Pairs	RTP or UDP Pairs	IPX / SPX Pairs
ALM1000T8	1 GB	25,000	n/a	n/a
Encryption Load Module	512 MB	12,500	n/a	n/a
10G Ethernet LSM	1 GB	9,5000	n/a	n/a

Note that scalable mode requires IxOS 4.0 or higher.

Endpoint Capabilities

Related Topics

[Performance Endpoint Support for IxChariot Functions](#) on page 2-3

[Operating System and Protocol Stack Support](#) on page 2-2

The following table indicates which Performance Endpoints have been tested with and are supported by Ixia products.

Table 2-9. Endpoint Compatibility

Endpoint	Ixia Product	
	Qcheck	IxChariot
Apple Macintosh (32-bit)	Yes	Yes
HP-UX	Yes	Yes
IBM AIX	Yes	Yes
Ixia Load Module	Yes	Yes
Linux 32-bit (x86) – TAR	Yes	Yes
Linux 32-bit (x86) – RPM	Yes	Yes
Linux 64-Bit (x86-64)	Yes	Yes
Linux on ARM	Yes	Yes
Linux on Lexra	Yes	Yes
Linux on OpenWrt	Yes	Yes
Linux on PowerPC	Yes	Yes
Microsoft Windows 32-bit (Windows NT/2000/XP/Server 2003/ Vista)	Yes	Yes
Microsoft Windows CE	Yes	Yes
Microsoft Windows 64-bit (Windows XP/Server 2003/Vista)	Yes	Yes
Microsoft Windows NT/2000/XP (Web-Based)	Yes	Yes
Sun Solaris (SPARC)	Yes	Yes
Sun Solaris Endpoint (x86)	Yes	Yes

3

Endpoint Initialization File

This chapter includes the following topics:

- [*Initialization File Description*](#) on page 3-1
- [*Keywords*](#) on page 3-3
- [*Configuring Endpoints for Large-Scale Customization*](#) on page 3-10

Initialization File Description

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 IxChariot or Qcheck Consoles.
- 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.
- Require the endpoint to accept only encrypted data during test setup.
- Set RAM and disk storage limits for payload data.
- Set the desired sockets datagram send buffer size and receive buffer size for video pairs, multicast video pairs, and IPTV pairs.
- Require a clock synchronization for each test (if the endpoints use *endpoint internal timers* as synchronizing mechanism).

endpoint.ini Example

On most operating systems, this file is named `endpoint.ini`. This file has the same format and structure on all the operating systems. Following is an example of an `endpoint.ini` file:

```

;-----
; ENDPOINT.INI file

ALLOW                                ALL
SECURITY_AUDITING                    NONE
AUDIT_FILENAME                       endpoint.aud
ENABLE_PROTOCOL                      ALL
MAX_PAYLOAD_DISK_USAGE               1073741824
MAX_PAYLOAD_MEMORY_USAGE             104857600
PAYLOAD_MEMORY_LIMIT_USAGE           10485760L
USE_ENCRYPTION                       OFF
SOCKET_SEND_BUFFER_SIZE              0
SOCKET_RECEIVE_BUFFER_SIZE           0
FORCE_CLOCKSYNC                      1
MANAGEMENT_PORT                      80
;-----

```

Keyword Default Values

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.

Table 3-1. Endpoint Initialization File Defaults

Keyword	Default Value
ALLOW	ALL
SECURITY_AUDITING	NONE
AUDIT_FILENAME	endpoint.aud
ENABLE_PROTOCOL	ALL
MAX_PAYLOAD_DISK_USAGE	1073741824
MAX_PAYLOAD_MEMORY_USAGE	104857600
PAYLOAD_MEMORY_LIMIT_USAGE	10485760
USE_ENCRYPTION	OFF
SOCKET_SEND_BUFFER_SIZE	0
SOCKET_RECEIVE_BUFFER_SIZE	0
FORCE_CLOCKSYNC	1
MANAGEMENT_PORT	10115

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 end-

points 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 [Configuring Endpoints for Large-Scale Customization](#) on page 3-10 for more information.

Keywords

This section describes the keywords that you can use in an `endpoint.ini` file:

- [ALLOW](#) on page 3-3
- [SECURITY_AUDITING](#) on page 3-4
- [AUDIT_FILENAME](#) on page 3-5
- [ENABLE_PROTOCOL](#) on page 3-6
- [USE_ENCRYPTION](#) on page 3-6
- [SOCKET_SEND_BUFFER_SIZE](#) on page 3-7
- [SOCKET_RECEIVE_BUFFER_SIZE](#) on page 3-8
- [FORCE_CLOCKSYNC](#) on page 3-8
- [MAX_PAYLOAD_DISK_USAGE](#) on page 3-9
- [MAX_PAYLOAD_MEMORY_USAGE](#) on page 3-9
- [PAYLOAD_MEMORY_LIMIT_USAGE](#) on page 3-9
- [MANAGEMENT_PORT](#) on page 3-10

ALLOW

This keyword determines which IxChariot or Qcheck Consoles can run tests using this endpoint.

To allow any IxChariot or Qcheck Console 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.

To allow only specific users to run tests with this endpoint, remove the “ALLOW ALL” line and specify one or more specific IxChariot or Qcheck Consoles 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
```

Specify a connection-oriented protocol (that is, TCP or SPX) as the first parameter and its corresponding network address as the second parameter. (Endpoints only listen for incoming tests on connection-oriented protocols.)

NOTE: Although you can *ALLOW* multiple addresses, IxChariot does not support the concurrent use of a single endpoint with multiple IxChariot consoles. That is, you cannot run tests from multiple consoles on the same endpoint at the same time.

The network address cannot be an alias or hostname; that is, 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.

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 IxChariot or Qcheck Consoles 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:

Table 3-2. Security Auditing

Parameter	Comment
NONE	Nothing is written to the audit file.
PASSED	Only access attempts that passed the <i>ALLOW</i> address check are logged.
REJECTED	Only access attempts that failed the <i>ALLOW</i> 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 3-4 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:

Table 3-3. Audit File Contents

Field	Comment
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 IxChariot Console	The network protocol used to contact Endpoint 1.
Network Address of IxChariot Console	The network address as seen by Endpoint 1. If you encounter problems setting up your <code>ALLOW</code> entries, this is the value to use for the protocol address.
Security Result	Whether this <code>SECURITY_AUDITING</code> “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
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 (refer to [ALLOW](#) on page 3-3) for information about support of the datagram protocols, IPX, RTP, and UDP.

**USE_
ENCRYPTION**

This keyword specifies whether or not the endpoint will use encrypted data during test setup. It takes the values described in [Table 3-4](#).

Table 3-4. USE_ENCRYPTION Settings

Setting	Description
OFF	The endpoint will not accept encrypted data.
ON	The endpoint will accept only encrypted data.

For Endpoint 1, this setting determines whether the endpoint will require encrypted data from the IxChariot Console. If the parameter is set to `ON`, then Endpoint 1 will reject unencrypted setup flows sent from the Console.

For Endpoint 2, this setting determines whether the endpoint will require encrypted data from Endpoint 1. If the parameter is set to `ON`, then Endpoint 2 will reject unencrypted setup flows sent from Endpoint 1,

Endpoint 1, however, can send either encrypted or unencrypted data to Endpoint 2, regardless of the setting of the `USE_ENCRYPTION` flag. The possible combinations are described in [Table 3-5](#).

Table 3-5. Effect of Encryption Settings

If Endpoint 1 USE_ENCRYPTION Setting is:	And Endpoint 2 USE_ENCRYPTION Setting is:	Then ...
OFF	OFF	Endpoint 1 accepts only unencrypted data from the Console, and sends unencrypted data to Endpoint 2.
ON	OFF	Endpoint 1 accepts only encrypted data from the Console, and sends unencrypted data to Endpoint 2.

Table 3-5. Effect of Encryption Settings (Continued)

If Endpoint 1 USE_ENCRYPTION Setting is:	And Endpoint 2 USE_ENCRYPTION Setting is:	Then ...
OFF	ON	Endpoint 1 accepts only unencrypted data from the Console, and sends encrypted data to Endpoint 2.
ON	ON	Endpoint 1 accepts only encrypted data from the Console, and sends encrypted data to Endpoint 2.

Refer to the *IxChariot User Guide* for more information about encrypting setup data.

Note: The encrypted setup flows feature is available in IxChariot release 6.30 and higher. Note that both the IxChariot Console and the IxChariot Performance Endpoints must be running a supported release level (6.30 or higher) for full feature support.

SOCKET_SEND_BUFFER_SIZE

This keyword specifies how much *send* buffer space the operating system should allocate for the sockets datagram service. The value that you specify is a request only. Each operating system sets its own maximum value and may or may not allocate all of the buffer space that you request.

This keyword is applicable only to the following pair types:

- Video pairs
- Multicast video pairs
- IPTV pairs

Syntax:

```
SOCKET_SEND_BUFFER_SIZE <value in bytes>
```

Example:

```
SOCKET_SEND_BUFFER_SIZE 32768
```

Set the value to zero to indicate that the operating system default value should be used:

```
SOCKET_SEND_BUFFER_SIZE 0
```

Note: For IPTV tests, you can also set the connection send and receive buffers through the IxChariot Console and through API calls. However, the endpoint.ini setting overrides those settings.

SOCKET_
RECEIVE_
BUFFER_SIZE

This keyword specifies how much *receive* buffer space the operating system should allocate for the sockets datagram service. The value that you specify is a request only. Each operating system sets its own maximum value and may or may not allocate all of the buffer space that you request.

This keyword is applicable only to the following pair types:

- Video pairs
- Multicast video pairs
- IPTV pairs

Syntax:

```
SOCKET_RECEIVE_BUFFER_SIZE <value in bytes>
```

Example:

```
SOCKET_RECEIVE_BUFFER_SIZE 524288
```

Set the value to zero to indicate that the operating system default value should be used:

```
SOCKET_RECEIVE_BUFFER_SIZE 0
```

Note: For IPTV tests, you can also set the connection send and receive buffers through the IxChariot Console and through API calls. However, the endpoint.ini setting overrides those settings.

FORCE_
CLOCKSYNC

This keyword specifies whether or not endpoints will synchronize their clocks before each test run.

Syntax:

```
FORCE_CLOCKSYNC <0 or 1>
```

where

- 0 specifies that a forced synchronization per test run is not required. In this case, the endpoints will periodically synchronize their clocks based on the estimated clock deviation computed from previous synchronizations.
- 1 specifies that the endpoints must synchronize their clocks before the start of each test run.

Note: This keyword is applicable only if the endpoints use *endpoint internal timers* as synchronizing mechanism. It has no effect if the clock synchronization run option is set to *Ixia hardware timestamps* or *external device*.

MAX_PAYLOAD_DISK_USAGE

This keyword defines the upper limit for the amount of payload data to be stored on permanent storage (such as hard disk).

Syntax:

```
MAX_PAYLOAD_DISK_USAGE <value in bytes>
```

Example:

```
MAX_PAYLOAD_DISK_USAGE 1073741824
```

This example limits the amount of payload data that can be stored on disk to 1 GB.

For platforms with no disk, the default value is 0 MB.

MAX_PAYLOAD_MEMORY_USAGE

This keyword defines the upper limit for the total amount of payload data that can be stored in RAM. The cumulative size of all payload files stored in memory cannot exceed this limit.

Note that payload data shared by multiple pairs will be downloaded only once. While this increases setup efficiency and reduced the total setup time, it also means that if the download of the payload fails, all pairs using that payload will abort with an error message.

Syntax:

```
MAX_PAYLOAD_MEMORY_USAGE <value in bytes>
```

Example:

```
MAX_PAYLOAD_MEMORY_USAGE 10485760
```

This example limits the amount of payload data that can be stored in RAM to 10 MB.

For platforms with no disk, the default value is 100 MB.

PAYLOAD_MEMORY_LIMIT_USAGE

This keyword specifies the maximum size of a payload file that can be stored in memory. Any payload file that exceeds this value will be stored on disk.

For example, if you set the PAYLOAD_MEMORY_LIMIT_USAGE to 5 MB, and you have one payload file that requires 4.9 MB of storage and another payload file that requires 5.01 MB, the first will be stored in memory and the second will be stored on disk,

Syntax:

```
PAYLOAD_MEMORY_LIMIT_USAGE <value in bytes>
```

Example:

```
PAYLOAD_MEMORY_LIMIT_USAGE 104857600
```

MANAGEMENT_ PORT

This example limits the amount of FEPL that can be stored in RAM to 100 MB.

For platforms with no disk, the default value is 100 MB.

This keyword specifies the TCP port that the endpoint will use for management traffic. If the keyword is omitted, the management port defaults to 10115.

Syntax:

```
MANAGEMENT_PORT <port number>
```

Example:

```
MANAGEMENT_PORT 80
```

Notes:

- For any given test, all endpoints and the IxChariot Console must use the same port number. If the port numbers differ, the test setup phase will fail.
- If you use a packet inspection firewall and you set port 80 as the management port, the firewall may reject packets carrying the management traffic.
- If the specified port is taken by another application (such as an IxChariot script), the endpoints will report an error.
- This keyword is valid for TCP only. The management port is not configurable for SPX transport: it is set at 10117.
- Clock synchronization traffic uses the same port as that selected for management traffic.

Refer to the *IxChariot User Guide* for instructions for setting the management port for the IxChariot Console.

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

```
SETUP -noinst -r -fl.\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 -fl.\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.

4

Ixia Load Module

Ixia provides Performance Endpoint software for several operating systems, including the Linux operating system that runs on Ixia load module ports. The Ixia Load Module Performance Endpoint allows you to use Ixia ports in much the same manner as other Performance Endpoints. This chapter provides instructions for using the Ixia Load Module Performance Endpoint, organized into the following chapters:

- [Installing the Ixia Performance Endpoint](#) on page 4-1
- [Logging and Messages](#) on page 4-2
- [Starting and Stopping Ixia Endpoints](#) on page 4-2
- [Supported Load Modules](#) on page 4-4

Installing the Ixia Performance Endpoint

Installation of the Ixia Load Module Performance Endpoint software is not required. Stack Manager automatically loads the Performance Endpoint software to the port CPU.

Updating the Ixia Performance Endpoint

If you need to update or replace the Ixia Load Module Performance Endpoint with another version, copy the endpoint.tar file to the C:\Program Files\Ixia\IxChariot\Packages folder, replacing the existing archive version. (You can obtain the endpoint.tar file from Customer Support, the Ixia web site, or the IxChariot distribution CD.)

About Stack Manager

IxChariot 6.10 (and higher) includes Stack Manager as an integrated tool for configuring Ixia ports for use in IxChariot tests.

Stack Manager is supported by IxOS 4.0 and higher. However, not all versions of IxOS support every feature in Stack Manager. Refer to the “IxOS Support” topic in the *Stack Manager User Guide* for a list of the features supported in each version of IxOS.

Logging and Messages

The endpoint maintains logs in `/var/log/endpoint.log`. The log file is created when an error occurs.

To view an error log, the log file should be moved over to the IxChariot Console and then viewed with the error log viewer, available in the Tools menu from the IxChariot Console main window.

Message CHR0181

You may receive message **CHR0181** while running a test. If the error was detected at the Linux computer, it says that the endpoint program on Linux has run out of system semaphores. Each instance of Endpoint 1 requires a system semaphore. The maximum number of semaphores is not configurable on Linux, which is hard-coded to a large value (128). To avoid this problem, stop other programs that use semaphores or decrease the number of tests that use the computer as Endpoint 1.

Starting and Stopping Ixia Endpoints

IxChariot Performance Endpoints on Ixia ports are automatically started when the Linux-based processor on the port is booted. If necessary, the procedures in this section can be used to stop the Performance Endpoint and restart it. One manner in which the Performance Endpoint may be restarted is to reboot the port using IxServer. This can be accomplished in one of three ways:

1. Restart *IxServer* on the chassis. This is the most extreme means of accomplishing the reboot. All use of all ports on the chassis will be immediately aborted. To accomplish this, you must:
 - a: Access the chassis' console.
 - b: Exit the running IxServer process. You will be asked for a confirmation of the termination; answer "yes".
 - c: Restart IxServer by double clicking the *IxServer* icon on the desktop.
2. Restart the individual ports using *IxServer*. To accomplish this, you must:
 - a: Access the chassis' console.
 - b: In the *IxServer* window, select *Tools..Diagnostics*.
 - c: For each port with an IxChariot endpoint that needs to be restarted:
 - i: Enter the card and the port in the fields provided.
 - ii: Press the *LP Reboot* button.
3. Follow the two steps listed below.

Stopping the Endpoint

In order to stop the IxChariot Performance Endpoint on an Ixia port, it is necessary to telnet to that port. The IP address of each port is of the form:

```
<base octet 1>.<base octet 2>.<card>.<port>
```

Base octet 1 and *base octet 2* are the first two octets of the chassis base address. The default base address is *10.0.0.0*. *Card* and *port* are the card and port number of the individual port. Thus, to telnet to the first port on card three for a chassis with a default base address, one would type:

```
telnet 10.0.3.1
```

The *user* name is *root* and no password is needed.

Once you are logged in, you are talking to a Linux system. It is necessary to find and kill all endpoint processes. Use the following two steps:

1. Type: `ps | grep endpoint`.
2. For each of the numbers in the *pid* column, type the command:

```
kill <pid>
```

Restarting the Endpoint

The IxChariot Performance Endpoint may be restarted using the following command, using the telnet session started in the previous section:

```
/bin/endpoint &
```

Supported Load Modules

To use Ixia load module ports in your IxChariot tests, you need an Ixia 400T, 1600T, 250, or Optixia chassis equipped with one or more of the following Ixia load modules¹:

Table 4-1. Load Modules Supported in IxChariot Tests

Load Module
10/100 TXS2
10/100 TXS8
10/100/1000 ALM T8
10/100/1000 CPM T8
10/100/1000 ELM ST2
10/100/1000 STX24
10/100/1000 STX4
10/100/1000 STXS2
10/100/1000 STXS24
10/100/1000 STXS4
10/100/1000 STXS4-256MB
10/100/1000 TX4
10/100/1000 TXS1
10/100/1000 TXS2
10/100/1000 TXS4
10/100/1000 TXS4-256MB
10/100/1000 XMS12
10/100/1000 XMS12-256MB
10/100/1000 XMSR12
10/100/1000 XMSR12-256MB
1000 SFP4
1000 SFPS1
1000 SFPS4
1000 SFPS4-256

1. Note the following measured limits on the number of pairs supported by these load modules: 500 for the LM1000TXS4 and ALM1000T-8 and 200 for the LM1000TXS8.

Table 4-1. Load Modules Supported in IxChariot Tests (Continued)

Load Module
1000 SFPS4-256MB
10G MSM
10G MSM POS/DCC/LAN/WAN/SRP/RPR
10G UNIPHY POS/BERT/LAN/WAN
10G UNIPHY POS/BERT/LAN/WAN 1310nm
10G UNIPHY POS/BERT/LAN/WAN DCC
10G UNIPHY POS/BERT/LAN/WAN RPR
10G UNIPHY POS/BERT/LAN/WAN SRP
10G UNIPHY POS/BERT/LAN/WAN SRP/RPR/DCC
10G UNIPHY POS/BERT/LAN/WAN SRP/RPR/DCC 1310nm
10G UNIPHY-FEC POS/BERT/LAN/WAN
10G UNIPHY-FEC POS/BERT/LAN/WAN DCC
10G UNIPHY-FEC POS/BERT/LAN/WAN RPR
10G UNIPHY-FEC POS/BERT/LAN/WAN SRP
10G UNIPHY-FEC POS/BERT/LAN/WAN SRP/RPR/DCC
10G UNIPHY-P POS/BERT/LAN/WAN
10G UNIPHY-P POS/BERT/LAN/WAN 1310nm
10G UNIPHY-P POS/BERT/LAN/WAN DCC
10G UNIPHY-P POS/BERT/LAN/WAN RPR
10G UNIPHY-P POS/BERT/LAN/WAN SRP
10G UNIPHY-P POS/BERT/LAN/WAN SRP/RPR/DCC 1310nm
10G UNIPHY-XFP LAN/WAN
10G UNIPHY-XFP POS/BERT/LAN/WAN
10G UNIPHY-XFP POS/BERT/LAN/WAN DCC
10G UNIPHY-XFP POS/BERT/LAN/WAN RPR
10G UNIPHY-XFP POS/BERT/LAN/WAN SRP
10G UNIPHY-XFP POS/BERT/LAN/WAN SRP/RPR/DCC
10G UNIPHY-XFP POS/LAN/WAN
10GE LAN
10GE LSM
10GE LSM XL6

Table 4-1. Load Modules Supported in IxChariot Tests (Continued)

Load Module
10GE LSM XM3
10GE LSM XMR3
10GE XAUI/BERT
10GE XENPAK/BERT
10GE XENPAK-P
2.5G MSM
2.5G MSM POS
2.5G MSM POS/SRP/RPR
ATM 622 Multi-Rate-256MB
ATM/POS 622 Multi-Rate
ATM/POS 622 Multi-Rate-256MB
ATM/POS 622 Multi-Rate-512MB
OC192c POS/10GE WAN
OC192c POS/BERT
OC192c POS/BERT/10GE WAN
OC192c POS/BERT/10GE WAN DCC
OC192c POS/BERT/10GE WAN RPR
OC192c POS/BERT/10GE WAN SRP
OC192c POS/BERT/10GE WAN SRP/RPR/DCC
OC192c VSR POS/BERT

For more information, refer to:

- The *Ixia Hardware Guide* for full load module specifications.
- The *Stack Manager User Guide* for a list of the load modules supported by each protocol and the number of interfaces supported by each.

5

HP-UX

This chapter explains the installation, configuration, and operation of the Performance Endpoint software for Hewlett-Packard's HP-UX 11.0 or later.

Topics in this chapter:

- [Installation Requirements for HP-UX Endpoints](#) on page 5-1
- [Endpoint Installation for HP-UX](#) on page 5-2
- [Configuring HP-UX Endpoints](#) on page 5-6
- [Running HP-UX Endpoints](#) on page 5-8
- [Logging and Messages](#) on page 5-9
- [Updates for HP-UX](#) on page 5-10

Note: Because of their lack of effective multi-threading support, HP-UX versions 9.0 and earlier are no longer supported.

Installation Requirements for HP-UX Endpoints

Here's what you need to run the endpoint program with HP-UX:

- A Hewlett-Packard computer capable of running HP-UX.
- At least 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 large tests involving hundreds of connections through a single endpoint, additional memory may be required.
- A hard disk with at least 4 MBytes of space available.
- HP-UX version 11.0 or later, with TCP/IP networking and corresponding networking hardware installed and configured. This version also supports IP Multicast.
- An Acrobat Reader to view the PDF files.

Acrobat readers are loaded on most computers for viewing other documents, but if you do not have one, they are available at Adobe's Web Site: www.adobe.com/prodindex/acrobat/readstep.html.

Endpoint Installation for HP-UX

Performance Endpoint File Name

The name of the HP-UX Performance Endpoint file is `pehpx_Mm.tar.Z`, where *Mm* is the major and minor IxChariot version number; for example *620* for IxChariot release 6.20.

Installation Procedures

First, ensure that you are logged in as a “root” user. Also, remember that all the commands and parameters discussed here are case-sensitive; use the combination of uppercase and lowercase letters as shown. The following instructions explain how to install an endpoint **from a CD-ROM** and **from the World Wide Web**.

Installation from CD-ROM

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

1. Put the CD-ROM in your CD-ROM drive.
2. Access to the CD-ROM is done through HP's Portable File System (PFS). PFS should already be configured and running on your system. For detailed information about PFS, consult your HP-UX documentation. If PFS is not running, a quick way to start it is to enter the following commands:

```
pfs_mountd -v &
pfsd -v &
```

3. If you receive an error that `pfs_mount` is not found, the command `pfs_mount` is not in your path. To find where the command is located, enter the following commands:

```
cd /
find * -name pfs_mount -print
```

4. The directory where the `pfs_mount` command is stored will then be shown. You will need to enter this path before the `pfs_mount` command.
5. Assuming your CD-ROM drive device name is `c201d4s0` and the mount point is `/cdrom`, enter the following commands. Otherwise, enter your device name and mount point instead of `c201d4s0` and `/cdrom`.

```
mkdir /cdrom
echo "/cdrom" >>/etc/pfs_exports
pfs_exportfs /cdrom
pfs_mount -v -x unix -o ro /dev/dsk/c201d4s0 /cdrom
```

6. The CD-ROM contains an archive of the endpoint package. First use the `rm` command to ensure a clean temporary install directory. Then, use the `tar` command to extract the archive contents from the CD-ROM:

```
cd /tmp
rm -fr temp tar -xvf
/cdrom/endpoint/hpux/pehpx_Mm.tar
```


7. Next, run the endpoint's installation to install our software:

```
./endpoint.install
```

8. You will see the license agreement, presented with the `more` command. Press the spacebar until the end of the agreement is displayed. You are asked whether you accept the terms and conditions of the agreement. If you do, enter "accept_license."

9. The endpoint installs itself in `/opt/Ixia`. During installation, you will see several status messages. Pay close attention to the output. If the installation is successful, you see the following message: "Installation of endpoint was successful."

10. You may instead see the following message:

```
Notice! There were potential problems with migrating from  
$oldInstallPath to $installPath. Review the warnings  
displayed above for further explanation.
```

11. If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.

12. After the installation is complete, use the `pfs_umount` command to unmount the file system from the CD-ROM:

```
pfs_umount /cdrom
```

13. If you need the disk space after installing the endpoint, you may delete the temporary directory and installation script. The installation script and temporary directory are not removed automatically.

14. To remove the temp files, enter:

```
rm -fr temp  
rm endpoint.install  
rm pehpx_Mm.tar
```

This is a good time to read the `README` file, installed with the endpoint in `/opt/Ixia`, for the latest information about the endpoint program. Use the following command to view the `README` file:

```
more /opt/Ixia/README
```

When you've completed installation, refer to [Configuring HP-UX Endpoints](#) on page 5-6 to make sure your endpoint is ready to be used in testing and monitoring.

Installation from the Web

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

1. First, use the `rm` command to ensure a clean temporary install directory (we'll use `/tmp` in this example).
2. Download the `pehpx_Mm.tar.Z` file to the `/tmp` directory.

Note: The endpoint filename is `pehpx_Mm.tar.Z`; (with a capital "Z"); however, the Internet Explorer browser you use to download it changes the filename to all lowercase. Therefore, when you specify the filename in the Save As dialog box, you should capitalize the "Z" at that time.

3. Uncompress the endpoint by using the `uncompress` command:

```
cd /tmp
uncompress pehpx_Mm.tar
tar -xvf pehpx_Mm.tar
```

4. From the directory where you've downloaded the endpoint, run the endpoint's installation script:

```
./endpoint.install
```
5. You will see the license agreement, presented with the `more` command. Press the spacebar until the end of the agreement is displayed. You are asked whether you accept the terms and conditions of the agreement. If you do, enter `"accept_license."`
6. The endpoint installs itself in `/opt/Ixia`. During installation, you will see several status messages. Pay close attention to the output. If the installation is successful, you see the following message: "Installation of endpoint was successful."
7. You may instead see the following message:

```
Notice! There were potential problems with migrating from
$oldInstallPath to $installPath. Review the warnings
displayed above for further explanation.
```
8. If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.
9. If you need the disk space after installing the endpoint, you may delete the temporary directory and installation script. The installation script and temporary directory are not removed automatically.
10. To remove the temp files, enter:

```
rm -fr temp
rm endpoint.install
rm pehpx_Mm.tar
```

This is a good time to read the `README` file, installed with the endpoint in `/opt/Ixia`, for the latest information about the endpoint program. Use the following command to view the `README` file:

```
more /opt/Ixia/README
```

When you've completed installation, refer to [Configuring HP-UX Endpoints](#) on page 5-6 to make sure your endpoint is ready to be used in testing and monitoring.

Unattended Installation for HP-UX

Unattended installation is available for the HP-UX endpoint. You can install the endpoint silently, that is, without providing additional user input.

Complete the steps, as described in [Endpoint Installation for HP-UX](#) on page 5-2 through the `tar` command. Next, run the endpoint's installation, adding the "accept_license" parameter:

```
./endpoint.install accept_license
```

What Happens During Installation

Here's what happens during the installation steps. The endpoint is installed into directory `/opt/Ixia`. The install directory is created with the following contents:

- The executable programs
- The `README` file
- Various install and uninstall programs
- 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.

- File `endpoint.ini`

See Chapter 3, [Endpoint Initialization File](#) for information about tailoring this file for individual endpoints.

The installation program stops any copy of the endpoint program that may currently be running and starts a copy of the newly installed endpoint. You can run tests immediately, without a reboot.

No changes are made to the `PATH` environment variable of the root user.

Installation also performs the following additional actions:

- Copies a startup/shutdown script to the `/sbin/init.d` directory.
- Links the startup/shutdown script to `/sbin/rc2.d/S900endpoint`. This is invoked by HP-UX when the computer boots up.
- Links the startup/shutdown script to `/sbin/rc1.d/K100endpoint`. This is invoked by HP-UX when the computer is shut down.
- Copies a configuration file to the `/sbin/rc.config.d` directory. This file should be modified to control whether the endpoint starts when your system boots. By default, the endpoint will start upon reboot.

Should you have reason to install an older endpoint, you should delete any safestore files. **Take 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:

```
rm *.q*.
```

3. Uninstall the current endpoint.
4. Install the desired endpoint.

Removing the Endpoint Package (Uninstall)

Enter the following command to remove the endpoint (you must be logged in as root to run this program):

```
/opt/Ixia/endpoint.remove
```

If the removal is successful, you see the following: “Removal of endpoint was successful.” This removes the files from `/opt/Ixia`, except for any files that were added to this directory that were not present at installation, such as the `endpoint.ini` file, or any other files you may need if you reinstall the product. For HP-UX version 11.0 systems, the removal script also leaves the `/opt/Ixia` directory.

Configuring HP-UX Endpoints

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.

1. Determine the network addresses of the computers to be used in tests.
2. Verify the network connections.

Let’s look at TCP/IP to see how to accomplish these tasks.

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.ixiacom.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

Here are two ways to determine the IP address of the local computer you're using:

- If you're using Hewlett Packard's System Administration Manager (SAM) graphical user interface, first open the Networking/Communications menu, and from there select "Network Interface Cards." A window pops up with a list of interface cards and their IP addresses.
- Alternatively, enter the following at a command prompt:

```
netstat -in
```

You may have several network interfaces. If you are using a LAN network, for example, look at the output for the `lan0` interface; your local IP address is shown in the "Address" column.

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:

```
ping xx.xx.xx.xx 64 1
```

Replace the x's with the IP address of the target computer. If Ping returns a message that says

```
1 packets transmitted, 1 packets received, 0% packet loss
```

then the Ping worked. Otherwise, there will be a delay, and then you'll see

```
1 packets transmitted, 0 packets received, 100% packet loss
```

This means that the Ping failed, and you can't reach the target computer.

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

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies "`port_number=AUTO`" on the `CONNECT_ACCEPT` command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Running HP-UX Endpoints

The following sections describe how to manually start and stop the endpoint program, and how to examine error log files if a problem occurs.

Starting an HP-UX Endpoint

On HP-UX, the endpoint program is installed so that it starts automatically each time HP-UX is rebooted. Screen output goes to file `/var/opt/Ixia/endpoint.console`. If you want to see any error messages generated at this endpoint, enter the following command:

```
tail -f /var/opt/Ixia/endpoint.console
```

The detailed information about the start and stop of each individual connection pair is written to file `endpoint.aud`. The contents of this file vary depending on how you've set the `SECURITY_AUDITING` keyword in your `endpoint.ini` file.

See Chapter 3, *Endpoint Initialization File* for more information about `endpoint.aud` and `SECURITY_AUDIT` settings.

Instead of automatic startup, you can choose to manually start the endpoint program at a command prompt. Ensure that you are logged in as a “root” user. To start the endpoint, enter:

```
/opt/Ixia/endpoint &
```

The “&” parameter indicates to HP-UX that the endpoint program should run in the background. The screen output from the endpoint program is interleaved with other UNIX commands. Just press Enter to enter more commands.

If you choose to manually start the endpoint, consider redirecting its output to the `endpoint.console` file. You can tell by the time stamp of the file when the endpoint program was started and stopped.

If the endpoint program is already running, you get the following message: “**CHR0183:** The endpoint program is already running. Only one copy is allowed at a time.”

Stopping an HP-UX Endpoint

The endpoint program has a special command-line option, `-k`. If you have an endpoint program you'd like to kill, go to a command prompt on the same computer and enter the following (you must be logged in as root to run this program):

```
/opt/Ixia/endpoint -k
```

The `-k` command-line option has the purpose of killing any endpoint process running on that computer. You should see the message “Sent exit request to the running endpoint,” which indicates that the endpoint program has been sent a request to stop.

If for some reason the request to stop is not handled by the running endpoint program correctly, you may need to use the UNIX “`kill -TERM`” command. Avoid using “`kill -9`” to stop the running endpoint program -- it doesn't clean up

what has been created (so you'll need to do the steps outlined in [Cleanup after Unexpected Errors](#) on page 5-9).

Cleanup after Unexpected Errors

If the endpoint should fail or be killed abnormally (or encounter assertion conditions), you may also need to do additional cleanup. If the endpoint is still running, try to stop it using the command `endpoint -k`. If that does not stop the endpoint, kill the endpoint using the UNIX `kill` command.

Next, enter the following command:

```
rm /var/opt/Ixia/.IXIA.ENDPOINT.PID
```

How to Tell If an HP-UX Endpoint Is Active

You can use traditional UNIX commands to determine if the endpoint program is active. At a command prompt, enter the following:

```
ps -ef | grep endpoint
```

If the endpoint program is running, you will see output similar to the following:

```
root 2516      1 0 Apr 22 ?        0:00 /opt/Ixia/endpoint
```

Disabling Automatic Startup

To disable automatic startup, edit the `/etc/rc.config.d/endpoint` file so that the `START_ON_INIT` variable is set to 0 (zero).

Messages CHR0174, CHR0204, CHR0210, or CHR0245

You may see one of these error messages if you've exceeded the soft file limit per process allowed by HP-UX. You can verify this by examining the `/var/opt/Ixia/endpoint.console` file for the following text:

```
%Internal DCE Threads problem (version CMA BL10+),  
terminating execution.  
% Reason: cma__ts_open: fd is too large  
% See 'cma_dump.log' for state information.
```

You may need to stop and restart the endpoint program using the methods outlined in [Starting an HP-UX Endpoint](#) on page 5-8 and [Stopping an HP-UX Endpoint](#) on page 5-8. You can use the HP-UX SAM facility to increase the number of open files allowed per process by changing the `maxfiles` kernel configurable parameters.

Logging and Messages

Although most error messages encountered on an endpoint are returned to the IxChariot or Qcheck Console, some may be logged to disk. Errors are logged to file `/var/opt/Ixia/endpoint.log`. To view an error log, use the program named `FMTLOG`. `FMTLOG` reads from a binary log file, and writes its formatted output to `stdout`. Here is the syntax of the `FMTLOG` command:

```
/opt/Ixia/fmtlog log_filename >output_filename
```

For example, enter the following to write a readable ASCII version of the error log to a filename `"myoutput"`:

```
/opt/Ixia/fmtlog /var/opt/Ixia/endpoint.log >myoutput
```

The endpoint code does a lot of internal checking on itself. Our software captures details related to the problem in an ASCII text file. Assertion failures are written to the file `/var/opt/Ixia/assert.err`. Save a copy of the file and send it to us via email for problem determination.

CORE and CMA_DUMP.LOG Files

We have seen situations where the endpoint core dumps on HP-UX, and the operating system writes a file named `cma_dump.log` to the directory `/opt/Ixia` or `/tmp`, and a file named `core` to `/opt/Ixia`. If a core dump occurs, please save a copy of the files `core` and `cma_dump.log` and return them to us for debugging.

Message CHR0181

You may receive the error message CHR0181 while running a test. If the error was detected at the HP-UX computer, it says that the endpoint program on HP-UX has run out of system semaphores. Each instance of Endpoint 1 requires a system semaphore. You can use the HP-UX SAM facility to increase the number of available system semaphores. Use the following procedure to change the kernel configurable parameters:

This can be done using the HP-UX SAM facility:

1. As a root user, start SAM by typing `sam`.
2. Open the Kernel Configuration menu.
3. Open the Configurable Parameters menu.
4. Update the `semmap`, `semmni`, `semmns`, and `semmnu` parameters as necessary.

After changing the kernel parameters, you must reboot HP-UX to have the changes take effect. See the HP-UX System Administration Tasks manual for the definitions of these parameters.

Updates for HP-UX

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.

Check the following Web sites for code and driver updates:

- Hewlett-Packard's Web site: www.hp.com
- HP Electronic Support Centers:
 - <http://us-support.external.hp.com/> (US, Canada, Asia-Pacific, and Latin America)
 - <http://europe-support.external.hp.com/> (Europe)

6

IBM AIX

This chapter explains the installation, configuration, and operation of the Performance Endpoint software for IBM's AIX on the RISC System/6000 (RS/6000).

Topics in this chapter:

- *Installation Requirements for AIX Endpoints* on page 6-1
- *Endpoint Installation for AIX* on page 6-2
- *Configuring AIX Endpoints* on page 6-6
- *Running AIX Endpoints* on page 6-7
- *Logging and Messages* on page 6-10
- *Updates for AIX* on page 6-10

Installation Requirements for AIX Endpoints

Here's what you need to run the endpoint program with AIX:

- An IBM RS/6000 computer capable of running AIX.
- At least 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. Large tests involving hundreds of connections through a single endpoint may require additional memory.

- A hard disk with at least 4 MBytes of space available.
- AIX version 4.1 or later, with TCP/IP networking and corresponding networking hardware installed and configured. This version also supports IP Multicast.
- An Acrobat Reader to view the PDF files.

Acrobat readers are loaded on most computers for viewing other documents, but if you do not have one, they are available at Adobe's Web site:

www.adobe.com/prodindex/acrobat/readstep.html.

Endpoint Installation for AIX

This section provides instructions for installing the AIX Performance Endpoint.

Performance Endpoint File Name

The name of the AIX Performance Endpoint file is `peaix_Mm.tar.Z`, where *Mm* is the major and minor IxChariot version number; for example *640* for IxChariot release 6.40.

Installation Procedures

First, ensure that you are logged in as a “root” user. Also, remember that all the commands and parameters discussed here are case-sensitive; use the combination of uppercase and lowercase letters as shown. The following instructions explain how to install an endpoint **from a CD-ROM** and **from the World Wide Web**.

Installation from CD-ROM

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

1. Put the endpoint CD-ROM in your CD-ROM drive.
2. Enter the following commands, assuming your CD-ROM drive device name is `cd0` and you’re able to create a temporary directory named `cdrom`:

```
mkdir /cdrom
mount -v cdrfs -r /dev/cd0 /cdrom
```

3. The CD-ROM contains an archive of the endpoint package. First, use the `rm` command to ensure a clean temporary install directory. Then use the `tar` command to extract the archive contents from the CD-ROM:

```
cd /tmp
rm -fr temp
tar -xvf /cdrom/endpoint/aix/peaix_Mm.tar
```

4. Next, run the endpoint’s installation script to install our software:

```
./endpoint.install
```

5. You will see the license agreement, presented with the “more” command. Press the spacebar until the end of the agreement is shown. You are asked whether you accept the terms and conditions of the agreement. If you do, enter “accept_license” and press Return.

The endpoint installs itself in `/usr/lpp/Ixia`. During installation, you will see several status messages. Pay close attention to the output. If the installation is successful, you see the message “Installation of endpoint was successful.”

You may instead see the following message:

```
Notice! There were potential problems with migrating from
$oldInstallPath to $installPath. Review the warnings
displayed above for further explanation.
```

If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.

6. After the installation is complete, use the `umount` command to unmount the file system from the CD-ROM:

```
umount /cdrom
```

If you need the disk space after installing the endpoint, you may delete the temporary directory and installation script. The installation script and temporary directory are not removed automatically.

To remove the temp files, enter:

```
rm -fr temp
rm endpoint.install
rm peaix_Mm.tar
```

This is a good time to read the `README` file, installed with the endpoint in `/usr/lpp/Ixia`, for the latest information about the endpoint program. Enter the `more` command to view the `README` file:

```
more /usr/lpp/Ixia/README
```

See [Configuring AIX Endpoints](#) on page 6-6 for information about your network connections.

If all connections are in order, you're ready to use this endpoint in testing and monitoring.

Installation from the Web

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

1. First, use the `rm` command to ensure a clean temporary install directory. Then save the endpoint to that directory (we'll use `/tmp` in this example).
2. Download the `peaix_Mm.tar.Z` file to the `/tmp` directory.
3. Uncompress the endpoint file by using the `uncompress` command:

```
cd /tmp
uncompress peaix_Mm.tar
tar -xvf peaix_Mm.tar
```
4. From the directory where you've downloaded the endpoint, run the endpoint's installation script to install our software:

```
./endpoint.install
```
5. You will see the license agreement, presented with the "`more`" command. Press the spacebar until the end of the agreement is shown. You are asked whether you accept the terms and conditions of the agreement. If you do, enter "`accept_license`" and press Return.

The endpoint installs itself in `/usr/lpp/Ixia`. During installation, you will see several status messages. Pay close attention to the output. If the installation is successful, you see the message "Installation of endpoint was successful."

You may instead see the following message:

```
Notice! There were potential problems with migrating from
$oldInstallPath to $installPath. Review the warnings
displayed above for further explanation.
```

If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.

Removing Temporary Files

If you need the disk space after installing the endpoint, you may delete the temporary directory and installation script. The installation script and temporary directory are not removed automatically.

To remove the temp files, enter:

```
rm -fr temp
rm endpoint.install
rm peaix_Mm.tar
```

This is a good time to read the `README` file, installed with the endpoint in `/usr/lpp/Ixia`, for the latest information about the endpoint program. Enter the `more` command to view the `README` file:

```
more /usr/lpp/Ixia/README
```

See [Configuring AIX Endpoints](#) on page 6-6 for information about your network connections.

If all connections are in order, you're ready to use this endpoint in testing and monitoring.

Unattended Installation for AIX

Unattended installation is available for the AIX endpoint. You can install the endpoint silently, that is, without providing any additional user input.

Complete the steps, as described in [Endpoint Installation for AIX](#) on page 6-2 through the `tar` command. Next, run the endpoint's installation, adding the "accept_license" parameter:

```
./endpoint.install accept_license
```

What Happens During Installation

Here's what happens during the installation steps. The endpoint is installed into the directory `/usr/lpp/Ixia`. A directory is created with the following contents:

- The executable programs.
- The `README` file.
- Various install and uninstall programs.
- 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.

- File `endpoint.ini`
- See Chapter 3, *Endpoint Initialization File* for information about tailoring this file for individual endpoints.

The installation program stops any copy of the endpoint program that may currently be running and starts a copy of the newly installed endpoint. You can run tests immediately, without a reboot.

Our software does the following so the endpoint is started every time your system boots:

- Copies the `rc.ixia` initialization script to the `/etc` directory.
- Updates `/etc/inittab` to invoke `/etc/rc.ixia`

No changes are made to the `PATH` environment variable of the root user.

Should you have reason to install an older endpoint, you should delete any safestore files using 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:

```
rm *.q*.
```
3. Uninstall the current endpoint.
4. Install the desired endpoint.

Removing the Endpoint Package (Uninstall)

Use the following command to remove the endpoint (you must be logged in as root to run this program): `/usr/lpp/Ixia/endpoint.remove`

If the removal is successful, you see the following: “Removal of endpoint was successful.”

This removes the files from `/usr/lpp/Ixia`, except for any files that were added to this directory that were not present at installation, such as the `endpoint.ini` file, and does not delete the directory. The remove program does not automatically delete files that have been added to the directory that you may need if you reinstall the product.

Configuring AIX Endpoints

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.

1. Determine the network addresses of the computers to be used in tests.
2. Verify the network connections.

Let's look at TCP/IP to see how to accomplish these tasks.

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. The alternative, domain names are in a format that is easier to recognize and remember, such as www.ixiacom.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

Here are two ways to determine the IP address of the local computer you're using:

- If you're using IBM's System Management Interface Tool (SMIT), first open the Communications Applications and Services menu, then the TCP/IP menu, and then the Minimum Configuration & Startup menu. Next, select the network interface used to reach other endpoints (for example, `en0` or `tr0`). SMIT displays the network interface's configuration; your host's IP address is in the "Internet ADDRESS" field.
- Alternatively, enter the following at a command prompt:

```
netstat -in
```

You may have several network interfaces. If you are using a LAN network, for example, look at the output for the `en0` interface; your local IP address is shown in the "Address" column.

Testing the TCP/IP Connection

Ping is a simple utility program, included in all TCP/IP implementations. To try out the connection from one computer to another, enter:

```
ping xx.xx.xx.xx 64 1
```

Replace the x's with the IP address of the target computer. If Ping returns a message that says "1 packets transmitted, 1 packets received, 0% packet loss," the Ping worked. Otherwise, there will be a delay, and then you'll see the following:

```
1 packets transmitted, 0 packets received, 100% packet loss
```

This means that the Ping failed, and you can't reach the target computer.

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

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies “port_number=AUTO” on the `CONNECT_ACCEPT` command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Maximum Value for the MSS Option

The IBM AIX Performance Endpoint supports the use of the Transmit Maximum Segment Size (MSS) option in testing. Note, however, that the highest valid MSS value is 1448 (12 bytes lower than that of the other operating systems). If you set the MSS value higher than 1448, the test will report a CHR0208 error.

Refer to “Setting the Transmit MSS Option” in the *IxChariot Scripts Development and Editing Guide* for additional information.

Running AIX Endpoints

The following sections describe how to manually start and stop the endpoint program, and how to examine error log files if a problem occurs.

Starting an AIX Endpoint

The endpoint program is installed so that it starts automatically each time AIX is rebooted. It sends its screen output to file `/var/adm/endpoint.console`. If you want to see any error messages generated at this endpoint, enter the following command:

```
tail -f /var/adm/endpoint.console
```

The detailed information about the start and stop of each individual connection pair is written to file `endpoint.aud`. The contents of this file vary depending on how you've set the `SECURITY_AUDITING` keyword in your `endpoint.ini` file.

See Chapter 3, *Endpoint Initialization File* for more information about `endpoint.aud` and `SECURITY_AUDIT` settings.

Instead of automatic startup, you can choose to manually start the endpoint program at a command prompt. Ensure that you are logged in as a “root” user. To start the endpoint, enter the following:

```
/usr/lpp/Ixia/endpoint &
```

The “&” parameter indicates to AIX that the endpoint program should run in the background. The screen output from the endpoint program is interleaved with other UNIX commands. Just press Return to enter more commands.

If you choose to manually start the endpoint, consider redirecting its output to the `endpoint.console` file. You can tell by the time stamp of the file when the endpoint program was started and stopped.

If the endpoint program is already running, you get the following message: “**CHR0183:** The endpoint program is already running. Only one copy is allowed at a time.”

Stopping an AIX Endpoint

The endpoint program has a special command-line option, `-k`. If you have an endpoint program you’d like to kill, go to a command prompt on the same computer and enter the following (you must be logged in as root to run this program):

```
/usr/lpp/Ixia/endpoint -k
```

The `-k` command-line option has the purpose of killing any endpoint process running on that computer. You should see the message “Sent exit request to the running endpoint,” which indicates that the endpoint program has been sent a request to stop.

If for some reason the request to stop is not handled by the running endpoint program correctly, you may need to use the UNIX “`kill -TERM`” command.

Cleanup after Unexpected Errors

If the endpoint should fail or be killed abnormally (or encounter assertion conditions), you may also need to do additional cleanup. If the endpoint is still running, try to stop it using the command “`endpoint -k`”. If that does not stop the endpoint, kill the endpoint using the UNIX “`kill`” command.

Next, enter the following command:

```
rm /var/adm/.IXIA.ENDPOINT.PID
```

How to Tell If an AIX Endpoint Is Active

You can use traditional UNIX commands to determine if the endpoint program is active. At a command prompt, enter:

```
ps -ef | grep endpoint
```

If the endpoint program is running, you will see output similar to this:


```
root 9888 1 0 19:19:54 - 0:00 /usr/lpp/Ixia/endpoint -G
7477 -T 3
root 7477 1 0 18:37:47 - 0:00 /usr/lpp/Ixia/endpoint
```

Disabling Automatic Startup

To disable automatic startup, comment out or remove the following lines from the /etc/rc.ixia script:

```
if test -f $installPath/endpoint; then
echo "Starting the Ixia Endpoint."
$installPath/endpoint 1>$outputPath/endpoint console 2>&1
&
fi
```

Logging and Messages

Although most error messages encountered on an endpoint are returned to the IxChariot or Qcheck Console, some may be logged to disk. Errors are saved in a file named *endpoint.log*, in the */var/adm* directory. To view an error log, use the program named *FMTLOG*. *FMTLOG* reads from a binary log file, and writes its formatted output to *stdout*. Use the following *FMTLOG* command:

```
/usr/lpp/Ixia/fmtlog /var/adm/endpoint.log  
>output_filename
```

The endpoint code does a lot of internal checking on itself. Our software captures details related to the problem in an ASCII text file named *assert.err* in the */var/adm* directory. Save a copy of the file and send it to us via email for problem determination.

Message CHR0181

You may receive message **CHR0181** while running a test. If the error was detected at the AIX computer, it says that the endpoint program on AIX has run out of system semaphores. Each instance of Endpoint 1 requires a system semaphore. The maximum number of semaphores is not configurable on AIX; it is hard-coded to a large value (4096). To avoid this problem, stop other programs that use semaphores, or decrease the number of connection pairs that use the AIX computer as Endpoint 1.

Updates for AIX

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.

Check the following Web site for code and driver updates:

<http://techsupport.services.ibm.com/rs6000/support>

7

Linux 32-Bit (x86)

This chapter explains the installation, configuration, and operation of the Performance Endpoint software for 32-bit Linux running on Intel x86 processors.

Topics in this chapter:

- [Linux 32-bit x86 Performance Endpoint File Names](#) on page 7-1
- [Protocols Supported](#) on page 7-2
- [Installation Requirements](#) on page 7-2
- [Installing 32-bit Linux x86 Endpoints](#) on page 7-3
- [Removing 32-bit Linux x86 Endpoints](#) on page 7-8
- [Configuring 32-bit Linux x86 Performance Endpoints](#) on page 7-9
- [Running Linux Endpoints](#) on page 7-11
- [Logging and Messages](#) on page 7-13

Endpoints are also available for the Linux AMD64 processor architecture. Refer to Chapter 8, [Linux 64-Bit \(x86-64\)](#) for more information.

Linux 32-bit x86 Performance Endpoint File Names

Ixia provides two versions of the 32-bit Linux x86 Performance Endpoint:

- `pelnx_Mm.tar.gz` – Zipped tar file
- `pelnx_Mm.rpm` – RPM (Red Hat Package Manager) distribution file

where “M” represents the major version and “m” represents the minor version. For example, `pelnx_650.rpm` is version 6.50 of the RPM distribution Performance Endpoint.

The two Performance Endpoints are the same: only the distribution method differs. This chapter provides installation instructions for both.

Protocols Supported

The 32-bit Linux x86 Performance Endpoint uses the Sockets interface shipped with Linux to support the following transport layer protocols:

- TCP
- UDP
- RTP

The Performance Endpoint supports both IPv4 and IPv6. IPX, SPX, and other network protocols are not supported.

Installation Requirements

Here are the requirements for installing and running the Linux x86 Performance Endpoint:

- A 32-bit x86 computer capable of running Linux well. We recommend a CPU such as an Intel Pentium III or better.
- A minimum of 64 MBytes of RAM.

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

- A hard disk with at least 8 MBytes of space available.
- Linux kernel 2.4.20 or better.

Note: Linux kernel 2.6.18 is required for IPTV testing.

We have tested with Linux distributions that implement Linux kernel 2.4.20. We have not tested this version of Performance Endpoint with earlier versions of the Linux kernel.

The Performance Endpoint requires the Linux operating system to enable “pthreads support” (which is at least version 2.0.6 of glibc). TCP/IP networking and corresponding networking hardware must be installed and configured, plus ELF support. Some older installations of Linux may not have this installed. At the Web site www.linuxdoc.org/HOWTO/Glibc2-HOWTO.html, you can find information about Linux, as well as download the file `glibc-2.0`, which you need to have loaded to install the endpoint. We have changed our installation procedures to check for this file, as it is required to run the endpoint.

Note that older versions of the Linux kernel may not properly support IPv6.

- An Acrobat Reader to view PDF files. Acrobat readers are loaded on most computers for viewing other documents, but if you do not have one, they are available at Adobe’s Web Site: www.adobe.com/prodindex/acrobat/read-step.html.

Installing 32-bit Linux x86 Endpoints

This section provides installation instructions for both versions of the 32-bit Linux x86 Performance Endpoint:

- [TAR-Based Endpoint Installation for 32-bit Linux](#) on page 7-3
- [RPM-Based Endpoint Installation for 32-bit Linux](#) on page 7-5
- [What Happens During Installation](#) on page 7-7

TAR-Based Endpoint Installation for 32-bit Linux

First, make sure that you are logged in as a “root” user. Also, remember that all commands and parameters discussed here are case-sensitive. Use the combination of uppercase and lowercase letters as shown in the following procedure. The following instructions explain how to install an endpoint from a CD-ROM and from the Ixia web site.

Installation from CD-ROM

To install the 32-bit Linux x86 Performance Endpoint from a CD-ROM:

1. Log in as *root*.
2. Put the CD-ROM in your CD-ROM drive.
3. Enter the following commands, assuming your CD-ROM drive device name is `/dev/cdrom` and you are able to create a temporary directory named `cdrom`:

```
mkdir /cdrom
mount /dev/cdrom /cdrom
```
4. The CD-ROM contains an archive of the endpoint package. First use the `rm` command to ensure a clean temporary install directory. Then use the `tar` command to extract the archive contents from the CD-ROM:

```
cd /tmp
rm -fr temp
tar -xvf /cdrom/endpoint/linux/pelnx_Mm.tar
```
5. Next, run the endpoint’s installation script to install the endpoint:

```
./endpoint.install
```
6. You will see the license agreement, presented with the “more” command. Press the spacebar until the end of the agreement is displayed. You are asked whether you accept the terms and conditions of the agreement. If you do, enter “accept_license.”

The endpoint installs itself in `/usr/local/ixia`. During installation you will see several status messages. When the installation is successful, you see the message “Installation of endpoint was successful.”

You may instead see the following message:

Notice! There were potential problems with migrating from `$oldInstallPath` to `$installPath`. Review the warnings displayed above for further explanation.

If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.

7. After the installation is complete, use the `UMOUNT` command to unmount the file system from the CD-ROM:

```
umount /cdrom
```

The installation script and temporary directory are not removed automatically if the installation is successful. If you need the disk space after installing the endpoint, you may delete the temporary directory and installation script.

8. To remove the temp files, enter:

```
rm -fr temp
rm endpoint.install
```

This is a good time to read the `README` file, installed with the endpoint in `/usr/local/ixia`, for the latest information about the endpoint program. Enter the `more` command to view the `README` file:

```
more /usr/local/Ixia/README
```

When you've completed installation, refer to [Configuring 32-bit Linux x86 Performance Endpoints](#) on page 7-9 to make sure your endpoint is ready to be used in testing and monitoring.

Installation from the Web

To install an endpoint you've downloaded from the Ixia web site:

1. Log in as `root`.
2. Use the `rm` command to ensure a clean temporary install directory (we'll use `/tmp` in this example).
3. Save the endpoint to the `/tmp` directory.
4. Uncompress the endpoint by using the `uncompress` command:

```
cd /tmp
uncompress pelnx_Mm.tar
tar -xvf pelnx_Mm.tar
```

5. From the directory where you've downloaded the endpoint, run the endpoint's installation script:

```
./endpoint.install
```

The endpoint installs itself in `/usr/local/ixia`. During installation, you will see several status messages. When the installation is successful, you see the message "Installation of endpoint was successful."

You may instead see the following message:

Notice! There were potential problems with migrating from `$oldInstallPath` to `$installPath`. Review the warnings displayed above for further explanation.

If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.

The installation script and temporary directory are not removed automatically if the installation is successful. If you need the disk space after installing the endpoint, you may delete the temporary directory and installation script.

6. To remove the temp files, enter:

```
rm -fr temp
rm endpoint.install
```

This is a good time to read the `README` file, installed with the endpoint in `/usr/local/ixia`, for the latest information about the endpoint program. Enter the `more` command to view the `README` file:

```
more /usr/local/Ixia/README
```

When you've completed installation, refer to [Configuring 32-bit Linux x86 Performance Endpoints](#) on page 7-9 to make sure your endpoint is ready to be used in testing and monitoring.

Unattended Installation for TAR-Based Linux

You can install the endpoint silently, that is, without providing any additional user input.

Complete the steps, as described in [TAR-Based Endpoint Installation for 32-bit Linux](#) on page 7-3, through the `tar` command. Next, run the endpoint's installation, adding the "accept_license" parameter:

```
./endpoint.install accept_license
```

RPM-Based Endpoint Installation for 32-bit Linux

First, ensure that you are logged in as a "root" user. Also, remember that all commands and parameters discussed here are case-sensitive. Use the combination of uppercase and lowercase letters as shown in the following procedure. The following instructions explain how to install an endpoint from a CD-ROM and from the World Wide Web.

Installation from CD-ROM

To install the 32-bit Linux x86 Performance Endpoint from a CD-ROM:

1. Log in as *root*.
2. Put the CD-ROM in your CD-ROM drive.
3. Enter the following commands, assuming your CD-ROM drive device name is `/dev/cdrom` and you are able to create a temporary directory named `cdrom`:

```
mkdir /cdrom
mount /dev/cdrom /cdrom
```
4. Copy the `pelnx_Mm.rpm` file from the CD-ROM to a local directory (like `tmp`, for example).

```
cp /cdrom/endpoint/linux/pelnx_Mm.rpm /tmp
```

5. Use the `RPM` command to install the endpoint:

```
rpm -Uvh /tmp/pelnx_Mm.rpm
```

During installation, you will see several status messages. Pay close attention to the output. When the installation is successful, you see the message “Installation of endpoint was successful.”

You may instead see the following message:

```
Notice! There were potential problems with migrating from
$oldInstallPath to $installPath. Review the warnings
displayed above for further explanation.
```

If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.

6. After the installation is complete, use the `UMOUNT` command to unmount the file system from the CD-ROM.

```
umount /cdrom
```

This is a good time to read the `README` file, installed with the endpoint in `/usr/local/ixia`, for the latest information about the endpoint program. Enter the `more` command to view the `README` file:

```
more /usr/local/Ixia/README
```

When you’ve completed installation, refer to [Configuring 32-bit Linux x86 Performance Endpoints](#) on page 7-9 to make sure your endpoint is ready to be used in testing and monitoring.

Installation from the Web

To install an endpoint you’ve downloaded from the Ixia web site:

1. Log in as `root`.
2. Use the `rm` command to ensure a clean temporary install directory (we’ll use `/tmp` in this example).
3. Save the `pelnx_Mm.rpm` file to the `/tmp` directory.
4. Use the `RPM` command to install the endpoint:

```
cd /tmp
rpm -Uvh pelnx_Mm.rpm
```

During installation, you will see several status messages. Pay close attention to the output. When the installation is successful, you see the message “Installation of endpoint was successful.”

You may instead see the following message:

```
Notice! There were potential problems with migrating from
$oldInstallPath to $installPath. Review the warnings
displayed above for further explanation.
```

If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.

This is a good time to read the `README` file, installed with the endpoint in `/usr/local/ixia`, for the latest information about the endpoint program. Enter the `more` command to view the `README` file:

```
more /usr/local/Ixia/README
```

When you've completed installation, refer to [Configuring 32-bit Linux x86 Performance Endpoints](#) on page 7-9 to make sure your endpoint is ready to be used in testing and monitoring.

What Happens During Installation

Here is what happens during the installation steps. The endpoint is installed into the directory `/usr/local/ixia`. A directory is created with the following contents:

- The executable programs
- The `README` file
- Various install and uninstall programs
- 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 Chapter 3, [Endpoint Initialization File](#) for information about tailoring this file for individual endpoints.

The installation program stops any copy of the endpoint program currently running and starts a copy of the newly installed endpoint. You can run tests immediately, without restarting your computer.

Our software displays information on how to update your system to have the endpoint start automatically upon restarting.

No changes are made to the `PATH` environment variable of the root user.

Should you have reason to install an older endpoint, you should delete any safestore files using 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:

```
rm *.q*.
```

3. Uninstall the current endpoint.
4. Install the desired endpoint.

Removing 32-bit Linux x86 Endpoints

Instructions for uninstalling 32-bit Linux x86 Performance Endpoints is provided below, for both TAR-based packages and RPM-based packages.

Removing the TAR-Based Endpoint Package (Uninstall)

You must be logged in as root to remove the endpoint package.

If you need to remove the endpoint package from your hard disk, first stop the endpoint program (if it is running) using the following command:

```
/usr/local/Ixia/endpoint -k
```

Then use the following command to remove the endpoint:

```
/usr/local/Ixia/endpoint.remove
```

If the removal is successful, you will see the following: “Removal of endpoint was successful.” This removes the files from `/usr/local/Ixia`, except for any files that were added to this directory that were not present at installation, such as the `endpoint.ini` file. This command does not delete the directory. The remove program does not automatically delete files added to the directory that you may need if you reinstall the product.

If anything goes wrong during the process of uninstalling the endpoint, a reinstalled endpoint may not run. You may need to do some extra cleanup. Check for the hidden file `/var/local/Ixia/.IXIA.ENDPOINT.PID` by using the `ls -a` command. This file should be manually removed. Enter the following command:

```
rm /var/local/Ixia/.IXIA.ENDPOINT.PID
```

Removing the RPM Endpoint Package (Uninstall)

You must be logged in as *root* to remove the endpoint package.

Use the following command to uninstall the Linux x86 Performance Endpoint:

```
rpm -e endpoint
```

If the removal is successful, you will see the following: “Removal of endpoint was successful.” This removes the files from `/usr/local/ixia`, except for any files that were added to this directory that were not present at installation, such as the `endpoint.ini` file. This command does not delete the directory. The remove program does not automatically delete files added to the directory that you may need if you reinstall the product.

If anything goes wrong during the process of uninstalling the endpoint, a reinstalled endpoint may not run. You may need to do some extra cleanup. Check for the hidden file `/usr/local/Ixia/.IXIA.ENDPOINT.PID`. You can use the command `ls -a` to view hidden files. Then enter the following command to delete it:

```
rm /usr/local/Ixia/.IXIA.ENDPOINT.PID
```

Configuring 32-bit Linux x86 Performance Endpoints

The 32-bit Linux x86 Performance 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. Take the following steps to verify that your network is ready for testing and/or monitoring:

1. Determine the network addresses of the computers for use in tests.
2. Verify the network connections.

Let's look at TCP/IP to see how to accomplish these tasks.

Configuration for TCP/IP

The 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. The alternative, domain names are in a format that is easier to recognize and remember, such as www.ixiacom.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 for TAR and RPM Linux

To determine the IP address of the local computer you are using, enter the following at a command prompt:

```
/sbin/ifconfig
```

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies “`port_number=AUTO`” on the `CONNECT_ACCEPT` command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Testing the TCP Connection

Ping is a simple utility program, included in all TCP/IP implementations. To try out the connection from one computer to another, enter the following:

```
ping xx.xx.xx.xx -c 1
```

Replace the x's with the IP address of the target computer. If Ping returns a message that says

```
1 packets transmitted, 1 packets received, 0% packet loss
```

then the Ping worked. Otherwise, there will be a delay, and you'll see

```
1 packets transmitted, 0 packets received, 100% packet loss
```

This means that the Ping failed, and you cannot reach the target computer.

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

Autostarting the Endpoint

For the endpoint to automatically start when your computer restarts, you must update your system `rc` scripts.

If your Linux system uses `rc.local`, which is used by some older Linux systems, add the following line to the `rc.local` file:

```
/usr/local/Ixia/endpoint 1>>/var/local/endpoint.console  
2>&1 &
```

Don't forget the ampersand (&) at the end of the line. If this character is not included, the boot process does not continue, and you may be unable to log in at the Console.

If you have previously installed the endpoint in a `Ganymede` directory, the install script displays the following message:

```
The endpoint install directory now uses $installPath  
instead of $oldInstallPath. If your rc.local referenced  
$oldInstallPath, you should change it to use the new  
directory.
```

If your Linux system is more recent, it probably supports System V `init rc` scripts. Red Hat software uses this type of `init rc` files. Copy `usr/local/ixia/rc2exec.lnx` to the appropriate places. For example, with Red Hat Linux 5.0, you may run these commands:

```
cp /usr/local/Ixia/rc2exec.lnx /etc/rc.d/init.d/endpoint  
ln -fs /etc/rc.d/init.d/endpoint /etc/rc.d/rc2.d/  
S81endpoint  
ln -fs /etc/rc.d/init.d/endpoint /etc/rc.d/rc3.d/  
S81endpoint  
ln -fs /etc/rc.d/init.d/endpoint /etc/rc.d/rc6.d/  
K81endpoint
```

For Red Hat Linux 5.2 or later, the recommended commands are the following:

```
cp /usr/local/Ixia/rc2exec.lnx /etc/rc.d/init.d/endpoint  
/sbin/chkconfig endpoint reset
```

Running Linux Endpoints

The following sections describe how to manually start and stop the endpoint program, and how to examine error log files if a problem occurs.

Starting a Linux Endpoint

The endpoint program is installed so that it starts automatically each time Linux is rebooted.

- On Slackware, it sends its screen output to file `/var/adm/endpoint.console`.
- On Red Hat, it sends its screen output to file `/var/local/endpoint.console`.

If you want to see any error messages generated at this endpoint, enter the following command:

```
tail -f /var/local/endpoint.console
```

The detailed information about the start and stop of each individual connection pair is written to file `endpoint.aud`. The contents of this file vary depending on how you've set the `SECURITY_AUDITING` keyword in your `endpoint.ini` file.

See Chapter 3, [Endpoint Initialization File](#) for more information about `endpoint.aud` and `SECURITY_AUDIT` settings.

Instead of automatic startup, you can choose to manually start the endpoint program at a command prompt. Ensure that you are logged in as a “root” user. To start the endpoint, enter the following:

```
/usr/local/Ixia/endpoint &
```

The “&” parameter indicates to Linux that the endpoint program should run in the background. The screen output from the endpoint program is interleaved with other UNIX commands. Just press **Return** to enter more commands.

If you choose to manually start the endpoint, consider redirecting its output to the `endpoint.console` file. You can tell by the time stamp of the file when the endpoint program was started or stopped.

If the endpoint program is already running, you get the following message, “**CHR0183**: The endpoint program is already running. Only one copy is allowed at a time.”

Use the `ps` command to check all running processes and make sure the endpoint is running (see the section, [How to Tell If a Linux Endpoint Is Active](#) on page 7-12 for more information). If you repeatedly get error message **CHR0183** but it appears that the endpoint is not running, you may need to do some extra cleanup.

Check for the hidden file `/usr/local/Ixia/IXIA.ENDPOINT.PID` by using the `ls -a` command. This file should be manually removed.

Stopping a Linux Endpoint

The endpoint program has a special command-line option, `-k`. If you'd like to kill an endpoint program, go to a command prompt on the same computer and enter the following (you must be logged in as root to run this program):

```
/usr/local/Ixia/endpoint -k
```

The `-k` command-line option has the purpose of killing any endpoint process running on that computer. You should see the message "Sent exit request to the running endpoint," which indicates that the endpoint program has been sent a request to stop.

If, for some reason, the request to stop is not handled correctly by the running endpoint program, you may need to use the UNIX "kill -TERM" command. Avoid using "kill -9" to stop the running endpoint program—it doesn't clean up what's been created (so you'll need to do the steps outlined in [Cleanup after Unexpected Errors](#) on page 7-12).

Cleanup after Unexpected Errors

If the endpoint should fail or be killed abnormally (or encounter assertion conditions), you may also need to do additional cleanup. If the endpoint is still running, try to stop it using the command "endpoint -k". If that does not stop the endpoint, kill the endpoint using the UNIX `kill` command.

Then enter the following command:

```
rm /usr/local/Ixia/.IXIA.ENDPOINT.PID
```

How to Tell If a Linux Endpoint Is Active

Use traditional UNIX commands to determine if a Linux endpoint is active. At a command prompt, enter:

```
ps axf | grep endpoint
```

If the endpoint program is running, you will see output similar to this:

```
366 p0 S 0:00 \_ /usr/local/Ixia/endpoint
367 p0 S 0:00 | \_ /usr/local/Ixia/endpoint
368 p0 S 0:00 | \_ /usr/local/Ixia/endpoint
369 p0 S 0:00 | \_ /usr/local/Ixia/endpoint
```

Disabling Automatic Startup

If you run a Linux system that uses `rc.local` to invoke applications, remove the invocation of `/usr/local/Ixia/endpoint` from `/etc/rc.d/rc.local`.

If you use a Linux system that supports System V style `init rc` scripts, remove `/etc/rc.d/rc2.d/S81endpoint` from `/etc/rc.d/rc2.d`.

If you are using Red Hat Linux versions 5.2 or later, and have enabled the automatic startup through the `CHKCONFIG` utility, you can also disable the automatic startup through the `CHKCONFIG` utility. Here is the syntax to use the `CHKCONFIG` utility to disable the automatic startup:

```
/sbin/chkconfig -del endpoint
```

Logging and Messages

While most error messages encountered on an endpoint are returned to the IxChariot or Qcheck Console, some may be logged to disk. Errors are saved in the following file:

```
/var/log/endpoint.log
```

A log file is not created until an error occurs. To view an error log, use the program named `FMTLOG`. `FMTLOG` reads from a binary log file, and writes its formatted output to `stdout`. Use the following `FMTLOG` command:

```
/usr/local/Ixia/fmtlog /var/log/endpoint.log  
>output_filename
```

The endpoint code does a lot of internal checking on itself. Our software captures details related to the problem in an ASCII text file:

```
/var/local/assert.err
```

Save a copy of the file and send it to us via email for problem determination.

Message CHR0181

You may receive message **CHR0181** while running a test. If the error was detected at the Linux computer, it says that the endpoint program on Linux has run out of system semaphores. Each instance of Endpoint 1 requires a system semaphore. The maximum number of semaphores is not configurable on Linux, which is hard-coded to a large value (128). To avoid this problem, stop other programs that use semaphores or decrease the number of tests that use the computer as Endpoint 1.

Increasing the Number of Concurrent Connections

Some parameters are tuned in Linux by rebuilding the Linux kernel. If you're adventurous and skilled enough, you can change the number of concurrent endpoint connections. Consult your Linux documentation for information about increasing the maximum open files allowed per process (this probably involves redefining `NR_FILES` and other macros). Alternatively, search Linux newsgroups on the Internet (using DejaNews, for example) for something like "max open files per process."

8

Linux 64-Bit (x86-64)

The following topics explain the installation, configuration, and operation of the Performance Endpoint software for 64-bit Linux distributions running on x86-64-based architectures.

Topics in this chapter:

- [Supported Distributions](#) on page 8-1
- [Installation Requirements](#) on page 8-1
- [Installing Linux 64-Bit Performance Endpoints](#) on page 8-2
- [Removing 64-Bit Linux Endpoints](#) on page 8-6
- [Configuring 64-Bit Linux Endpoints](#) on page 8-7
- [Running 64-Bit Linux Endpoints](#) on page 8-8
- [Logging and Messages](#) on page 8-10

Supported Distributions

These performance endpoints are supported on the following 64-bit Linux distributions:

- Redhat Enterprise Server
- SuSE 9.2

Refer to Chapter 7, [Linux 32-Bit \(x86\)](#), of this manual for detailed information about Performance Endpoints for 32-bit versions of Linux.

Installation Requirements

Here is what you need to run the endpoint program with 64-bit Linux:

- A computer capable of running a 64-bit Linux distribution.

The 64-bit version of the Linux Performance Endpoint requires an x86-64-compatible CPU, such as the AMD Opteron and AMD Athlon processors, or Intel Pentium 4 and Xeon processors with EM64T.

- At least 64 MBytes of random access memory (RAM).
The total RAM requirement depends on RAM usage of the underlying protocol stack and the number of concurrent endpoint pairs. For tests involving over one hundred connections through a single endpoint, additional memory may be required.
- A hard disk with at least 24 MBytes of space available.
- Linux kernel 2.4 with *pthread*s support (which is at least version 2.2 of *glibc*). TCP/IP networking and corresponding networking hardware must be installed and configured, plus ELF support.
- Linux kernel 2.4.20 is required for IPv6 support.
- An Acrobat Reader to view the PDF files.

Acrobat readers are loaded on most computers for viewing other documents, but if you don't have one, they are available at Adobe's Web Site:

www.adobe.com/prodindex/acrobat/readstep.html.

Installing Linux 64-Bit Performance Endpoints

This section provides installation instructions for both versions of the Linux 64-bit Performance Endpoint:

- *TAR-Based Installation of the x86-64 Linux Endpoint* on page 8-2
- *RPM-Based Installation for the x86-64 Linux Endpoint* on page 8-4
- *What We Do During Installation* on page 8-5

TAR-Based Installation of the x86-64 Linux Endpoint

First, make sure that you are logged in as a “root” user. Also, remember that all commands and parameters discussed here are case-sensitive. Use the combination of uppercase and lowercase letters shown. The following instructions explain how to install an endpoint **from a CD-ROM** and **from the World Wide Web**.

Installation from CD-ROM

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

1. Put the CD-ROM in your CD-ROM drive.
2. Enter the following commands, assuming your CD-ROM drive device name is `/dev/cdrom` and you are able to create a temporary directory named `cdrom`:

```
mkdir /cdrom
mount /dev/cdrom /cdrom
```

3. The CD-ROM contains an archive of the endpoint package. First use the `rm` command to ensure a clean temporary install directory. Then use the `uncompress` and `tar` commands to extract the archive contents from the CD-ROM:

```
cd /tmp
rm -fr temp
```

Use the following commands to extract the archive contents:

```
uncompress /cdrom/endpoint/linux  
            /pelinux_amd64_610.tar.gz  
tar -xvf ./pelinux_amd64_610.tar
```

4. Next, run the endpoint's installation script to install the endpoint:

```
./endpoint.install
```

5. You will see the license agreement, presented with the “more” command. Press the space bar until the end of the agreement is displayed. You are asked whether you accept the terms and conditions of the agreement. If you do, enter “accept_license” and press the ENTER key.

The endpoint installs itself in `/usr/local/Ixia`. During installation you will see several status messages. When the installation is successful, you see the message “Installation of endpoint was successful.”

After the installation is complete, use the `UMOUNT` command to unmount the file system from the CD-ROM:

```
umount /cdrom
```

The installation script and temporary directory are not removed automatically if the installation is successful. If you need the disk space after installing the endpoint, you may delete the temporary directory and installation script.

To remove the temp files, enter:

```
rm -fr temp  
rm endpoint.install
```

This is a good time to read the `README` file, installed with the endpoint in `/usr/local/Ixia`, for the latest information about the endpoint program. Enter the `more` command to view the `README` file:

```
more /usr/local/Ixia/README
```

When you've completed installation, your endpoint should be ready to be used in testing and monitoring.

Installation from the Web

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

1. First use the `rm` command to ensure a clean temporary install directory (we'll use `/tmp` in this example).

```
cd /tmp  
rm -fr temp
```

2. Save the endpoint to the `/tmp` directory.
3. Use the `uncompress` and `tar` commands to extract the archive contents.

Use the following commands to extract the archive contents:

```
uncompress ./pelinux_amd64_610.tar.gz
tar -xvf ./pelinux_amd64_610.tar
```

4. From the directory where you've downloaded the endpoint, run the endpoint's installation script:


```
./endpoint.install
```
5. You will see the license agreement, presented with the "more" command. Press the space bar until the end of the agreement is displayed. You are asked whether you accept the terms and conditions of the agreement. If you do, enter "accept_license" and press the ENTER key.

The endpoint installs itself in `/usr/local/Ixia`. During installation you will see several status messages. When the installation is successful, you see the message "Installation of endpoint was successful."

The installation script and temporary directory are not removed automatically if the installation is successful. If you need the disk space after installing the endpoint, you may delete the temporary directory and installation script.

To remove the temp files, enter:

```
rm -fr temp
rm endpoint.install
```

This is a good time to read the `README` file, installed with the endpoint in `/usr/local/Ixia`, for the latest information about the endpoint program. Enter the `more` command to view the `README` file:

```
more /usr/local/Ixia/README
```

When you've completed installation, your endpoint should be ready to be used in testing and monitoring.

Unattended Installation

You can install the endpoint silently: that is, without providing any additional user input.

Complete the first three steps in the procedures described above (through the `tar` command). Next, run the endpoint's installation, adding the "accept_license" parameter:

```
./endpoint.install accept_license
```

RPM-Based Installation for the x86-64 Linux Endpoint

Use the RPM-based installation if you are installing the endpoint on Red Hat or SuSE 64-bit Linux distributions.

First, make sure that you are logged in as "root". Also, remember all commands and parameters discussed here are case-sensitive. Use the combination of upper-case and lowercase letters as shown in the text. The following instructions explain how to install an endpoint **from a CD-ROM** and **from the World Wide Web**.

Installation from CD-ROM

The following instructions describe how to install the endpoint on a computer with a CD-ROM drive.

1. Put the CD-ROM in your CD-ROM drive.
2. Enter the following commands, assuming your CD-ROM drive device name is `/dev/cdrom` and you are able to create a temporary directory named `cdrom`:

```
mkdir /cdrom
mount /dev/cdrom /cdrom
```

3. Copy the RPM file from the drive to a local directory (for example, `tmp`).

```
cp /cdrom/endpoint/linux/pelinux_amd64_610.rpm /tmp
```

4. Use the RPM command to install the endpoint:

```
rpm -Uvh /tmp/pelinux_amd64_610.rpm
```

5. After the installation is complete, use the `UMOUNT` command to unmount the file system from the CD-ROM:

```
umount /cdrom
```

During installation, you will see several status messages. Pay close attention to the output. When the installation is successful, you see the message “Installation of endpoint was successful.”

Installation from the Web

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

1. First, use the `rm` command to ensure a clean temporary install directory (we’ll use `/tmp` in this example).

```
cd /tmp
rm -fr temp
```

2. Save the endpoint to the `/tmp` directory.

3. Use the RPM command to install the endpoint:

```
rpm -Uvh /tmp/pelinux_amd64_610.rpm
```

During installation, you will see several status messages. Pay close attention to the output. When the installation is successful, you see the message “Installation of endpoint was successful.”

What We Do During Installation

Here is what happens during the installation steps. The endpoint is installed into the directory `/usr/local/Ixia`. A directory is created with the following contents:

- the executable programs;
- the `README` file;
- various install and uninstall programs;

- 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 Chapter 3, *Endpoint Initialization File* for information about tailoring this file for individual endpoints.

The installation program stops any copy of the endpoint program currently running and starts a copy of the newly installed endpoint. You can run tests immediately, without restarting your computer.

Our software displays information on how to update your system to have the endpoint start automatically upon restarting.

No changes are made to the `PATH` environment variable of the root user.

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:

```
rm *.q*.
```

3. Uninstall the current endpoint.
4. Install the desired endpoint.

Removing 64-Bit Linux Endpoints

Instructions for uninstalling 64-bit Linux endpoints is provided below, for both TAR-based packages and RPM-based packages.

Removing the TAR-Based Endpoint Package (Uninstall)

You must be logged in as root to remove the endpoint package. If you need to remove the endpoint package from your hard disk, first stop the endpoint program (if it is running) using the following command:

```
/usr/local/Ixia/endpoint -k
```

Then use the following command to remove the endpoint:

```
/usr/local/Ixia/endpoint.remove
```

If the removal is successful, you will see the following: “Removal of endpoint was successful.” This removes the files from `/usr/local/Ixia`, except for any files that were added to this directory that were not present at installation, such as the `endpoint.ini` file. This command does not delete

the directory. The remove program does not automatically delete files added to the directory that you may need if you reinstall the product.

If anything goes wrong during the process of uninstalling the endpoint, a reinstalled endpoint may not run. You may need to do some extra cleanup. Check for the hidden file `/var/local/Ixia/.IXIA.ENDPOINT.PID` by using the `ls -a` command. This file should be manually removed. Enter the following command:

```
rm /var/local/Ixia/.IXIA.ENDPOINT.PID
```

Removing the RPM-Based Endpoint Package (Uninstall)

You must be logged in as the root user to remove the endpoint package. If you need to remove the endpoint package from your hard disk, you must first stop the endpoint program (if it is running). To do so, enter the following command:

```
/usr/local/Ixia/endpoint -k
```

Use the following command to remove the endpoint:

```
rpm -e endpoint
```

If the removal is successful, you will see the following message: “Removal of endpoint was successful.” This removes the files from `/usr/local/Ixia`, except for any files that were added to this directory following the installation (such as the `endpoint.ini` file). The directory is not removed, nor does the remove program automatically delete files added to the directory that you may need if you reinstall the product.

Configuring 64-Bit Linux Endpoints

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. Take the following steps to verify that your network is ready for testing and/or monitoring:

- Determine the network addresses of the computers for use in tests.
- Verify the network connections.

The following topics explain how to accomplish these tasks for TCP/IP.

Configuration for TCP/IP

The 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`. The alternative, domain names are in a format that is easier to recognize and remember, such as `www.ixiacom.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 the IP address of the local computer you are using, enter the following at a command prompt:

```
/sbin/ifconfig
```

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies “port_number=AUTO” on the `CONNECT_ACCEPT` command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Testing the TCP Connection

Ping is a simple utility program, included in all TCP/IP implementations. To try out the connection from one computer to another, enter the following:

```
ping xx.xx.xx.xx -c 1
```

Replace the x’s with the IP address of the target computer. If Ping returns a message that says

```
1 packets transmitted, 1 packets received, 0% packet loss
```

the Ping worked. Otherwise, there will be a delay, and you’ll see

```
1 packets transmitted, 0 packets received, 100% packet loss
```

This means that the Ping failed, and you cannot reach the target computer.

Make sure that you can run Ping successfully from the IxChariot or Ixia Qcheck Console 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 64-Bit Linux Endpoints

The following topics describe how to manually start and stop the endpoint program, and how to examine error log files if a problem occurs.

Autostarting the Endpoint

For the endpoint to automatically start when your computer restarts, you must update your system `rc` scripts.

Use the following command:

```
cp /usr/local/Ixia/rc2exec.lnx /etc/rc.d/init.d/endpoint
/sbin/chkconfig endpoint reset
```


Starting a 64-Bit Linux Endpoint

The endpoint program is installed so that it starts automatically each time Linux is rebooted.

It sends its screen output to file `/var/local/endpoint.console`.

If you want to see any error messages generated at this endpoint, enter the following:

```
tail -f /var/local/endpoint.console
```

The detailed information about the start and stop of each individual connection pair is written to file `endpoint.aud`. The contents of this file vary depending on how you've set the `SECURITY_AUDITING` keyword in your `endpoint.ini` file.

See Chapter 3, *Endpoint Initialization File* for more information about `endpoint.aud` and `SECURITY_AUDIT` settings.

Instead of automatic startup, you can choose to manually start the endpoint program at a command prompt. Ensure that you are logged in as a “root” user. To start the endpoint, enter the following:

```
/usr/local/Ixia/endpoint &
```

The “&” parameter indicates to Linux that the endpoint program should run in the background. The screen output from the endpoint program is interleaved with other Linux commands. Just press RETURN to enter more commands.

If you choose to manually start the endpoint, consider redirecting its output to the `endpoint.console` file. You can tell by the time stamp of the file when the endpoint program was started or stopped.

If the endpoint program is already running, you get the following message, “**CHR0183**: The endpoint program is already running. Only one copy is allowed at a time.”

Use the `ps` command to check all running processes and make sure the endpoint is running. If you repeatedly get error message **CHR0183**, but it appears that the endpoint is not running, you may need to do some extra cleanup. Check for the hidden file `/usr/local/Ixia/.IXIA.ENDPOINT.PID` by using the `ls -a` command. This file should be manually removed.

Stopping a 64-Bit Linux Endpoint

The endpoint program has a special command-line option, `-k`. If you'd like to kill an endpoint program, go to a command prompt on the same computer and enter the following (you must be logged in as root to run this program):

```
/usr/local/Ixia/endpoint -k
```

The `-k` command-line option has the purpose of killing any endpoint process running on that computer. You should see the message “Sent exit request to the running endpoint,” which indicates that the endpoint program has been sent a request to stop.

If for some reason the request to stop is not handled correctly by the running endpoint program, you may need to use the Linux “`kill -TERM`” command. Avoid using “`kill -9`” to stop the running endpoint program—it doesn’t clean up what’s been created (so you’ll need to do the steps outlined in the following topics).

Cleanup after Unexpected Errors

If the endpoint should fail or be killed abnormally (or encounter assertion conditions), you may also need to do additional cleanup. If the endpoint is still running, try to stop it using the command “`endpoint -k`” (described above). If that does not stop the endpoint, kill the endpoint using the Linux `kill` command.

Then enter the following command:

```
rm /usr/local/Ixia/.IXIA.ENDPOINT.PID
```

How to Tell If a 64-Bit Linux Endpoint Is Active

Use traditional Linux commands to determine if a 64-bit Linux endpoint is active. At a command prompt, enter:

```
ps axf | grep endpoint
```

If the endpoint program is running, you will see output similar to this:

```
11118 pts/1 S 0:00 \_ grep endpoint
7652 pts/0 S 0:00 /usr/local/Ixia/endpoint
7653 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7654 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7655 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7656 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
```

Disabling Automatic Startup

Use the following command to disable the automatic startup:

```
/sbin/chkconfig --del endpoint
```

Increasing the Number of Concurrent Connections

Some parameters are tuned in Linux by rebuilding the Linux kernel. If you’re adventurous and skilled enough, you can change the number of concurrent endpoint connections. Consult your 64-bit Linux documentation for information about increasing the maximum open files allowed per process (this probably involves redefining `NR_FILES` and other macros). Alternatively, search Linux newsgroups on the Internet for something like “max open files per process.”

Logging and Messages

While most error messages encountered on an endpoint are returned to the IxChariot or Qcheck Console, some may be logged to disk. Errors are saved in the following file:

```
/var/log/endpoint.log
```

The log file is not created until an error occurs. To view an error log, use the program named `FMTLOG`. `FMTLOG` reads from a binary log file, and writes its formatted output to `stdout`. Use the following `FMTLOG` command:

```
/usr/local/Ixia/fmtlog /var/log/endpoint.log  
>output_filename
```

The endpoint code performs a good deal of internal checking. Our software captures details related to the problem in an ASCII text file:

```
/var/local/assert.err.
```

Save a copy of the file and send it to us via email for problem determination.

Message CHR0181

You may receive message CHR0181 while running a test. If the error was detected at the Linux computer, it says that the endpoint program on Linux has run out of system semaphores. Each instance of Endpoint 1 requires a system semaphore. The maximum number of semaphores cannot be configured on Linux, which is hard-coded to a large value (128). To avoid this problem, stop other programs that use semaphores or decrease the number of tests that use the computer as Endpoint 1.

9

Linux on ARM Processors

The following topics explain the installation, configuration, and operation of the Performance Endpoint software for 32-bit Linux running on an ARM platform. ARM Linux is a port of the Linux Kernel to ARM processor based machines. The ARM Linux kernel has been ported to a wide range of systems, including network devices, hand held devices, and embedded devices. This chapter includes the following topics:

- [*Linux on ARM Performance Endpoints*](#) on page 9-1
- [*Installing the Linux 32-bit on ARM Endpoint*](#) on page 9-2
- [*Configuring the Linux 32-Bit on ARM Endpoint*](#) on page 9-3
- [*Running the Linux 32-Bit on ARM Endpoint*](#) on page 9-5

Other chapters in this manual describe additional Linux endpoints.

Linux on ARM Performance Endpoints

Ixia provides four distinct Performance Endpoints for Linux running on ARM processors:

- **Linux / ARM (Little Endian)** – Performance Endpoint for Linux running on little endian ARM platforms.
- **Linux / ARM (Big Endian)** – Performance Endpoint for Linux running on little endian ARM platforms.
- **Linux / ARM (Statically-Linked Little Endian)** – Performance Endpoint for Linux running on little endian ARM platforms. This Performance Endpoint includes a statically-linked link library.
- **Linux / ARM (uClibc)** – Performance Endpoint for Linux running on little endian ARM platforms. This Performance Endpoint is compiled with uClibc, a small C standard library designed for embedded Linux systems.

About Endianness

Ixia provides big endian and little endian Performance Endpoints for 32-bit Linux systems running on the ARM platform.

Endianness refers to the byte order used by a computer when it stores a value in memory. Big endian architectures store the most significant byte in a memory location with the lowest address, while little endian architectures store the most significant byte in a memory location with the highest address. Big endian architectures include Motorola 68000, SPARC, and System/370. Little endian architectures include the MOS Technology 6502 and Intel x86.

Some architectures can be configured either way. These include ARM, PowerPC (excluding the PPC970/G5), MIPS, among others. The endpoint that you will use is based on the endianness of the ARM-based device that you are using.

Installing the Linux 32-bit on ARM Endpoint

Requirements

Here is what you need to run the Linux 32-bit on ARM endpoint program:

- A device with an ARM-compatible CPU.
- 800 KB of free RAM. (This RAM requirement is for a minimal test only. As more pairs are added to a test, more memory is required.)
- 2.2 MB of permanent storage.
- A Linux operating system (Linux kernel 2.4.20), such as Monta Vista Linux.
- Glibc 2.3.3 or newer (except for the Linux / ARM uClibc Performance Endpoint).

We have tested with packages that implement Linux kernel 2.4.20 and 2.6.9. We have not tested this version of the Performance Endpoint with any earlier versions of the Linux kernel.

No Log Files are Created

In order to conserve RAM, the Performance Endpoint for Linux 32-bit on ARM does not generate the `endpoint.log` and `assert.err` files. All error messages are sent to the standard output device.

TAR-Based Installation for Linux 32-Bit on ARM Endpoints

All commands and parameters discussed here are case-sensitive. Use the combination of uppercase and lowercase letters as shown in the text. You can install from the IxChariot CD-ROM or download the Performance Endpoint from the Ixia web site. The commands that follow assume you obtained the file from the Ixia web site.

This manual uses *Mm* to represent the product release number in Performance Endpoint file names; for example, `pelinux_arm_Mm.tar`. The *M* specifies the major release while the *m* specifies the minor release. For example, the little endian Performance Endpoint for Release 6.50 is named `pelinux_arm_650.tar`.

To decompress the Performance Endpoint:

1. Ensure that you are logged in as root.
2. `cd` to the directory where you will extract the archive contents.
3. Extract the archive contents. For example:

```
tar -xvf pelinux_arm_Mm.tar
```

The procedure for installing these files on an ARM-based device is device-specific. When you have completed the installation, your endpoint should be ready to be used in testing and monitoring.

What We Do During Installation

Here is what happens during the installation steps. The endpoint is installed in a customer-chosen directory. The following contents are placed in that directory:

- the Performance Endpoint executable.
- the README file.
- the Ixia EULA (End-User License Agreement).
- 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 `endpoint.ini` file. See Chapter 3, [Endpoint Initialization File](#) for information about tailoring this file for individual endpoints.
- a file that holds the text messages that will be displayed by the Performance Endpoint and by the IxChariot Console.

Uninstalling

To uninstall the endpoint, it is sufficient to delete the endpoint directory and all of its contents.

Configuring the Linux 32-Bit on ARM Endpoint

The endpoint dynamically configures its own programs, so you do not need to update the configuration files for your communications software. However, your communications software must be configured and running correctly. Take the following steps to verify that your network is ready for testing and/or monitoring:

- Determine the network addresses of the devices for use in tests.
- Verify the network connections.

The following topics explain how to accomplish these tasks for TCP/IP.

Supported Protocols

The Performance Endpoint for Linux 32-bit on ARM supports IPv4 over TCP, UDP, and RTP. It does not support IPv6, IPX, SPX, or APPC.

Configuration for TCP/IP

The 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 IPv4 address is a 32-bit numeric address. It is represented in dotted notation as a set of four numbers separated by periods, such as 192.168.46.202. The alternative—domain names—are in a format that is easier to recognize and remember, such as www.ixiacom.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 device.

Determining Your IP Network Address

To determine the IP address of the local device you are using, enter the following at a command prompt:

```
ifconfig
```

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies “port_number=AUTO” on the `CONNECT_ACCEPT` command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Testing the TCP Connection

Ping is a simple utility program included in all TCP/IP implementations. To verify the connection from one device to another, enter the following:

```
ping xx.xx.xx.xx -c 1
```

Replace `xx.xx.xx.xx` with the IP address of the target device. You will know that you can reach the target host if Ping returns this message:

```
1 packets transmitted, 1 packets received, 0% packet loss
```

If Ping fails to reach the target host, it returns this message:

```
1 packets transmitted, 0 packets received, 100% packet loss
```

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

Running the Linux 32-Bit on ARM Endpoint

The following topics describe how to manually start and stop the endpoint program.

Starting the Linux 32-Bit on ARM Endpoint

Use the following steps to manually start the endpoint program at a command prompt:

1. Ensure that you are logged in as root.
2. To start the endpoint, change to the directory in which you installed the endpoint, then enter the following command:

```
./endpoint &
```

The optional “&” parameter indicates to Linux that the endpoint program should run in the background. When running in the foreground, the screen output from the endpoint program is interleaved with other Linux commands. Just press RETURN to enter more commands.

If you choose to manually start the endpoint, consider redirecting its output to the `endpoint.console` file. For example:

```
./endpoint > endpoint.console
```

You can tell by the time stamp of the file when the endpoint program was started or stopped.

If the endpoint program is already running, you get the following message, “**CHR0183**: The endpoint program is already running. Only one copy is allowed at a time.”

Use the `ps` command to check all running processes and make sure the endpoint is running (see [How to Tell if the Linux 32-Bit on ARM Endpoint is Active](#) on page 9-6). If you repeatedly get error message **CHR0183**, but it appears that the endpoint is not running, you may need to do some extra cleanup. Check for the hidden file `/var/log/.ENDPOINT.PID` by using the `ls -a` command. This file should be manually removed.

Stopping the Linux 32-Bit on ARM Endpoint

The endpoint program has a special command-line option, `-k`. If you’d like to kill an endpoint program, go to a command prompt on the same device and enter the following (you must be logged in as root to run this program):

```
./endpoint -k
```

The `-k` command-line option has the purpose of killing any endpoint process running on that device. You should see the message “Sent exit request to the running endpoint,” which indicates that the endpoint program has been sent a request to stop.

If for some reason the request to stop is not handled correctly by the running endpoint program, you may need to use the Linux “`kill -TERM`” command. Avoid using “`kill -9`” to stop the running endpoint program—it doesn’t clean up what’s been created (so you’ll need to do the steps outlined in the following topics).

Clean-up After Unexpected Errors

If the endpoint should fail or be killed abnormally (or encounter assertion conditions), you may also need to do additional cleanup. If the endpoint is still running, try to stop it using the command “`endpoint -k`” (described above). If that does not stop the endpoint, kill the endpoint using the Linux `kill` command.

Then enter the following command:

```
rm /var/log/.ENDPOINT.PID
```

How to Tell if the Linux 32-Bit on ARM Endpoint is Active

Use traditional Linux commands to determine if a Linux endpoint is active. For example:

```
ps axf | grep endpoint
```

If the endpoint program is running, you will see output similar to this:

```
11118 pts/1 S 0:00 \_ grep endpoint
7652 pts/0 S 0:00 /usr/local/Ixia/endpoint
7653 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7654 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7655 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7656 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
```

The sample output listed above indicates that the endpoint was installed in `/usr/local/Ixia`. However, given that the endpoint does not provide an installer, you install the endpoint in a directory of your own choosing.

10

Linux on Lexra

The following topics explain the installation, configuration, and operation of the Performance Endpoint software for 32-bit Linux running on a Lexra platform. (Ixia customers have successfully used this Performance Endpoint on MIPS platforms (big endian) as well as the Lexra platform.)

- [Installing the Linux 32-bit on Lexra Endpoint](#) on page 10-1
- [Configuring the Linux 32-bit on Lexra Endpoint](#) on page 10-2
- [Running the Linux 32-bit on Lexra Endpoint](#) on page 10-4

Other chapters in this manual describe additional Linux endpoints.

Installing the Linux 32-bit on Lexra Endpoint

Requirements

Here is what you need to run the Linux 32-bit on Lexra endpoint program:

- A device with a Lexra-compatible CPU. We tested with Lexra LX5280.
- 1.7 MBytes of flash memory available.
- 4 MBytes of free RAM.
- Linux kernel 2.4.18.

We have tested with packages that implement Linux kernel 2.4.18. We have not tested this version of the Performance Endpoint with any earlier versions of the Linux kernel.

No log files are created

In order to conserve RAM, the Performance Endpoint for Linux 32-bit on Lexra does not generate the endpoint.log and assert.err files. All error messages are sent to the standard output device.

TAR-Based Installation for Linux 32-bit on Lexra Endpoints

All commands and parameters discussed here are case-sensitive. Use the combination of uppercase and lowercase letters as shown in the text. You can install from the IxChariot CD-ROM or download the Performance Endpoint from the Ixia web site.

The following commands assume you obtained the file from the Ixia web site. The Performance Endpoint file is named `pelex_Mm.tar.gz`, where “M” represents the major version and “m” represents the minor version. For example, the Performance Endpoint for Release 6.10 is `pelex_610.tar.gz`.

To decompress the Performance Endpoint:

1. Ensure that you are logged in as root.
2. `cd` to the directory where you will extract the archive contents.
3. Extract the archive contents:

```
gzip -d pelex_Mm.tar.gz
tar -xvf pelex_Mm.tar
```

The procedure for installing these files on the Lexra-based device is device-specific. When you have completed the installation, your endpoint should be ready to be used in testing and monitoring.

What We Do During Installation

Here is what happens during the installation steps. The endpoint is installed in a customer-chosen directory. The following contents are placed in that directory:

- the Performance Endpoint executable.
- the README file.
- the Ixia End-User License Agreement.
- 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 Chapter 3, [Endpoint Initialization File](#) for information about tailoring this file for individual endpoints.
- a file that holds the text messages that will be displayed by the Performance Endpoint and by the IxChariot Console.

Configuring the Linux 32-bit on Lexra Endpoint

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. Take the following steps to verify that your network is ready for testing and/or monitoring:

- Determine the network addresses of the computers for use in tests.
- Verify the network connections.

The following topics explain how to accomplish these tasks for TCP/IP.

Configuration for TCP/IP

The 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 IPv4 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. The alternative, domain names are in a format that is easier to recognize and remember, such as www.ixiacom.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.

Note: The Performance Endpoint for Linux 32-bit on Lexra supports TCP and UDP only. It does not support IPv6, IPX, SPX, or other network protocols.

Determining Your IP Network Address

To determine the IP address of the local computer you are using, enter the following at a command prompt:

```
ifconfig
```

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies “port_number=AUTO” on the `CONNECT_ACCEPT` command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Testing the TCP Connection

Ping is a simple utility program, included in all TCP/IP implementations. To try out the connection from one computer to another, enter the following:

```
ping xx.xx.xx.xx -c 1
```

Replace the x’s with the IP address of the target computer. If Ping returns a message that says

```
1 packets transmitted, 1 packets received, 0% packet loss
```

the Ping worked. Otherwise, there will be a delay, and you’ll see

```
1 packets transmitted, 0 packets received, 100% packet loss
```

This means that the Ping failed, and you cannot reach the target computer.

Make sure that you can run Ping successfully from the IxChariot or Ixia Qcheck Console 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 the Linux 32-bit on Lexra Endpoint

The following topics describe how to manually start and stop the endpoint program.

Starting the Linux 32-bit on Lexra Endpoint

Use the following steps to manually start the endpoint program at a command prompt:

1. Ensure that you are logged in as root.
2. To start the endpoint, change to the directory in which you installed the endpoint, then enter the following command:

```
./endpoint &
```

The “&” parameter indicates to Linux that the endpoint program should run in the background. The screen output from the endpoint program is interleaved with other Linux commands. Just press RETURN to enter more commands.

If you choose to manually start the endpoint, consider redirecting its output to the `endpoint.console` file. You can tell by the time stamp of the file when the endpoint program was started or stopped.

If the endpoint program is already running, you get the following message, “**CHR0183**: The endpoint program is already running. Only one copy is allowed at a time.”

Use the `ps` command to check all running processes and make sure the endpoint is running. If you repeatedly get error message **CHR0183**, but it appears that the endpoint is not running, you may need to do some extra cleanup. Check for the hidden file `/var/log/.ENDPOINT.PID` by using the `ls -a` command. This file should be manually removed.

Stopping the Linux 32-bit on Lexra Endpoint

The endpoint program has a special command-line option, `-k`. If you’d like to kill an endpoint program, go to a command prompt on the same computer and enter the following (you must be logged in as root to run this program):

```
./endpoint -k
```

The `-k` command-line option has the purpose of killing any endpoint process running on that computer. You should see the message “Sent exit request to the running endpoint,” which indicates that the endpoint program has been sent a request to stop.

If for some reason the request to stop is not handled correctly by the running endpoint program, you may need to use the Linux “`kill -TERM`” command. Avoid using “`kill -9`” to stop the running endpoint program—it doesn’t clean up what’s been created (so you’ll need to do the steps outlined in the following topics).

Cleanup after Unexpected Errors

If the endpoint should fail or be killed abnormally (or encounter assertion conditions), you may also need to do additional cleanup. If the endpoint is still running, try to stop it using the command “`endpoint -k`” (described above). If that does not stop the endpoint, kill the endpoint using the Linux `kill` command.

Then enter the following command:

```
rm /var/log/.ENDPOINT.PID
```

How to Tell If the Linux 32-bit on Lexra Endpoint Is Active

Use traditional Linux commands to determine if a Linux endpoint is active. For example:

```
ps axf | grep endpoint
```

If the endpoint program is running, you will see output similar to this:

```
11118 pts/1 S 0:00 \_ grep endpoint
7652 pts/0 S 0:00 /usr/local/Ixia/endpoint
7653 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7654 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7655 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7656 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
```

The sample output listed above indicates that the endpoint was installed in `/usr/local/Ixia`. However, given that the endpoint does not provide an installer, you install the endpoint in a directory of your own choosing.

11

Linux on OpenWrt (MIPS Platforms)

This chapter explains the installation and operation of the Performance Endpoint software for Linux 32-bit on OpenWrt running on a MIPS platform.

OpenWrt is a 32-bit Linux distribution for embedded devices. It provides a fully writable file system with package management.

This chapter includes the following topics:

- [Performance Endpoint Overview](#) on page 11-1
- [Installing the Linux 32-bit on OpenWrt Endpoint](#) on page 11-2
- [Removing the Linux 32-bit on OpenWrt Endpoint](#) on page 11-3
- [TCP/IP Sockets Interface Support](#) on page 11-3
- [Running the Linux 32-bit on OpenWrt Endpoint](#) on page 11-4

Other chapters in this manual describe additional Linux endpoints.

Performance Endpoint Overview

File Names

Ixia provides two versions of the Linux 32-bit on OpenWrt Performance Endpoint:

- pelinux_mipsle_*Mm*.ipk – IPKG distribution
- pelinux_mipsle_*Mm*.tar – tar distribution

where “M” represents the major version and “m” represents the minor version. For example, pelinux_mipsle_640.ipk is version 6.40 of the IPKG distribution Performance Endpoint.

Linux kernel 2.4.30

Ixia has tested the Performance Endpoint on a Linksys WRT54GL device running the WhiteRussian RC5 version of OpenWrt, which is based on Linux kernel 2.4.30.

Little Endian

This is a Little Endian performance endpoint. (Big endian architectures store the most significant byte in a memory location with the lowest address, while little endian architectures store the most significant byte in a memory location with the highest address.)

Installing the Linux 32-bit on OpenWrt Endpoint

Requirements

Here is what you need to install and run the Linux 32-bit on OpenWrt Performance Endpoint:

- A device running OpenWrt with a MIPS Little Endian architecture.
- 2 MB of flash memory available.
- 4 MB of free RAM.

No Log Files are Created

In order to conserve RAM, the Performance Endpoint for Linux 32-bit on OpenWrt does not generate the `endpoint.log` and `assert.err` files. All errors are sent to the standard output device.

Linux 32-Bit on OpenWrt Endpoints Installation Instructions

All commands and parameters discussed here are case-sensitive. Use the combination of uppercase and lowercase letters as shown in the text. You can install from the IxChariot CD-ROM or download the Performance Endpoint from the Ixia web site. The following commands assume you obtained the file from the Ixia web site.

TAR-Based Installation

To decompress the Performance Endpoint:

1. Ensure that you are logged in as root.
2. Extract the archive contents:

```
tar -xvf pelinux_mipsle_Mm.tar
```

IPKG-Based Installation

To install the IPKG-based Performance Endpoint:

1. Ensure that you are logged in as root.
2. Execute the following command:

```
ipkg install pelinux_mipsle_Mm.ipk
```

What We Do During Installation

When you execute the `tar` or the `ipkg` command, the following Performance Endpoint files are placed in the `temp` directory:

- the Performance Endpoint executable.
- the README file.

- the Ixia EULA (End-User License Agreement).
- 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 `endpoint.ini` file. See Chapter 3, [Endpoint Initialization File](#) for information about tailoring this file for individual endpoints.
- a file that holds the text messages that will be displayed by the Performance Endpoint and by the IxChariot Console.

Removing the Linux 32-bit on OpenWrt Endpoint

The procedures for uninstalling the Performance Endpoint are specific to the type of installation you performed: tar-based or ipkg-based.

Removing TAR-Based Installations

To remove the tar-based Performance Endpoint (`pelinux_mipsle_Mm.tar`):

1. Ensure that you are logged in as root.
2. Delete all the files that were installed in the temp directory.

Removing IPKG-Based Installations

To remove the IPKG-based Performance Endpoint (`pelinux_mipsle_Mm.ipk`):

1. Ensure that you are logged in as root.
2. Execute the following command:

```
ipkg remove endpoint
```

TCP/IP Sockets Interface Support

The endpoint dynamically configures its own programs, so you do not need to update the configuration files for your communications software. However, your communications software must be configured and running correctly. For example, to use domain names, you need either a Domain Name Server (DNS) set up in your network or an `/etc/hosts` file on each device.

Determining Your IP Network Address

To determine the IP address of the local device you are using, enter the following at a command prompt:

```
ifconfig
```

Supported Protocols

The Performance Endpoint for Linux 32-bit on OpenWrt uses the Sockets interface to the TCP/IP support shipped with Linux. It supports the following protocols:

- IPv4 over TCP
- UDP
- RTP

Neither APPC, IPX, SPX, nor any other network protocols are supported in this version.

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies “port_number=AUTO” on the `CONNECT_ACCEPT` command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Running the Linux 32-bit on OpenWrt Endpoint

The following topics describe how to manually start and stop the endpoint program.

Starting the Linux 32-bit on OpenWrt Endpoint

Use the following steps to manually start the endpoint program at a command prompt:

1. Ensure that you are logged in as root.
2. To start the endpoint, change to the directory in which the Performance Endpoint executable resides, then enter the following command:

```
./endpoint &
```

The optional “&” parameter indicates to Linux that the endpoint program should run in the background. When running in the foreground, the screen output from the endpoint program is interleaved with other Linux commands. Just press RETURN to enter more commands.

If you choose to manually start the endpoint, consider redirecting its output to the `endpoint.console` file. For example:

```
./endpoint > endpoint.console
```

You can tell by the time stamp of the file when the endpoint program was started or stopped.

If the endpoint program is already running, you get the following message, “**CHR0183**: The endpoint program is already running. Only one copy is allowed at a time.”

Use the `ps` command to check all running processes and make sure the endpoint is running (see [How to Tell if the Linux 32-bit on OpenWrt Endpoint is Active](#) on page 11-5). If you repeatedly get error message **CHR0183**, but it appears that the endpoint is not running, you may need to do some extra cleanup. Check for the hidden file `/var/log/.ENDPOINT.PID` by using the `ls -a` command. This file should be manually removed.

Stopping the Linux 32-bit on OpenWrt Endpoint

The endpoint program has a special command-line option, `-k`. If you’d like to kill an endpoint program, go to a command prompt on the same device and enter the following (you must be logged in as root to run this program):

```
./endpoint -k
```

The `-k` command line option has the purpose of killing any endpoint process running on that device. You should see the message “Sent exit request to the running endpoint,” which indicates that the endpoint program has been sent a request to stop.

If for some reason the request to stop is not handled correctly by the running endpoint program, you may need to use the Linux “`kill -TERM`” command. Avoid using “`kill -9`” to stop the running endpoint program—it doesn’t clean up what’s been created (so you’ll need to do the steps outlined in the following topics).

Clean-up After Unexpected Errors

If the endpoint should fail or be killed abnormally (or encounter assertion conditions), you may also need to do additional cleanup. If the endpoint is still running, try to stop it using the command “`endpoint -k`” (described above). If that does not stop the endpoint, kill the endpoint using the Linux `kill` command.

Then enter the following command:

```
rm /var/log/.ENDPOINT.PID
```

How to Tell if the Linux 32-bit on OpenWrt Endpoint is Active

Use traditional Linux commands to determine if a Linux endpoint is active. For example:

```
ps axf | grep endpoint
```

If the endpoint program is running, you will see output similar to this:

```
11118 pts/1 S 0:00 \_ grep endpoint
7652 pts/0 S 0:00 /usr/local/Ixia/endpoint
```

```
7653 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7654 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7655 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7656 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
```

12

Linux on PowerPC

The following topics explain the installation, configuration, and operation of the Performance Endpoint software for 32-bit Linux running on the PowerPC platform. The Performance Endpoint is designed for Linux implementations that are optimized for PowerPC-compatible embedded systems. This chapter includes the following topics:

- [Installing the Linux on PowerPC Performance Endpoint](#) on page 12-1
- [Configuring the Linux on PowerPC Endpoint](#) on page 12-3
- [Running the Linux on PowerPC Endpoint](#) on page 12-4

Other chapters in this manual describe additional Linux endpoints.

Installing the Linux on PowerPC Performance Endpoint

Requirements

Here is what you need to run the Linux on PowerPC Performance Endpoint program:

- A device with a PowerPC 405-compatible CPU.
- 4 MBytes of free RAM.
- A Linux operating system that implements Linux kernel 2.4.19 (or higher).

Note: Linux kernel 2.6.18 is required for IPTV testing.

We have tested with packages that implement Linux kernel 2.4.19. We have not tested this version of the Performance Endpoint with any earlier versions of the Linux kernel.

No Log Files are Created

In order to conserve RAM, the Performance Endpoint for Linux on PowerPC does not generate the endpoint.log and assert.err files. All error messages are sent to the standard output device.

TAR-Based Installation for Linux on PowerPC Endpoints

All commands and parameters discussed here are case-sensitive. Use the combination of uppercase and lowercase letters as shown in the text. You can install from the IxChariot CD-ROM or download the Performance Endpoint from the Ixia web site.

Extracting the Archive Contents

The following commands assume that you obtained the file from the Ixia web site. The Performance Endpoint file is named `pelinux_ppc_amcc_Mm.tar`, where “M” represents the major version and “m” represents the minor version. For example, the Performance Endpoint for Release 6.50 is named `pelinux_ppc_amcc_650.tar`.

To decompress the Performance Endpoint:

1. Ensure that you are logged in as root.
2. `cd` to the directory where you will extract the archive contents.
3. Extract the archive contents:

```
tar -xvf pelinux_ppc_amcc_Mm.tar
```

The files are extracted to a `temp` directory. See [Contents of the tar file](#) on page 12-2 for a description of the files that are extracted from the tar file.

Installing the Performance Endpoint

The procedure for installing these files on a PowerPC-based device is device-specific. Once you have completed the installation for your device, the endpoint will be ready for use in testing and monitoring.

Note that the endpoint is installed in RAM. Therefore, rebooting the system deletes the endpoint and all related files.

Contents of the tar file

When you decompress the Performance Endpoint (as described in [TAR-Based Installation for Linux on PowerPC Endpoints](#) on page 12-2), the following files are placed in a `temp` directory:

- the Performance Endpoint executable.
- the README file.
- the Ixia EULA (End-User License Agreement).
- 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 `endpoint.ini` file. See Chapter 3, [Endpoint Initialization File](#) for information about tailoring this file for individual endpoints.
- a file that holds the text messages that will be displayed by the Performance Endpoint and by the IxChariot Console.

Uninstalling

Because the files are installed to RAM only, there is no uninstall operation required. Rebooting the system deletes the endpoint and all related files.

Configuring the Linux on PowerPC Endpoint

The endpoint dynamically configures its own programs, so you do not need to update the configuration files for your communications software. However, your communications software must be configured and running correctly. Take the following steps to verify that your network is ready for testing and/or monitoring:

- Determine the network addresses of the devices that will be used in tests.
- Verify the network connections.

The following topics explain how to accomplish these tasks for TCP/IP.

Supported Protocols

The Performance Endpoint for Linux on PowerPC supports IPv4 over TCP, UDP, and RTP. It does not support IPv6, IPX, SPX, or APPC.

To use domain names, you need either a Domain Name Server (DNS) set up in your network or an `/etc/hosts` file on each device.

Determining Your IP Network Address

To determine the IP address of the local device you are using, enter the following at a command prompt:

```
ifconfig
```

Sockets Port Number

TCP/IP applications use Sockets port numbers to determine to which application program to connect within a device.

The TCP/IP sockets port used by IxChariot endpoints is 10115. This port number is used during the initialization of a test. During the actual running of the test, other port numbers are used. If the script specifies “port_number=AUTO” on the `CONNECT_ACCEPT` command, additional ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Testing the TCP Connection

Ping is a simple utility program included in all TCP/IP implementations. To verify the connection from one device to another, enter the following:

```
ping xx.xx.xx.xx -c 1
```

Replace `xx.xx.xx.xx` with the IP address of the target device. You will know that you can reach the target host if Ping returns this message:

```
1 packets transmitted, 1 packets received, 0% packet loss
```

If Ping fails to reach the target host, it returns this message:

```
1 packets transmitted, 0 packets received, 100% packet loss
```

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

Running the Linux on PowerPC Endpoint

The following topics describe how to manually start and stop the endpoint program.

Starting the Linux on PowerPC Endpoint

Use the following steps to manually start the endpoint program at a command prompt:

1. Ensure that you are logged in as root.
2. To start the endpoint, change to the directory in which you installed the endpoint, then enter the following command:

```
./endpoint &
```

The optional “&” parameter indicates to Linux that the endpoint program should run in the background.

If the endpoint program is already running, you get the following message, “**CHR0183**: The endpoint program is already running. Only one copy is allowed at a time.”

Use the `ps` command to check all running processes and make sure the endpoint is running (see [How to Tell if the Linux on PowerPC Endpoint is Active](#) on page 12-5). If you repeatedly get error message **CHR0183**, but it appears that the endpoint is not running, you may need to do some extra cleanup. Check for the hidden file `/var/log/.ENDPOINT.PID` by using the `ls -a` command. This file should be manually removed.

Stopping the Linux on PowerPC Endpoint

The endpoint program has a special command-line option, `-k`. If you’d like to kill an endpoint program, go to a command prompt on the same device and enter the following (you must be logged in as root to run this program):

```
./endpoint -k
```

The `-k` command-line option has the purpose of killing any endpoint process running on that device. You should see the message “Sent exit request to the running endpoint,” which indicates that the endpoint program has been sent a request to stop.

If for some reason the request to stop is not handled correctly by the running endpoint program, you may need to use the Linux “`kill -TERM`” command. Avoid using “`kill -9`” to stop the running endpoint program—it doesn’t clean up what’s been created (so you’ll need to do the steps outlined in the following topics).

Cleanup After Unexpected Errors

If the endpoint should fail or be killed abnormally (or encounter assertion conditions), you may also need to do additional cleanup. If the endpoint is still running, try to stop it using the command “`endpoint -k`” (described above). If that does not stop the endpoint, kill the endpoint using the Linux `kill` command.

Then enter the following command:

```
rm /var/log/.ENDPOINT.PID
```

How to Tell if the Linux on PowerPC Endpoint is Active

Use traditional Linux commands to determine if a Linux endpoint is active. For example:

```
ps axf | grep endpoint
```

If the endpoint program is running, you will see output similar to this:

```
11118 pts/1 S 0:00 \_ grep endpoint
7652 pts/0 S 0:00 /usr/local/Ixia/endpoint
7653 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7654 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7655 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
7656 pts/0 S 0:00 \_ /usr/local/Ixia/endpoint
```

The sample output listed above indicates that the endpoint was installed in `/usr/local/Ixia`. However, given that the endpoint does not provide an installer, you install the endpoint in a directory of your own choosing.

13

Mac OS X

This chapter explains the installation, configuration, and operation of the Performance Endpoint software for Mac OS X.

Topics in this chapter:

- *Platforms Supported* on page 13-1
- *Installing the Mac OS Performance Endpoint* on page 13-1
- *Configuring Mac OS X Endpoints* on page 13-3
- *Running Mac OS X Endpoints* on page 13-4
- *Logging and Messages* on page 13-5
- *Updates for Mac OS X* on page 13-6

Platforms Supported

The Mac OS X Performance Endpoint is a 32-bit program that runs on the following Mac OS X platforms:

- PowerPC G4 processor (32-bit) systems, such as the eMac.
- PowerPC G5 processor (64-bit) systems, such as the Power Macintosh G5.

Installing the Mac OS Performance Endpoint

Here is what you need to run the endpoint program with Mac OS X:

- An Apple computer capable of running Mac OS X 10.3.
- 128 MBytes of random access memory (RAM).
- The total RAM requirement depends on 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 10 MBytes of space available

Installation Procedure

First, ensure that you are logged in as a user with administrative privileges.

Next, find the Mac OS X endpoint from our web site's endpoint library at: http://www.ixiacom.com/support/endpoint_library/ and double-click on the endpoint file (for example *pemac_640.dmg*). The endpoint will be downloaded and the installation started. You should follow the instructions to complete the installation. During the installation, you will be offered the opportunity to view the `README` file, which contains the latest information about the endpoint program.

The endpoint is installed in your *Applications* folder as a MAC application. To start the endpoint, browse the application folder and double click on the endpoint icon. The `README` file contains instructions on how to install the endpoint as a service.

When you've completed installation, refer to [Configuring Mac OS X Endpoints](#) on page 13-3 to make sure your endpoint is ready to be used in testing and monitoring.

Removing the Endpoint (Uninstall)

Using Finder, delete the Endpoint bundle.

What Happens During Installation

Here is what happens during the installation steps. The endpoint is installed into the *Applications* folder. A directory is created with the following contents:

- The executable programs
- The `README` file
- Various install and uninstall programs
- 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 Chapter 3, [Endpoint Initialization File](#) for information about tailoring this file for individual endpoints.

If an earlier version of the endpoint is installed, you will be asked if you wish to upgrade. If you agree, the installation program stops any copy of the endpoint program currently running and starts a copy of the newly installed endpoint. You can run tests immediately, without restarting your computer.

Downgrading to an older version of the Endpoint

To downgrade to an older version of the endpoint:

1. Follow the steps to uninstall the Endpoint (refer to [Removing the Endpoint \(Uninstall\)](#) on page 13-2).
2. Manually delete this directory: `/Library/Receipts/pemac.pkg`.

3. Then follow the installation instructions for the older Endpoint.

Configuring Mac OS X Endpoints

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. Take the following steps to verify that your network is ready for testing and/or monitoring:

1. Determine the network addresses of the computers for use in tests.
2. Verify the network connections.

Let's look at TCP/IP to see how to accomplish these tasks.

Configuration for TCP/IP

The 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 192.168.46.202. The alternative, domain names are in a format that is easier to recognize and remember, such as www.ixiacom.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 the IP address of the local computer you are using, enter the following in a Terminal window:

```
/sbin/ifconfig
```

Testing the TCP Connection

Ping is a simple utility program, included in all TCP/IP implementations. To try out the connection from one computer to another, enter the following:

```
ping xx.xx.xx.xx -c 1
```

Replace the x's with the IP address of the target computer. If Ping returns a message that says

```
1 packets transmitted, 1 packets received, 0% packet loss
```

then the Ping worked. Otherwise, there will be a delay, and you'll see

```
1 packets transmitted, 0 packets received, 100% packet loss
```

This means that the Ping failed, and you cannot reach the target computer.

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

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies “port_number=AUTO” on the `CONNECT_ACCEPT` command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Running Mac OS X Endpoints

The following sections describe how to manually start and stop the endpoint program, and how to examine error log files if a problem occurs.

The endpoint icon, located in the *Applications* folder, may be used to manually start the Mac OS X endpoint. Alternatively, you may set up the endpoint to automatically start with your computer by dragging and dropping the icon into the `/library/StartupItems` folder using Finder.

If you want to see any error messages generated by the endpoint, use the `fmtlog` command to view the *Endpoint.log* file located in `/private/var/log`.

The detailed information about the start and stop of each individual connection pair is written to file `endpoint.aud`. The contents of this file vary depending on how you’ve set the `SECURITY_AUDITING` keyword in your `endpoint.ini` file.

See Chapter 3, [Endpoint Initialization File](#) for more information about `endpoint.aud` and `SECURITY_AUDIT` settings.

If the endpoint program is already running, you get the following message, “**CHR0183**: The endpoint program is already running. Only one copy is allowed at a time.”

Use the `ps` command to check all running processes and make sure the endpoint is running (see the section, [How to Tell If a Mac OS X Endpoint Is Active](#) on page 13-5 for more information). If you repeatedly get error message **CHR0183** but it appears that the endpoint is not running, you may need to do some extra cleanup. Check for the file `/private/var/log/.ENDPOINT.PID` by using Finder. This file should be manually removed.

Stopping a Mac OS X Endpoint

If the endpoint was started manually, it may be terminated by selecting **Quit** from the desktop icon.

If the endpoint was started automatically, then it may be terminated by using the `SystemStarter` command:

```
sudo SystemStarter Stop Endpoint
```

A password may be required.

If the endpoint does not stop, then you will need to use

```
kill -9 <pid>
```

to stop the running endpoint program. See [How to Tell If a Mac OS X Endpoint Is Active](#) below for instructions on using the `ps` command and determining the process id (pid) of the endpoint. With the “-9” argument, the endpoint doesn’t clean up what’s been created (so you’ll need to do the steps outlined in [Cleanup after Unexpected Errors](#) on page 13-5).

Cleanup after Unexpected Errors

If the endpoint should fail or be killed abnormally (or encounter assertion conditions), you may also need to do additional cleanup. Enter the following command:

```
rm -f /usr/local/ixia/IXIA.ENDPOINT.PID
```

How to Tell If a Mac OS X Endpoint Is Active

Use traditional UNIX commands to determine if a Mac OS X endpoint is active. At a command prompt, enter:

```
ps ax | grep endpoint
```

If the endpoint program is running, you will see output similar to this:

```
855 ?? S 3:19:90 ./endpoint
2846 std R+ 0:00:00 grep endpoint
```

Disabling Automatic Startup

If you wish to disable the Mac OS X from running as a service, then stop it as described above and remove the endpoint folder from the `/Library/StartupItems` folder.

Logging and Messages

While most error messages encountered on an endpoint are returned to the IxChariot or Qcheck Console, some may be logged to disk. Errors are saved in the following file:

- `/private/var/log/endpoint.log`

To view an error log, use the IxChariot Console’s Tool menu, View Error Log choice.

The endpoint code does a lot of internal checking on itself. Our software captures details related to the problem in an ASCII text file:

- `/private/var/log/assert.err`

Save a copy of the file and send it to us via email for problem determination.

Updates for Mac OS X

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.

Use the Software Update program that is included with Mac OS to keep your Mac software up to date.

14

Microsoft Windows 32-Bit

This chapter explains the installation, configuration, and operation of the Performance Endpoint software for 32-bit Windows operating systems.

Topics in this chapter:

- [Windows Operating Systems Supported](#) on page 14-1
- [Installation Requirements for the 32-bit Windows Endpoint](#) on page 14-2
- [Installing the Endpoint](#) on page 14-3
- [Uninstalling the Endpoint](#) on page 14-9
- [Configuring Windows Endpoints](#) on page 14-10
- [Running Windows Endpoints](#) on page 14-13
- [Logging and Messages](#) on page 14-16
- [Getting the Latest Fixes and Service Updates](#) on page 14-16

Windows Operating Systems Supported

The Performance Endpoint software for 32-bit Windows runs on the following Microsoft Windows operating systems:

- Windows NT
- Windows 2000,
- Windows XP, and Windows XP Tablet PC Edition
- Windows Server 2003.
- Windows Vista (32-Bit),

Note that separate endpoint executables are provided for Windows CE, 64-bit Windows operating systems, and the Web-based endpoint. For detailed information about these endpoints, refer to the following chapters in this manual: Chapter 15, [Microsoft Windows CE](#); Chapter 16, [Microsoft Windows 64-Bit](#); Chapter 18, [Web-Based Performance Endpoint](#).

The following Microsoft Windows endpoints have been archived:

- Windows 3.1
- Windows 95 and Windows 95 with WinSock 2
- Windows 98
- Windows ME
- Windows NT 4 for Alpha
- Windows XP 64-bit Edition (IA-64)

The archived endpoints will not support new features in recent releases of Ixia products. However, they are still available from the Ixia Web site at www.ixia.com/support/ixchariot.

Installation Requirements for the 32-bit Windows Endpoint

Here is what you need to run the endpoint program with any of these 32-bit Windows operating systems: Windows NT, Windows 2000, Windows XP, Windows Server 2003, or Windows Vista:

- A computer fully capable of running the selected Windows operating system.

The minimum hardware requirements vary for each of the 32-bit Windows operating systems. Refer to your Windows documentation or to the Microsoft web site to determine the requirements for the specific operating system that you are using.

Note that the total memory requirements depends on the RAM usage of the underlying protocol stack and the number of concurrent connection pairs. For 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.
- A 32-bit version of Windows NT, Windows 2000, Windows XP, Windows Server 2003, or Windows Vista.

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

- for IP QoS: Windows 2000 requires the QoS Packet Scheduler.
- for IPv6 Multicast: Windows XP, Windows Server 2003, or Windows Vista is required.
- The latest service packs for Windows NT. On Windows NT with Service Pack 3, Microsoft Internet Explorer version 4.0 and higher is required. Service Pack 6 is not supported (use Service Pack 6a instead).

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 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, Windows 2000, and Windows XP, 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.

IxChariot 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, Windows Server 2003, and Windows Vista.

Quality of Service (QoS) support for TCP/IP is part of Microsoft Windows 2000, Windows XP, Windows Server 2003, Windows Vista., and Windows CE 6.0. On Windows NT, ToS is available for UDP and RTP only. See the *User Guide* for IxChariot for more information.

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.

- **for APPC**

The APPC protocol is no longer supported by the Ixia Performance Endpoints for Windows on any 32-bit or 64-bit operating system.

We recommend that you keep up-to-date with the latest Windows operating system service levels. [Getting the Latest Fixes and Service Updates](#) on page 14-16 discusses where to get the latest software upgrades.

Installing the Endpoint

You can install the Performance Endpoint using the Windows Control Panel, or using unattended (silent) mode:

- For Control Panel installation, refer to [Installing from CD-ROM](#) on page 14-4 or [Installing from a Downloaded Executable](#) on page 14-6.
- For unattended installation, refer to [Silent Mode Installation](#) on page 14-8.

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 14-10 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.

Performance Endpoint Filenames

There are two 32-bit Windows Performance Endpoint files:

- `pewindows32_Mn.exe` – Use this file for all 32-bit Windows operating systems except for Windows Vista.
- `pevista32_Mn.exe` – Use this file only for the 32-bit Windows Vista operating system.

The Performance Endpoint file names identify the product release: “M” represents the major version and “m” represents the minor version. For example, the 32-bit Windows Performance Endpoint for Release 6.40 is named `pewindows32_640.exe`.

User and System Permission Requirements

The endpoint for 32-bit Windows 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 `C:\Program Files\Ixia\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 try to install from the wrong mode.

Before Installing an Older Endpoint

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.

Installing from CD- ROM

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

1. Shut down any programs that are running.
2. Put the CD-ROM in your CD-ROM drive.
3. If the installer does not start automatically, display the files on the CD-ROM drive and double-click the installer executable (`pewindows_Mm.exe`).

The installer displays a splash screen and the first installation dialog:



4. Click Next to continue.

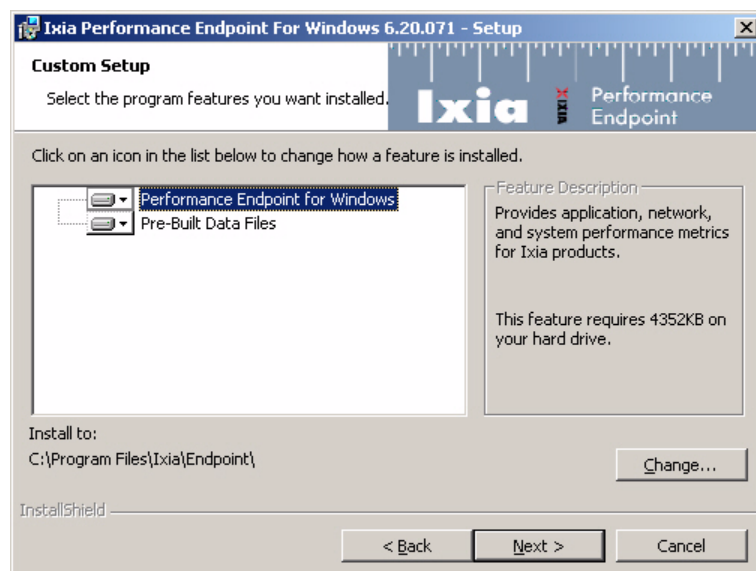
The installer displays the Ixia Software End User License Agreement.

5. To proceed with the installation, Click Yes to accept the license agreement.

The installer checks for an existing endpoint installation. If there is an endpoint installed that is of an earlier version, it gives you the option to either delete it or quit the installation. If there is an endpoint of the same version, it provides options for repairing, modifying, or removing the endpoint.

6. If the installer displays the Previous Version Detected dialog, select “Remove the Performance Endpoint”, then click Next to proceed with the installation.

The installer removes the prior endpoint (if necessary), and then displays the Custom Setup dialog.



7. If you want to install the endpoint in a folder other than the default folder (C:\Program Files\Ixia\Endpoint), click Change, then specify the path.

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.

8. Select the desired options from the Performance Endpoint Installation Options.

There are two options, both of which are selected by default:

- Performance Endpoint for Windows: You cannot de-select this option.
- Pre-built Data Files: This option allows you to set various data types (in addition to ZEROS and NOCOMPRESS) during testing. We recommend you leave this option selected. You can save a small amount of disk space by not installing the files used for compression testing; however, 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.

9. Click Install when the Ready to Install dialog appears.

The installer now copies the files and installs the Performance Endpoint.

10. Click Finish when the Setup Complete dialog appears.

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

When you've completed installation, refer to [Configuring Windows Endpoints](#) on page 14-10 to make sure your endpoint is ready for testing and monitoring.

Installing from a Downloaded Executable

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

1. Shut down any programs that are running.
2. Download and save the `pewindows_Mm.exe` file to a local directory.
3. Use the Windows Explorer to navigate to the file and double-click to start the installation.

The installer displays a splash screen and the first installation dialog:



4. Click Next to continue.

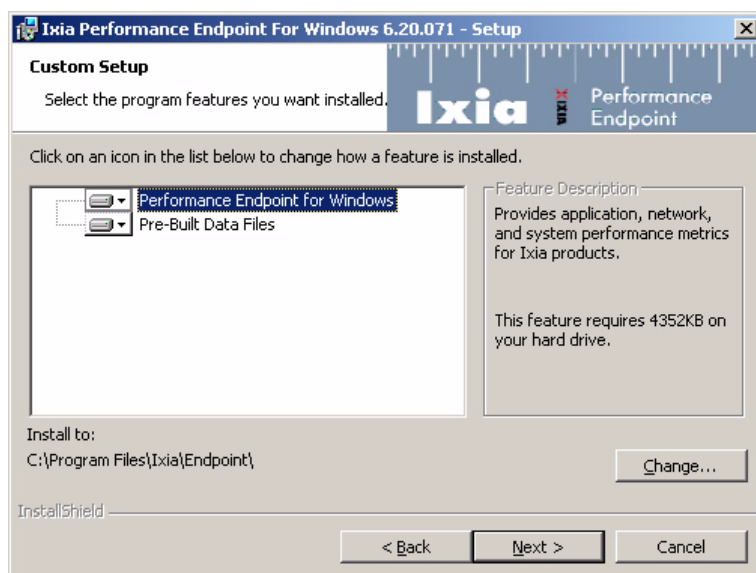
The installer displays the Ixia Software End User License Agreement.

5. To proceed with the installation, Click Yes to accept the license agreement.

The installer checks for an existing endpoint installation. If there is an endpoint installed that is of an earlier version, it gives you the option to either delete it or quit the installation. If there is an endpoint of the same version, it provides options for repairing, modifying, or removing the endpoint.

6. If the installer displays the Previous Version Detected dialog, select "Remove the Performance Endpoint", then click Next to proceed with the installation.

The installer removes the prior endpoint (if necessary), and then displays the Custom Setup dialog.



7. If you want to install the endpoint in a folder other than the default folder (C:\Program Files\Ixia\Endpoint), click Change, then specify the path.

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.

8. Select the desired options from the Performance Endpoint Installation Options.

There are two options, both of which are selected by default:

- Performance Endpoint for Windows: You cannot de-select this option.
- Pre-built Data Files: This option allows you to set various data types (in addition to ZEROS and NOCOMPRESS) during testing. We recommend you leave this option selected. You can save a small amount of disk space by not installing the files used for compression testing; however, 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.

9. Click Install when the Ready to Install dialog appears.

The installer now copies the files and installs the Performance Endpoint.

When you've completed installation, refer to [Configuring Windows Endpoints](#) on page 14-10 to make sure your endpoint is ready for testing and monitoring.

Silent Mode Installation

To install the Performance Endpoint using silent mode, enter the following command from the command line:

```
C:\>filename /s /v"/l*v install.log /qb!"
```

where *filename* is the name of the specific Performance Endpoint. For example, for the release 6.40 Windows Vista Performance Endpoint, the command is:

```
C:\>pevista32_640.exe /s /v"/l*v install.log /qb!"
```

This command performs a silent install operation and generates a log file containing a report of the actions performed.

Installing the Windows Endpoint with SMS

You can automatically install and uninstall Performance Endpoints, using Microsoft's Systems Management Server (SMS). Refer to your SMS documentation for instructions.

What Happens During Installation

Here is what happens during the installation steps. Let's say you install the endpoint into the directory C:\Program Files\Ixia\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 Chapter 3, *Endpoint Initialization File* for information about tailoring this file for individual endpoints.

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 14-13.

Uninstalling the Endpoint

You can uninstall the Performance Endpoint using the Windows Control Panel, or using unattended (silent) mode.

Removing the Endpoint Package (Uninstall)

To remove the Performance Endpoint package using the Windows GUI:

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 **Ixia Endpoint for Windows** and press **Add/Remove**. The uninstallation program begins. After the program is completed, the endpoint should be uninstalled.

Silent Mode Uninstall

To uninstall the Performance Endpoint using silent mode, enter the following command from the command line:

```
C:\>filename /s /x /v"/1*v uninstall.log /qb!"
```

where *filename* is the name of the specific Performance Endpoint. For example, for the release 6.40 Windows Vista Performance Endpoint, the command is:

```
C:\>pevista32_640.exe /s /x /v"/1*v uninstall.log /qb!"
```

This command performs a silent uninstall operation and generates a log file containing a report of the actions performed.

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.ixiacom.com/support/chariot/knowledge_base.php.

Configuring Windows Endpoints

The endpoint program uses the network application programming interfaces, such as Winsock, 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 32-bit Windows:

- [Sockets Port Number](#) on page 14-10
- [Windows Configuration for IPX and SPX](#) on page 14-10
- [Windows Configuration for TCP/IP](#) on page 14-11

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies “port_number=AUTO” on the `CONNECT_ACCEPT` command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Windows 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 IxChariot, 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 map-

ping once, and use the alias names ever after. The underlying file, named `spxdir.dat`, is like the `HOSTS` file used in TCP/IP.

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 similar to the following:

```
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.

An IxChariot Console user may observe that stopping can take between 20 and 50 seconds when running connections using SPX on Windows NT, doing loop-back (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 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. IPv6 addresses are represented by up to 8 colon separated hex digit pairs, such as 0::FF. An alternative, domain names are in a format that is easier to recognize and remember, such as www.ixiacom.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 computer's local IP address, enter the following command:

```
IPCONFIG
```

If your TCP/IP stack is configured correctly, your output will look similar to the following:

```
Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : ixiacom.com
    IP Address. . . . . : 10.200.24.12
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.200.24.1
```

Its local IP address is shown in the first row; here it's 10.200.24.12.

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 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 IxChariot or Qcheck Console 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 on a 32-bit Windows operating systems, as well as some of the messages and information that become available during testing with this endpoint. The endpoint is controlled from the Services dialog box. For Windows 2000, 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 “Ixia Endpoint.”

Only a user ID with Administrator authority is permitted to start or stop Windows NT, Windows 2000, Windows XP, Windows Server 2003, or Windows Vista 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 the operating system. There are two ways to restart the endpoint service:

1. At a command prompt, enter:

```
net start IxiaEndpoint
```

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

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 IxiaEndpoint
```

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

Disable Your Screen Saver

Screen savers in Windows can significantly lower the throughput that is 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 32-bit Windows 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 IxChariot.

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).

SetAddr 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.

SetAddr Known Limitations:

- IPv4 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 **Ixia 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.

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

1. On the Start menu, click **Settings**, then **Control Panel**.
The Control Panel opens.
2. Double-click the **Services** icon.
3. Double-click **Ixia 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 Ixia 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 IxChariot or Qcheck Console, 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.

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 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>.

15

Microsoft Windows CE

This chapter describes the installation, configuration, and operation of the Performance Endpoint software for Microsoft Windows CE.

Topics in this chapter:

- [*Available Performance Endpoints for Windows CE*](#) on page 15-1
- [*Installation Requirements*](#) on page 15-2
- [*Network Protocol Stacks*](#) on page 15-2
- [*Endpoint Installation for Windows CE*](#) on page 15-3
- [*Removing the Endpoint Package \(Uninstall\)*](#) on page 15-5
- [*Windows CE Configuration for TCP/IP*](#) on page 15-5
- [*Running Windows CE Endpoints*](#) on page 15-6
- [*Logging and Messages*](#) on page 15-8
- [*Limitations of the Windows CE Endpoint*](#) on page 15-8

Available Performance Endpoints for Windows CE

Ixia provides four distinct Performance Endpoints for Windows CE:

- **pewcex86** – Performance Endpoint for Windows CE running on Intel x86 processors.
- **pewcearm** – Performance Endpoint for Windows CE running on Intel Strong Arm and XScale processors.
- **pewcearm_cl** – Command line version of the Windows CE Performance Endpoint running on Intel Strong Arm and XScale processors. This is the same as the wcearm endpoint minus the GUI.
- **pewcearm_disk** – Performance Endpoint for Windows CE running on Intel Strong Arm and XScale processors, with file storage support.

You can run both streaming and non-streaming tests using the Windows CE Performance Endpoints. You can also run IP Multicast tests that include these endpoint as part of a multicast group.

Most IxChariot testing parameters are supported, but note exceptions in [Limitations of the Windows CE Endpoint](#) on page 15-8.

Installation Requirements

[Table 15-1](#) describes the requirements for installing and running the Microsoft Windows CE Performance Endpoint software.

Table 15-1. Windows CE Performance Endpoints

Windows CE Endpoint	Supported Processors	Operating System Version	RAM Required
pewcex86	Intel x86 compatible	Windows CE 4.2, 4.3, 5.0	64 MB
pewcearm	Intel Strong Arm, Intel XScale	Windows CE 4.2, 4.3, 5.0	64 MB
pewcearm_disk	Intel Strong Arm, Intel XScale	Windows CE 4.2, 4.3, 5.0	64 MB
pewcearm_cl	Intel Strong Arm, Intel XScale	Windows CE 4.2, 4.3, 5.0	64 MB

The Performance Endpoint file names include the product release number. For example, *pwececx86_640.exe* is version 6.40 of the Windows CE Intel x86-compatible Performance Endpoint.

The Windows CE Performance Endpoint supports Windows Mobile 5.0, a compact operating system that is packaged with a suite of basic applications for mobile devices. Windows Mobile is powered by Windows CE 5.0 and uses the .NET Compact Framework. It runs on devices such as Pocket PCs, Smartphones, and Portable Media Centers.

Network Protocol Stacks

We recommend that you configure your networking software—and make sure that it is working correctly—before installing the Performance Endpoint software.

We suggest that you use the built-in network protocol stack. In addition, you may need to purchase and configure a wireless or wired adapter.

The TCP/IP and UDP/RTP protocols are supported by the Performance Endpoint for Windows CE. The Windows CE Performance Endpoints run on any IP network, regardless of topology. For example, we have tested it with 802.11a/b/g wireless links and 10/100/1000 Ethernet links.

Endpoint Installation for Windows CE

Installing the *pewcearm* Performance Endpoint

The following installation instructions assume that the Windows CE device to be tested is already synched to your desktop computer:

Follow these steps to install the GUI version of the endpoint:

1. From your desktop PC, navigate to the Windows CE endpoint at www.ixiacom.com/support/ixchariot.
2. Download the Windows CE endpoint package to your desktop PC.
3. Copy the file `pewcearm_Mm.exe` to the Windows Clipboard using the Windows Explorer. *Mm* is the endpoint release number; for example, 640 for release 6.40.
4. Paste the file to the following directory:

[Mobile Device]\My Pocket PC\Windows\Start Menu

The endpoint is now ready for use. Refer to [Running Windows CE Endpoints](#) on page 15-6 for additional instructions.

Installing the *pewcearm_cl* Performance Endpoint

Follow these steps to install the command line version of the endpoint:

1. From your desktop PC, navigate to the Windows CE endpoint at www.ixiacom.com/support/ixchariot.
2. Download the Windows CE endpoint package to your desktop PC.
3. Copy the file `pewcearm_cl_Mm.exe` to your Windows CE device, using the tools available on your device.

Once you have copied the endpoint, it is ready for use. Refer to [Running Windows CE Endpoints](#) on page 15-6 for additional instructions.

Installing the *pewcearm_disk* Performance Endpoint

Follow these steps to install the `pewcearm_disk` endpoint:

1. From your Windows CE device, use your Web browser to navigate to the Windows CE endpoint at www.ixiacom.com/support/ixchariot.
2. Download the **pewcearm-disk-Mm.exe** self-extracting archive file to a disk drive on your Windows CE device.
3. Double-click the `pewcearm-disk-Mm.exe` file to extract the contents.

You can place these files anywhere on the storage device (whether in the root directory, or in a user-defined directory). The self-extracting archive includes all the files you need to run the Performance Endpoint, including:

- `pewcearm_disk.exe` (the Performance Endpoint executable)
- `endpoint.ini`
- `echr.msg`
- the `cmpfiles` directory

Installing the pewcex86 Performance Endpoint

4. Modify the endpoint.ini, as required for your testing.

Refer to Chapter 3, *Endpoint Initialization File*, for information about the endpoint.ini file.

Once you have copied the endpoint, it is ready for use. Refer to *Running Windows CE Endpoints* on page 15-6 for additional instructions.

The following installation instructions assume that the Windows CE device to be tested is already synched to your desktop computer:

Follow these steps to install the endpoint:

1. From your desktop PC, navigate to the Windows CE endpoint at www.ixiacom.com/support/ixchariot.
2. Download the Windows CE endpoint package to your desktop PC.
3. Copy the file `pewcex86_Mm.exe` to the Windows Clipboard using the Windows Explorer. *Mm* is the endpoint release number; for example, 630 for release 6.30.
4. Paste the file to the following directory:

```
[Mobile Device]\My Pocket PC\Windows\Start Menu
```

The endpoint is now ready for use. Refer to *Running Windows CE Endpoints* on page 15-6 for additional instructions.

Alternate Installation

Since the Windows CE for the x86 architecture is similar to standard Windows, the `pewcex86_Mm.exe` executable may be copied from another computer via a network share or FTP. *Mm* is the endpoint release number; for example, 640 for release 6.40. It may be installed in any location on the Windows CE drive and executed from that location.

Note: If the Start menu on the Pocket PC where you're installing the endpoint has already reached the maximum number of icons it can display, the endpoint software is automatically copied to the directory `[Mobile Device]\My Pocket PC\Windows\Start Menu\Programs`.

See the following HP business support document for more information:
http://h20000.www2.hp.com/bizsupport/TechSupport/Document.jsp?locale=en_US&taskId=115&prodSeriesId=306693&prodTypeId=215348&objectID=PSD_MH030919_CW01.

Removing the Endpoint Package (Uninstall)

The following installation instructions assume that the Windows CE pocket PC or device to be tested is already synched to your desktop computer:

Delete `pewcearm_Mm.exe` (for the Strong Arm version) or `pewcx86_Mm.exe` (for the x86 based version) from the following directory on your desktop PC:

```
[Mobile Device]\My Pocket PC\Windows\Start Menu
```

On the x86 version, if the `pewcx86_Mm.exe` executable was installed in an alternate location, find and delete that file. *Mm* is the endpoint release number; for example, 640 for release 6.40.

Windows CE 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. The alternative, domain name, is in a format that is easier to recognize and remember, such as `www.ixiacom.com`. To use domain names, you need a Domain Name Server (DNS) set up in your network.

Determining Your IP Network Address

On your Windows CE device, tap **Start > Settings > Connections** and tap the **Network Adapters** icon. Select an adapter and then tap **Properties**.

Look at your adapter configuration. If you are using DHCP, your adapter configuration may not show your address. In that case, contact your network administrator to find out which IP address the DHCP server has assigned to the adapter.

If you are using the command line version of the endpoint, the procedure for determining your IP address depends on the tools available on the device.

Testing the TCP 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 xxx.xxx.xxx.xxx
```

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

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

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies “port_number=AUTO” on the CONNECT_ACCEPT command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the CONNECT_ACCEPT commands (usually Endpoint 2) uses the port number specified in the script.

Running Windows CE Endpoints

The following sections describe how to start and stop an endpoint, and how to check the version of an endpoint. A final section describes how the endpoint handles error messages.

Intel Strong Arm and XScale Processor Based Operation

Following are instructions for starting and stopping the three versions of the pewcearm Performance Endpoint.

Starting the *pewcearm* Performance Endpoint

On your Windows CE device, tap **Start > pewcearm_Mm.exe**. *Mm* is the endpoint release number; for example, 640 for release 6.40.

Starting the *pewcearm_cl* Performance Endpoint

Procedures for starting and stopping the command line version of the Performance Endpoint depend on the tools available on the device. For example, for some devices you will enter **endpoint** at the command line to start the endpoint and use CTRL-C to stop the endpoint.

Starting the *pewcearm_disk* Performance Endpoint

On your Windows CE device, navigate to the directory where you have installed the Performance Endpoint files, then tap the executable (*pewcearm_disk.exe*) to start the endpoint.

Stopping the *pewcearm* and *pewcearm_disk* Performance Endpoint

To stop the endpoint program, use the following menu path on your Windows CE device:

1. Tap **Start > Settings > System > Memory > Running Programs**.
2. Select **Performance Endpoint** and then tap **Stop**.

Stopping the *pewcearm_cl* Performance Endpoint

Procedures for starting and stopping the command line version of the Performance Endpoint depend on the tools available on the device. For example, for some devices you will enter **endpoint** at the command line to start the endpoint and use CTRL-C to stop the endpoint.

Intel x86 Processor Based Operation

Starting the *pewcex86* Endpoint

On your Windows CE device, tap **Start > pewcex86_Mm.exe**. If the executable was installed in an alternate location, find and tap on the *pewcex86_Mm.exe* executable. *Mm* is the endpoint release number; for example, 640 for release 6.40.

Stopping the *pewcex86* Endpoint

To stop the *pewcex86* endpoint program, use the following menu path on your Windows CE device:

1. Click on the **X** at the top right corner of the application, or use the **File > Exit** menu choice.

NOTE: On some versions of Windows CE, such as the iPac, the Ixia endpoint application is surrounded by an outer window. Make sure to press the **X** on the inner window to stop the endpoint.

Checking the Endpoint Version

The current version should be displayed on the endpoint main window.

If you are using the command line version of the endpoint, the procedure for displaying the endpoint version depends on the tools available on the device.

Logging and Messages

All error messages encountered on an endpoint are returned to the IxChariot or Qcheck Console.

For the `pewcearm_disk` Performance Endpoint, some error messages are logged to disk. These messages 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
```

In addition, if an assertion failure occurs, the `pewcearm_disk` Performance Endpoint writes a file named `assert.err` to the directory where you installed the endpoint.

Note that only the `pewcearm_disk` Performance Endpoint provides support for disk storage. The other Windows CE Performance Endpoints (`pewcex86`, `pewcearm`, and `pewcearm_cl`) do not provide disk support.

Limitations of the Windows CE Endpoint

The Windows CE Performance Endpoints do not support the following IxChariot test parameters:

- Disabling the UDP checksum.
- DiffServ QoS templates.
- Traceroute testing.
- Application scripts with `.cmp` data files as the datatype.

Scripts that use `.cmp` files by default, such as the Internet scripts, will run only on the `pewcearm_disk` version of this endpoint.

As a work-around on the other versions of the endpoint, edit the scripts to use `NOCOMPRESS` as the `send_datatype` instead of a `.cmp` file.

Additional Limitations:

- Support for CPU Utilization on Windows CE is device-dependent. For more information, see <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/wcemain4/html/cerefGetIdleTime.asp>.
- By default, Windows CE will not support a UDP IxChariot test with a datagram window of more than two datagrams. The test will time out with error message **CHR0216**. This problem will only occur if you adjust the `send_buffer_size` or Window Size parameter to include more than two UDP datagrams in a window.

This Windows CE limitation has been documented in the Microsoft Knowledge Base article Q290206. The article explains that the default internal UDP buffer queue size on Windows CE is 2. To support applications that deliver more than 2 datagrams in a very short time, the default limit can be raised to a value between 2 and 10 hex. For example, change the following Registry setting:

```
[HKEY_LOCAL_MACHINE\Comm\Afd]  
DgramBuffer=dword:8
```

The device must be reset for this parameter to take effect.

16

Microsoft Windows 64-Bit

This chapter explains the installation, configuration, and operation of the Performance Endpoint software for 64-bit Microsoft Windows operating systems.

Topics in this chapter:

- *Operating Systems and Processors Supported* on page 16-1
- *Installation Requirements for the Windows 64-Bit Endpoint* on page 16-2
- *Microsoft Windows 64-Bit Performance Endpoint Installation* on page 16-3
- *Uninstalling the Endpoint* on page 16-5
- *Configuring Windows 64-bit Performance Endpoints* on page 16-6
- *Running Microsoft Windows 64-Bit Performance Endpoints* on page 16-8
- *Logging and Messages* on page 16-12
- *Getting the Latest Fixes and Service Updates* on page 16-12

Operating Systems and Processors Supported

The Performance Endpoint software runs on the following 64-bit Microsoft Windows operating systems:

- Windows Vista (64-bit)
- Windows XP x64 Edition (Workstation and Server versions)
- Windows Server 2003 64-bit Edition

This endpoint supports the 64-bit Windows operating systems running on the following processor families:

- AMD64 processors (such as the AMD Opteron/ Athlon FX/Athlon 64 CPU).
- Intel Extended Memory 64 Technology (Intel EM64T) processors, including the Xeon processor.

Note that this endpoint does not support the Intel Itanium processors.

Separate endpoint executables are provided for 32-bit Windows systems, Windows CE, the Web-based endpoint, and the archived Windows endpoints. For information about these endpoints, refer to the following chapters in this manual: Chapter 15, *Microsoft Windows CE*; Chapter 14, *Microsoft Windows 32-Bit*; Chapter 18, *Web-Based Performance Endpoint*.

Installation Requirements for the Windows 64-Bit Endpoint

The installation requirements for Microsoft Windows 64-Bit Performance Endpoint are:

- A computer equipped with either of the processor types:
 - AMD64 processor, including the AMD Opteron, Athlon FX, or Athlon 64 processors.
 - Intel Extended Memory 64 Technology (Intel EM64T) processor, including the Xeon processor.
- 512 MByte of random access memory (RAM).

The Microsoft recommended hardware requirements vary for each of the 64-bit Windows operating systems. Refer to your Windows documentation or to the Microsoft web site to determine the requirements for the specific operating system that you are using.

Note that 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 10 MBytes of space available.
- One of the Microsoft Windows 64-bit operating systems:
 - Windows Vista (64-bit)
 - Windows XP x64 Edition (Workstation and Server versions)
 - Windows Server 2003 64-bit Edition

We recommend that you remain up-to-date with the latest Windows service levels.

Supported Protocols

The Performance Endpoint for Microsoft Windows XP 64-bit Edition supports the following protocols:

- IPv4 and IPv6
- TCP, UDP, and RTP

The Microsoft Windows 64-bit operating systems do not support IPX and SPX.

Microsoft Windows 64-Bit Performance Endpoint Installation

You can install the Performance Endpoint using the Windows GUI, or using unattended (silent) mode:

- For interactive installation, refer to [Interactive Installation](#) on page 16-3.
- For unattended installation, refer to [Silent Mode Installation](#) on page 16-5.

We recommend configuring your networking software—and ensuring that it is working correctly—before installing Ixia Endpoint software. See the Help for your networking software, and see [Configuring Windows 64-bit Performance Endpoints](#) on page 16-6 for more information.

Performance Endpoint Filenames

There are two 64-bit Windows Performance Endpoint files:

- `pewindows_64bit_Mn.exe` – Use this file for all 64-bit Windows operating systems except for Windows Vista.
- `pevista64_Mn.exe` – Use this file only for the 64-bit Windows Vista operating system.

The Performance Endpoint file names identify the product release: “M” represents the major version and “m” represents the minor version. For example, the 64-bit Windows Vista Performance Endpoint for Release 6.40 is named `pevista64_640.exe`.

User and System Permission Requirements

The endpoint for Windows 64-bit Performance Endpoint is installed and runs as a service. Only a user with Administrator authority is permitted to install services. To successfully install the endpoint, you must be logged in with Administrator authority. If you are installing the endpoint in an NTFS directory, the permissions of the directory 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 Ixia\Endpoint directory or the directory where you’ve installed the endpoint, plus any relevant subdirectories, if any.

Interactive Installation

To install the Microsoft Windows 64-Bit Performance Endpoint interactively:

1. Log onto the target machine with a user ID that has Administrative privileges.
2. Either download the endpoint executable (`pewindows_64bit_Mn.exe` or `pevista64_Mn.exe`) from the Ixia web site, or access it from the product CD.
3. Double-click the file to start the installation.

The installer displays a splash screen and the Welcome dialog.

4. Click Next to continue.

The installer displays the Ixia Software End User License Agreement.

5. To proceed with the installation, Click Yes to accept the license agreement.
6. If you have a Performance Endpoint already installed on your machine, the installer detects this and displays the Installation Options dialog.

To uninstall the older version, follow these steps:

a: Ensure that the “Remove existing version of the product” is selected.

b: Select *Next* to continue.

The installer removes the older version of the Performance Endpoint from your machine.

c: Select *OK* to continue.

The installer displays the Custom Setup dialog.

7. Select *Next* to use the standard setup selections and continue with the installation.

We recommend that you install the pre-built data files (this is the default behavior). You can save a small amount of disk space by not installing the files used for compression testing; however, 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.

8. If you want to install the endpoint in a folder other than the default folder (C:\Program Files\Ixia\Endpoint), click Browse, then select the path.

We recommend installing the endpoint 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.

9. Click Next to continue.

The installer displays the Start Copying Files dialog.

10. Click Next to continue.

The installer now copies the files and installs the Performance Endpoint. Once the installation is complete, the installer displays the Setup Complete dialog.

11. Click Finish to complete the installation.

Windows services are controlled from the Services dialog box, accessible by selecting **Programs>Administrative Tools>Services** from the Start menu. If you want to restart a service without restarting Windows, use the Services dialog box. Go to the Services dialog, select **Ixia Endpoint**, and select a Startup type from the pull-down. Press Start to start the endpoint.

You can also manually start the endpoint after installation. See [Starting a Windows 64-bit Performance Endpoint](#) on page 16-8 for instructions.

To prevent the endpoint from running automatically on startup, see [Disabling Automatic Startup](#) on page 16-9.

When you’ve completed installation, refer to [Configuring Windows 64-bit Performance Endpoints](#) on page 16-6 to make sure your endpoint is ready for testing and monitoring.

Silent Mode Installation

To install the Performance Endpoint using silent mode, enter the following command from the command line:

```
C:\>filename /s /v"/l*v install.log /qb!"
```

where *filename* is the name of the specific Performance Endpoint. For example, for the release 6.40 Windows Vista Performance Endpoint, the command is:

```
C:\>pevista64_640.exe /s /v"/l*v install.log /qb!"
```

This command performs a silent install operation and generates a log file containing a report of the actions performed.

What We Do During Installation

Here's what happens during the installation steps. Let's say you install the endpoint into the directory `\Program Files\Ixia\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 Chapter 3, [Endpoint Initialization File](#) for information about tailoring the `.ini` file for individual endpoints.

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. Controlling the endpoint from the Services dialog box is described in [Running Microsoft Windows 64-Bit Performance Endpoints](#) on page 16-8.

Uninstalling the Endpoint

You can uninstall the Performance Endpoint using the Windows Control Panel, or using unattended (silent) mode.

Removing the Endpoint Package (Uninstall)

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

1. Click **Start > Settings > Control Panel**.
2. Click **Add or Remove Programs**. The Add or Remove Programs Properties dialog box is shown.
3. Highlight **Ixia Endpoint** and press **Change/Remove**.

The un-installation program begins. After the program is completed, the endpoint should be uninstalled.

Silent Mode Uninstall

To uninstall the Performance Endpoint using silent mode, enter the following command from the command line:

```
C:\>filename /s /x /v"/1*v uninstall.log /qb!"
```

where *filename* is the name of the specific Performance Endpoint. For example, for the release 6.40 Windows Vista Performance Endpoint, the command is:

```
C:\>pevista64_640.exe /s /x /v"/1*v uninstall.log /qb!"
```

This command performs a silent uninstall operation and generates a log file containing a report of the actions performed.

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.ixiacom.com/support/chariot/knowledge_base.php.

Configuring Windows 64-bit Performance Endpoints

The endpoint program uses network application programming interfaces such as WinSock 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 topics describe how to accomplish these steps for the Windows 64-bit Performance Endpoint.

64-bit Windows 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 (IPv4) or 128-bit (IPv6) numeric address. IPv4 addresses are represented in dotted notation as a set of four numbers separated by periods, such as 199.72.46.202. IPv6 addresses are represented by up to 8 colon separated hex digit pairs, such as 0::FF. An alternative—domain names—are in a format that is easier to recognize and remember, such as www.ixiacom.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 an 64-bit Windows computer's local IP address, enter the following at a command prompt:

```
IPCONFIG
```

If your TCP/IP stack is configured correctly, your output will look like the following:

```
Windows IP Configuration

Ethernet adapter Local Area Connection:
    Connection-specific DNS Suffix . : 
    IP Address. . . . . : 10.41.2.19
    Subnet Mask . . . . . : 255.255.0.0
    Default Gateway . . . . . : 10.41.1.254
```

The local IP address is shown in the first row; here it is 10.41.2.19.

For IP addresses not configured by DHCP, you can also find your IP address using the graphical user interface. Select **Start - Settings - Control Panel**, then double-click on the **Network Connections** icon. Select **Local Area Connection** and click **Properties**. In the Local Area Connection Properties dialog box, double-click **Internet Protocol (TCP/IP)** in the list. Your IP address and subnet mask are shown.

To determine a 64-bit Windows computer's local hostname, enter the following at a command prompt:

```
HOSTNAME
```

The current hostname is shown in the first row.

From the graphical user interface, return to Internet Protocol (TCP/IP) configuration. Press Advanced and then select the DNS tab to see or change your DNS servers. If the DNS tab is empty, avoid using domain names as network addresses; use numeric IP addresses instead.

The default location for the `/etc/hosts` file is the following:

```
c:\WINDOWS\SYSTEM32\DRIVERS\ETC\HOSTS
```

Trying Out the TCP/IP Connection

Ping and *ping6* are simple utility programs, included in all TCP/IP implementations. They are used to check the connection from one computer to another using either IPv4 or IPv6 addresses. For *ping*, enter the following at a 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.

For *ping6* enter an address in standard IPv6 format.

Make sure that you can run ping/ping6 successfully from the IxChariot or Ixia Qcheck Console to each computer serving as Endpoint 1, and between each pair

of endpoints involved in a test, before starting your testing with TCP/IP. When using alternate networks, the alternate networks need to be tested as well.

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies “port_number=AUTO” on the `CONNECT_ACCEPT` command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the `CONNECT_ACCEPT` commands (usually Endpoint 2) uses the port number specified in the script.

Running Microsoft Windows 64-Bit Performance Endpoints

The following topics describe starting and stopping an endpoint running on a Microsoft Windows 64-bit operating system, as well as some of the messages and information that become available during testing with this endpoint. The Windows 64-bit Performance Endpoint is controlled from the Services dialog box, which you access by selecting **Start - Settings - Administrative Tools - Services** from the Start menu. The Services dialog box lets you start or stop the endpoint, listed as **Ixia Endpoint**.

Only a user with Administrator authority is permitted to start or stop 64-bit Windows services.

Starting a Windows 64-bit Performance 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 the operating system. There are two ways to restart the endpoint service:

1. At a command prompt, enter:

```
net start IxiaEndpoint
```
2. In the Services dialog box, double-click Ixia Endpoint and press Start. The status changes to “started” when the endpoint is successfully started.

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

Stopping a Windows 64-bit Performance Endpoint

There are two ways to stop the endpoint service:

1. At a command prompt, enter the following:

```
net stop IxiaEndpoint
```
2. In the Services dialog box, double-click Ixia Endpoint and click Stop. The status is blank when the endpoint program has stopped.

Disable Your Screen Saver

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

Disable NIC Power Save Mode

If your NIC is configured to power down after some period of non-traffic, this might cause your test to fail.

Disabling Automatic Startup

To disable the automatic starting of the Windows 64-bit Performance Endpoint, take the following steps:

1. From the Windows Start menu, select **Programs\Administrative Tools\Services**. The Services dialog is shown.
2. Double-click **Ixia Endpoint**.
3. From the Startup type menu, select Manual.
4. Press OK to save the new setting and exit the dialog. 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 64-bit Performance Endpoint Is Active

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

The SetAddr Utility for 64-bit Windows

Endpoints for Windows operating systems now ship with a utility that helps you quickly create virtual IP addresses on 64-bit Windows 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 IxChariot.

When you install a Windows endpoint, Setaddr.exe for 64-bit Windows is automatically installed in the same directory. 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).

SetAddr 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 1,000 virtual addresses.

SetAddr Known Limitations

- A version of SetAddr is also available for Windows NT, Windows 2000, and Windows XP/2003 32-bit computers. This 64-bit Windows version of SetAddr does not work on 32-bit systems.
- 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 have bench-

marked 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.

Logging and Messages

While most endpoint error messages are returned to the IxChariot or Ixia Qcheck Console, 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`. 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 has 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.

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. To keep your 64-bit Windows operating system up-to-date, you should use the Windows Update function available from your Start Menu.

17

Sun Solaris

This chapter explains the installation, configuration, and operation of the Performance Endpoint software for Sun Solaris version 2.4 (or later).

Topics in this chapter:

- *Platforms Supported* on page 17-1
- *Installation Requirements for Solaris Endpoints* on page 17-1
- *Endpoint Installation for Sun Solaris* on page 17-2
- *Removing the Endpoint Package (Uninstall)* on page 17-7
- *Configuring Solaris Endpoints* on page 17-7
- *Running Solaris Endpoints* on page 17-8
- *Logging and Messages* on page 17-10
- *Updates for Sun Solaris* on page 17-11

Platforms Supported

The Performance Endpoint software operates on the “SPARC” and “x86” versions of Solaris.

- SPARC computers contain CPUs made by Sun Microsystems and others.
- x86 computers are commonly known as “Intel-compatible PCs”; they contain CPUs made by Intel, AMD, Cyrix, or others.

Installation Requirements for Solaris Endpoints

Here’s what you need to run the endpoint program with Sun Solaris:

- A computer capable of running Sun Solaris well.

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

For x86 computers, this implies a CPU such as an Intel 80386, 80486, a member of the Pentium family, or equivalent. A Pentium or better is recommended.

- At least 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 large tests involving hundreds of connections through a single endpoint, additional memory may be required.

- A hard disk with at least 4 MBytes of space available.
- Sun Solaris version 2.4 or later, with TCP/IP networking and corresponding networking hardware installed and configured. This version also supports IP Multicast.
- An Acrobat Reader to view the PDF files.

Acrobat readers are loaded on most computers for viewing other documents, but if you do not have one, they are available at Adobe's Web site:

www.adobe.com/prodindex/acrobat/readstep.html.

Endpoint Installation for Sun Solaris

Performance Endpoint File Name

The name of the Solaris Performance Endpoint file is `pesun_Mm.tar`, where *Mm* is the major and minor IxChariot version number; for example *630* for IxChariot release 6.30.

Installation Procedures

First, make sure that you are logged in as a “root” user. Also, remember that all the commands and parameters discussed here are case-sensitive; use the combination of uppercase and lowercase letters as shown. The following instructions explain how to install an endpoint from a CD-ROM and from the World Wide Web.

Note: To install version 4.4 of the Performance Endpoint for Sun Solaris over a previous version of the endpoint, you need to modify the admin file to contain “instance=overwrite” and “conflict=nocheck.”

Installation from CD-ROM

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

1. Put the CD-ROM in your CD-ROM drive.
2. Next, enter the `VOLCHECK` command, which tells Solaris that the CD-ROM is inserted in the drive and is readable. `VOLCHECK` returns quickly to the command prompt, without a message.

`volcheck`
3. The CD-ROM contains an archive of the endpoint package. First use the `rm` command to ensure a clean temporary install directory. Then use the `tar` command to extract the archive contents from the CD-ROM.

For SPARC systems, enter:

```
cd /tmp
rm -fr endpoint
tar -xvf /cdrom/endpoint/solaris/pesun_Mm.tar
```

For x86 systems, enter:

```
cd /tmp
rm -fr endpoint
tar -xvf /cdrom/endpoint/s86/pes86_Mm.tar
```

4. Next, install the endpoint package using the `pkgadd` command:

```
pkgadd -d /tmp endpoint
```

The `pkgadd` command is not part of the endpoint installation. It is part of the standard Solaris installation and can be found in the `/usr/bin` directory.

5. You will see the license agreement, presented with the `pg` command. Press the spacebar until the end of the agreement is displayed. You are asked whether you accept the terms and conditions of the agreement. If you do, enter “accept_license” and press Return.

6. Next, you are asked the following question:

```
This package contains scripts which will be executed
with super user permission during the process of
installing this package.
```

```
Do you want to continue with the installation of this
package [y,n,?]
```

Enter a lowercase “y” to complete the installation script. About 20 lines of text give the status of the installation. When it’s finished, the last line reads:

```
Installation of <endpoint> was successful.
```

You may instead see the following message:

```
Notice! There were potential problems with migrating
from $oldInstallPath to $installPath. Review the
warnings displayed above for further explanation.
```

If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.

7. Use the following commands to delete the archive contents from the temporary working directory:

```
cd /tmp
rm -fr endpoint
```

8. Remove the CD-ROM by entering `eject` at a command prompt.

This is a good time to read the `README` file, installed with the endpoint in `/opt/ixia`, for the latest information about the endpoint program.

When you’ve completed installation, refer to [Configuring Solaris Endpoints](#) on page 17-7 to make sure your endpoint is ready to be used in testing and monitoring.

Installation from the Web

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

1. First, use the `rm` command to ensure a clean temporary install directory (we'll use `tmp` in this example).

For SPARC systems:

- Download the `pesun_Mm.tar.Z` file to the `/tmp` directory.
- Uncompress the endpoint file by using the `uncompress` command:

```
cd /tmp
uncompress pesun_Mm.tar
tar -xvf pesun_Mm.tar
```

For x86 systems:

- Download the `pes86_Mm.tar.Z` file to the `/tmp` directory.
- Uncompress the endpoint file by using the `uncompress` command:

```
cd /tmp
uncompress pes86_Mm.tar
tar -xvf pes86_Mm.tar
```

2. Next, install the endpoint package using the `pkgadd` command:

```
pkgadd -d /tmp endpoint
```

The `pkgadd` command is not part of the endpoint installation. It is part of the standard Solaris installation and can be found in the `/usr/bin` directory.

3. You will see the license agreement, presented with the `pg` command. Press the spacebar until the end of the agreement is displayed. You are asked whether you accept the terms and conditions of the agreement. If you do, enter "accept_license."

4. You are next asked the following question:

```
This package contains scripts which will be executed
with super user permission during the process of
installing this package. Do you want to continue with
the installation of this package [y,n,?]
```

Enter a lowercase "y" to complete the installation script. About 20 lines of text give the status of the installation. When it's finished, the last line reads, "Installation of <endpoint> was successful."

You may instead see the following message:

```
Notice! There were potential problems with migrating
from $oldInstallPath to $installPath. Review the
warnings displayed above for further explanation.
```

If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.

5. Use the following commands to delete the archive contents from the temporary working directory:

```
cd /tmp
rm -fr endpoint
rm pes86_Mm.tar
```

This is a good time to read the `README` file, installed with the endpoint in `/opt/ixia`, for the latest information about the endpoint program.

When you've completed installation, refer to [Configuring Solaris Endpoints](#) on page 17-7 to make sure your endpoint is ready to be used in testing and monitoring.

Installation Defaults File for Solaris

The admin file defines default installation actions to be taken when administrative input is required during install, for example, whether to allow a new package to overwrite an older version, whether an installation can be run with super user authority, and so on. The admin file is found in `/var/sadm/install/admin/default`. The man pages (`man -s 4 admin`) describe its format and content; please read the man pages if you are unfamiliar with the admin file.

To install version 4.4 of the Endpoint for Sun Solaris over a previous version of the endpoint, you need to modify the admin file to contain `"instance=overwrite"` and `"conflict=nocheck."`

If you want non-interactive install capability, modify the admin file to contain `"action=nocheck"` so that the endpoint package scripts can be run with super user authority.

Unattended Installation for Solaris

Unattended installation is available for the Sun Solaris endpoint. You install an endpoint once, manually, while the install facility saves your input in a *response* file. You can then install that same endpoint silently on other computers, that is, without providing input other than the response file.

First, complete the steps described in [Endpoint Installation for Sun Solaris](#) on page 17-2, using the `tar` command. Next create a response file, using the `pkgask` command:

```
pkgask -r /tmp/endpoint.response -d /tmp endpoint
```

The endpoint license agreement is displayed with the `pg` command. Press the spacebar until the end of the agreement is displayed. Next, you are asked whether you accept the terms and conditions of the agreement. If you do, enter `"accept_license."`

You should see the following displayed:

```
Response file </tmp/endpoint.response> was created.
Processing of request script was successful.
```

Use the following command to install other Solaris endpoints in unattended mode (this single command is split over two lines):

```
pkgadd -n -a /tmp/endpoint/root/opt/ixia/admin
-r /tmp/endpoint.response -d /tmp endpoint
```

The `pkgadd` command is not part of the endpoint installation. It is part of the standard Solaris installation and can be found in the `/usr/bin` directory.

When `pkgadd` is finished, the last line reads, “Installation of <endpoint> was successful.”

You may instead see the following message:

```
Notice! There were potential problems with migrating from
$oldInstallPath to $installPath. Review the warnings
displayed above for further explanation.
```

If you see this message, please review the entire output from the install script for an explanation of the warnings and further instructions.

The response file may be used to install the endpoint on each of your Sun Solaris computers.

What Happens During Installation

Here’s what happens during the installation steps. The endpoint is installed into the directory `/opt/ixia`. A directory is created with the following contents:

- The executable programs
- The `README` file
- Various install and uninstall programs
- 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 Chapter 3, *Endpoint Initialization File* for information about tailoring this file for individual endpoints.

The installation program stops any copy of the endpoint program that may currently be running and starts a copy of the newly installed endpoint. You can run tests immediately, without a reboot.

Our software copies an S81 endpoint initialization script to the `/etc/rc2.d` directory so the endpoint is started every time your system boots.

No changes are made to the `PATH` environment variable of the root user.

Should you have reason to install an older endpoint, you should delete any safestore files using 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:

```
rm *.q*.
```

3. Uninstall the current endpoint.
4. Install the desired endpoint.

Removing the Endpoint Package (Uninstall)

To remove the endpoint package from your hard disk, first stop the endpoint program (if it is running). Enter the following command:

```
/opt/Ixia/endpoint -k
```

Use the following command to remove the endpoint package (you must be logged in as root to run `pkgrm`):

```
pkgrm endpoint
```

Enter a lowercase “y” when you’re asked if you want to remove this package. About 10 lines of text give the status of the uninstallation. When it’s finished, the last line reads, “Removal of <endpoint> was successful.”

This removes the files from `/opt/ixia`, except for any files that were added to this directory that were not present at installation, such as the `endpoint.ini` file, and does not delete the directory. The removal program does not automatically delete files that have been added to the directory that you may need if you reinstall the product.

Configuring Solaris Endpoints

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.

1. Determine the network addresses of the computers to be used in tests.
2. Verify the network connections.

The following sections discuss how to accomplish these tasks.

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. The alternative, domain names are in a format that is easier to recognize and remember, such as `www.ixiacom.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

Here are two ways to determine the IP address of the local computer you’re using:

1. If you’re using Sun’s OpenWindows graphical user interface, right-click on the outer desktop background. One of the options in this Workspace menu

that pops up is Workstation Info. Click on it to display Workstation Information about your computer, including your local Internet address.

```
netstat -in
```

2. As an alternative, enter the following at a command prompt:

Your local IP address is shown in the left-hand column, if there are active connections.

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:

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

Replace the x's with the IP address of the target computer. If Ping returns a message that says "xx.xx.xx.xx is alive," the Ping worked.

Otherwise, there will be a delay, and then you'll see "no answer from xx.xx.xx.xx." This means that the Ping failed, and you can't reach the target computer.

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

Sockets Port Number

IP networks use *network addresses* to forward traffic across a network to a specific device, and they use *port numbers* to deliver traffic to a specific application running on the selected device.

IxChariot uses a designated *management port* to transport test management traffic between the console and the endpoints. The management port is one of the following:

- SPX transport: port 10117
- TCP transport: either port 10115 (the default) or a user-selected port. (Refer to [MANAGEMENT_PORT](#) on page 3-10 for more information about selecting a port for management traffic.)

IxChariot uses other ports for test traffic. If an IxChariot script specifies "port_number=AUTO" on the CONNECT_ACCEPT command, ports are dynamically acquired from the protocol stack. Otherwise, the endpoint issuing the CONNECT_ACCEPT commands (usually Endpoint 2) uses the port number specified in the script.

Running Solaris Endpoints

The following sections describe how to manually start and stop the endpoint program, and how to examine error log files if a problem occurs.

Starting a Solaris Endpoint

The endpoint program is installed so it will start automatically each time Solaris is rebooted. It sends its screen output to file `/var/adm/endpoint.console`. If you want to see any error messages generated at this endpoint, enter the following command:

```
tail -f /var/adm/endpoint.console
```

The detailed information about the start and stop of each individual connection pair is written to file `endpoint.aud`. The contents of this file vary depending on how you've set the `SECURITY_AUDITING` keyword in your `endpoint.ini` file.

See Chapter 3, [Endpoint Initialization File](#) for more information about `endpoint.aud` and `SECURITY_AUDIT` settings.

Instead of automatic startup, you can choose to manually start the endpoint program at a command prompt. Ensure that you are logged in as a “root” user. To start the endpoint, enter:

```
/opt/ixia/endpoint &
```

The “&” parameter indicates to Solaris that the endpoint program should run in the background. The screen output from the endpoint program is interleaved with other UNIX commands. Just press Return to enter more commands.

If you choose to manually start the endpoint, consider redirecting its output to the `endpoint.console` file. You can tell by the time stamp of the file when the endpoint program was started and stopped.

If the endpoint program is already running, you get the following message, “CHR0183: The endpoint program is already running. Only one copy is allowed at a time.”

Stopping a Solaris Endpoint

The endpoint program has a special command-line option, `-k`. If you have an endpoint program you'd like to kill, go to a command prompt on the same computer and enter the following (you must be logged in as root to run this program):

```
/opt/ixia/endpoint -k
```

The `-k` command-line option has the purpose of killing any endpoint program running on that computer. You should see the message “Sent exit request to the running endpoint,” which indicates that the endpoint program has been sent a request to stop.

If for some reason the request to stop is not handled by the running endpoint program correctly, you may need to use the UNIX “`kill -TERM`” command.

Cleanup after Unexpected Errors

If the endpoint should fail or be killed abnormally (or encounter assertion conditions), you may also need to do additional cleanup. If the endpoint is still running, try to stop it using the command “`endpoint -k`”. If that does not stop the endpoint, kill the endpoint using the UNIX `KILL` command.

Next, enter the following command:

```
rm /var/adm/.IXIA.ENDPOINT.PID
```

How to Tell If a Solaris Endpoint Is Active

You can use traditional UNIX commands to determine if the endpoint program is active. At a command prompt, enter:

```
ps -ef | grep endpoint
```

If the endpoint program is running, it shows up with the following string in the right-most column of the output, “/opt/ixia/endpoint.”

Disabling Automatic Startup

To disable automatic startup, remove the /etc/rc2.d/S81 endpoint file.

Logging and Messages

While most error messages encountered on an endpoint are returned to the IxChariot or Qcheck Console, some may be logged to disk. Errors are saved in a file named `endpoint.log`, in the `/var/adm` directory. To view an error log, use the Ixia program named `FMTLOG`. `FMTLOG` reads from a binary log file, and writes its formatted output to `stdout`. Use the following `FMTLOG` command:

```
/opt/ixia/fmtlog log_filename >output_filename
```

The endpoint code does a lot of internal checking on itself. Our software captures details related to the problem in an ASCII text file named `assert.err` in the `/var/adm` directory. Save a copy of the file and send it to us via email for problem determination.

Known Problems

You might see some operating-system problems during streaming tests. With test scripts running at a very fast rate or with many pairs using small datagram buffer sizes, the operating system may lock up.

Specifically, we have seen lock-up problems with Solaris version 2.6 and later when running certain kinds of streaming tests. We ran a 35-pair IxChariot test in which each pair used the Voice over IP Send script (`Voips.scr`). This script specifies small buffers (40 bytes each) at 64 kbps. Running this test to a Sun Ultra 5 computer (as the Endpoint 2) caused Solaris to completely lock up; the computer did not respond to network, keyboard, or mouse input.

We determined that the Endpoint 2 computer was overwhelmed with thousands of small datagrams, which the TCP/IP network stack could not process quickly enough. Either the RAM (in our case, the computer had 64 MB of RAM) or CPU power needs to be increased to handle the load.

We’ve also seen a recurring problem with Sun Solaris x86, version 2.4. The endpoint may stop, and a core dump may occur during testing. We have traced this problem to a Solaris software bug, which is solved with the latest OS patch. Download the patch from one of the following Web sites:

<http://access1.sun.com/patch.public/cgi-bin/readme2html.cgi?patch=101946&type=rec>

http://access1.sun.com/patch.public/cgi-bin/show_list.cgi/rec/Solaris_Intel_2.4

Message CHR0181

You may receive message CHR0181 while running a test. If the error was detected at the Sun Solaris computer, it says that the endpoint program on Sun Solaris has run out of system semaphores. Each instance of Endpoint 1 requires a system semaphore. The maximum number of semaphores is not configurable on Sun Solaris; it is hard-coded to a large value. To avoid this problem, stop other programs that use semaphores or decrease the number of tests that use the computer as Endpoint 1.

Updates for Sun Solaris

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.

Sun posts code and driver updates directly to the following Web sites:

- www.sun.com/
- Anonymous FTP to <ftp://ftp.sun.com/>

18

Web-Based Performance Endpoint

This chapter describes the operation of the web-based Performance Endpoint.

Topics in this chapter:

- [*Web-Based Performance Endpoint Overview*](#) on page 18-1
- [*Running the Web-Based Endpoint*](#) on page 18-2
- [*Error Handling*](#) on page 18-3
- [*Compatibility with Other Endpoints*](#) on page 18-3
- [*Stopping the Web-Based Endpoint*](#) on page 18-4

Web-Based Performance Endpoint Overview

Ixia Performance Endpoints are lightweight software agents that allow for testing and monitoring of computers and computer networks. Endpoints are available for more than 20 operating systems and are continually updated to support new features in IxChariot and Qcheck.

Unlike the endpoints for all of the other supported platforms, the Web-Based Performance Endpoint was not designed to be installed on a computer. Users of the Web-Based endpoint can either run it from the World Wide Web or save it to a local hard disk, but as soon as they restart, or log out of, the computer where it is running, the endpoint stops running.

The Web-Based endpoint runs on the following 32-bit Windows operating systems:

- Windows NT
- Windows 2000
- Windows XP (32-bit version)
- Windows Server 2003 (32-bit version)
- Windows Vista (32-bit version).

While endpoints for other operating systems still run as long as the computer where they're installed is powered on, the Web-Based endpoint stops running as soon as the user logs out or restarts. Nothing has been written to the Registry on the computer where it ran.

The Web-Based endpoint supports most IxChariot and Qcheck functions. A few features are not supported. The following table summarizes the IxChariot and Qcheck features that are not supported:

Table 18-1. Unsupported Features

Function	Comment
APPC protocol	APPC is no longer supported, beginning with IxChariot 6.10.
SPX, IPX protocols	Not supported.
Endpoint.ini file	Default settings cannot be changed.
Application script datatypes (other than ZEROES or NOCOMPRESS)	IxChariot scripts that use a <code>send_datatype</code> parameter will fail.
Traceroute testing	Not supported.

Running the Web-Based Endpoint

To run the Web-Based endpoint on your local computer, use the Web browser on that computer to navigate to http://www.ixiacom.com/support/endpoint_library/. Click the link labeled **Web-Based Endpoint**.

Depending upon the browser or download utility you are using, you may have the option of running the software from its present location (the Ixia web site), as well as saving it to disk:

- Run from Location:

The "Run from Location" option downloads the Performance Endpoint to memory, but does not store a copy on your local disk. If the download is successful, you'll see a message stating that the endpoint has been started. Click **OK** to close the message.

There's nothing else you need to do. The endpoint is ready for testing with IxChariot or Qcheck.

- Save to Disk:

If you want to save the Performance Endpoint to disk, save it to the folder where you save your temporary files, such as Temp.

Starting the Endpoint

If you selected “Run from Location, the endpoint software starts automatically as soon as it is downloaded to memory.

If you saved the endpoint to your local disk, you need to manually start it using this procedure:

1. Navigate to the folder where you’ve saved the endpoint.
2. Double-click the file `endpoint.exe` to start the endpoint.

You will see a message stating that the endpoint has been started.

3. Click **OK** to close the message.

Note: If you are running the Performance Endpoint from the command line, you can use the `/nologo` switch to suppress the display of the pop-up window.

After you start the endpoint, there’s nothing else you need to do. The endpoint is ready for testing with IxChariot or Qcheck.

Restarting the Endpoint

When you save the endpoint to a local hard disk, it makes no difference where you save it. When you restart the computer, a copy of the executable `endpoint.exe` will still be on your hard drive, but it will no longer run until you restart it. Restart the executable by double-clicking it in the Windows Explorer.

Error Handling

Unlike endpoints for other operating systems, the Web-Based endpoint doesn’t log errors it encounters. However, it does report errors to IxChariot and Qcheck.

In the case of a connection failure or other failure during testing, the endpoint vanishes silently. The Console will notify you that it can no longer reach the endpoint. You should return to the Web and re-enable the endpoint in the case of such a failure.

Compatibility with Other Endpoints

The Web-Based endpoint cannot run on a computer where another endpoint is already running. For example, you cannot run the Web-Based endpoint on a computer where you have the 32-bit Windows Performance Endpoint installed and running. When you attempt to download it, you’ll receive an error message.

Correspondingly, if the Web-Based endpoint is running on a computer, you can install one of the conventional endpoints on that computer, but the conventional endpoint will not start running automatically once the installation completes. If the Web-Based endpoint executable is running, you must therefore either stop it before installing another endpoint, or else restart the computer after you complete the installation. The new endpoint will then start running automatically (and the Web-Based endpoint will no longer be present).

Stopping the Web-Based Endpoint

The Web-Based endpoint stops automatically as soon as you restart your computer or log out. However, you can also stop the endpoint manually.

To stop the Web-Based endpoint:

1. Click **Ctrl+Alt+Delete** to access the Windows Task Manager.
2. Click the **Processes** tab.
3. Highlight the process `endpoint.exe`. Click **End Process** to stop the endpoint and remove it from your computer.



Archived Endpoint Specifications

This appendix describes the IxChariot Performance Endpoints that have been archived. It contains the following topics:

- [Operating System and Protocol Stack Support](#) on page A-1
- [Performance Endpoint Support for IxChariot Functions](#) on page A-3
- [Endpoint Computer Resource Guidelines](#) on page A-4

Operating System and Protocol Stack Support

[Table A-1](#) identifies the supported operating system and protocol stack software for the archived endpoints. The table lists the software with which we have tested the archived Performance Endpoints for each operating system.

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

Table A-1. Archived Endpoints - Operating System Compatibility

Archived Endpoint	OS version	TCP, UDP, RTP	IP Multicast version	IPX/SPX stack
Cobalt RaQ/RaQ2 (MIPS)	Linux v. 2.0 for MIPS	included	kernel 2.0.32	no
Cobalt RaQ3 (x86)	kernel 2.0.32	included	kernel 2.0.32	no
Compaq Tru64 UNIX	Digital UNIX 4.0B or Compaq Tru64 Unix for Alpha	included	v4.0B	no
FreeBSD UNIX	BSD v3.1	included	v3.1	no
IBM MVS	MVS/ESA SP v4R2.2	See " MVS TCP/IP Stacks " on page A-3"	no	no

Table A-1. Archived Endpoints - Operating System Compatibility (Continued)

Archived Endpoint	OS version	TCP, UDP, RTP	IP Multicast version	IPX/SPX stack
IBM OS/2	OS/2 Warp 4, Warp Connect 3	Download TCP 4.1	Download TCP 4.1	Download Novell Netware Client v2.12
Linux IA-64 (TurboLinux)	kernel 2.4.0test7-42	included	kernel 2.4.0test7-42	no
Microsoft Windows 3.1	Windows 3.1 or Windows for Workgroups 3.11	see " Microsoft Windows 3.1 TCP/IP Stacks on page A-2"	Chameleon 7.0, as E2	no
Microsoft Windows 95	Windows 95	included	no	Download Novell Netware Client v3.21
Microsoft Windows 95 with WinSock 2	Windows 95 with WinSock 2 installed	Download WinSock 2	included	included
Microsoft Windows 98	Windows 98	included	included	included
Microsoft Windows Millennium Edition (Me)	Windows Me	included	included	included
Microsoft Windows NT 4 for Alpha	Windows NT4 SP 3	included	SP3 (IGMPv1) SP4 (IGMPv2)	included
Microsoft Windows XP 64-bit Edition (IA-64 processors)	Windows XP (64-bit)	included	included	no
Novell NetWare	v3.12	included	v4.0	included
SCO UnixWare	UnixWare v2.1	included	v7.0	no
SGI IRIX	IRIX v6.2 with patches	included	v6.2	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 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 IxChariot.
- SOLVE:TCPaccess versions 4.1 and 5.2 stack from Sterling Software. A set of PTFs is required for operation with version 4.1.

Performance Endpoint Support for IxChariot Functions

The following table describes the capabilities of the archived Performance Endpoints. These endpoints may not support new functionality in the latest versions of IxChariot.

Table A-2. Archived Performance Endpoint Capabilities per OS

Endpoint OS	IP QoS (DiffServ, GQoS, TOS)	Trace-route	CPU Utilization	VoIP Tests	Video Pair Tests	IPv6 Tests	802.11 Statistics
Cobalt RaQ or RaQ2 (MIPS)	TOS	No	Yes	No	No	No	No
Cobalt RaQ3 (x86)	TOS	Yes	Yes	Yes	No	No	No
Compaq Tru64 UNIX	TOS	No	Yes	No	No	No	No
FreeBSD UNIX	TOS	No	Yes	No	No	No	No
IBM MVS	No	No	No	No	No	No	No
IBM OS/2	TOS	No	Yes	No	No	No	No
Linux IA-64 (TurboLinux)	TOS	Yes	Yes	No	Yes	No	No
Microsoft Windows 3.1	No	No	No	No	No	No	No
Microsoft Windows 95	No	No	Yes	No	No	No	No
Microsoft Windows 95 with WinSock 2	TOS (UDP, RTP)	Yes	Yes	No	No	No	No
Microsoft Windows 98	GQoS (RSVP), TOS (UDP, RTP)	Yes	Yes	Yes	No	No	No
Microsoft Windows Me	GQoS (RSVP)	Yes	Yes	Yes	No	No	No

Table A-2. Archived Performance Endpoint Capabilities per OS (Continued)

Endpoint OS	IP QoS (DiffServ, GQoS, TOS)	Trace- route	CPU Utiliti- zation	VoIP Tests	Video Pair Tests	IPv6 Tests	802.11 Statistics
Microsoft Windows NT 4 for Alpha	No	Yes	Yes	No	No	No	No
Microsoft Windows 98 (Web-Based)	Yes	No	Yes	Yes	No	No	No
Microsoft Windows XP 64-bit Edition (IA-64)	DiffServ, GQoS, TOS	No	No	Yes	Yes	No	No
Novell NetWare	No	No	No, v3.12; Yes, v4.0	No	No	No	No
SCO UnixWare	TOS (bits 3-5)	No	No	No	No	No	No
SGI IRIX	TOS	No	Yes	No	No	No	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 IxChariot. The type of information you need depends upon your usage. The following topics provide recommended endpoint computer specifications according to different testing scenarios.

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 (“n/a” indicates that the protocol is not supported by the endpoint). The shaded rows list archived endpoints.

Table A-3. Calculating Memory Requirements

Operating System	Base RAM (in KB)	TCP KB/ pair	UDP KB/ pair	RTP KB/ pair	SPX KB/ pair	IPX KB/ pair
MVS	666	25-48	24-52	24-52	n/a	n/a
NetWare	1100	80-110	320-340	320-340	70-100	260-280
OS/2	1096	50-65	150-170	150-170	315-340	150-170
Windows 3.1	550	72-600	72-600	72-600	n/a	n/a
Windows 95/98/Me	1100	40-65	100-145	100-145	40-65	55-75

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 *IxChariot 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. The shaded rows list archived endpoints.

Table A-4. Endpoint Pair Capacity

Operating System	Installed RAM	TCP pairs	RTP or UDP pairs	SPX pairs	IPX pairs
NetWare 4.12	64 MB	500	200	100	100
OS/2 4.0	32 MB	500	200	20	20
Windows 3.1	8 MB	1	1	n/a	n/a
Windows 95/98/Me	16 MB	18	100	40	175
Win64 (Itanium based)	768 MB	175	120	n/a	n/a

Notes:

- 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.

Index

Numerics

1600T, Ixia chassis [4-4](#)
250, Ixia chassis [4-4](#)
400T, Ixia chassis [4-4](#)

A

AIX [6-1](#)
ALLOW keyword [3-3](#)
Apple Macintosh [13-1](#)
archived endpoints [1-1](#)
AUDIT_FILENAME keyword [3-5](#)
automatic upgrade
 self-install [3-11](#)

B

big endian [9-2](#)
buffer size, socket [3-7](#), [3-8](#)

C

calculating memory requirements [2-6](#), [A-4](#)
capacities of endpoints [2-7](#), [A-5](#)
CE, *See* Windows CE endpoint
cleanup after errors [5-9](#)
 HP-UX endpoint [5-9](#)
 IBM AIX endpoint [6-8](#)
 Linux 32-bit x86 (RPM) endpoint [7-8](#)
 Linux-OpenWrt endpoint [11-5](#)
 Mac OS X endpoint [13-2](#)
 Sun Solaris endpoint [17-9](#)
clock synchronizaton [3-8](#)
CMPFILES directory [14-8](#)
 HP-UX [5-5](#)
 IBM AIX [6-4](#)

Linux 32-bit x86 [7-7](#)
Mac OS X [13-2](#)
Mac OS X endpoint [13-2](#)
Sun Solaris [17-6](#)
Windows 32-bit [14-8](#)

E

ENABLE_PROTOCOL keyword [3-6](#)
endianness [9-2](#)
endpoint capabilities
 IxChariot [2-3](#), [2-4](#), [A-3](#)
endpoint capacities [2-7](#), [A-5](#)
endpoint initialization file [3-1](#)
 default keywords [3-1](#)
 keywords [3-3](#)
endpoint versions [1-1](#)
endpoint.aud [3-5](#)
endpoint.console [5-8](#), [6-7](#)
endpoint.ini [3-1](#)
 ALLOW keyword [3-3](#)
 AUDIT_FILENAME keyword [3-5](#)
 ENABLE_PROTOCOL keyword [3-6](#)
 FORCE_CLOCKSYNC keyword [3-8](#)
 MANAGEMENT_PORT keyword [3-10](#)
 MAX_PAYLOAD_DISK_USAGE keyword [3-9](#)
 MAX_PAYLOAD_MEMORY_USAGE
 keyword [3-9](#)
 PAYLOAD_MEMORY_LIMIT_USAGE
 keyword [3-9](#)
 SECURITY_AUDITING keyword [3-4](#)
 SOCKET_RECEIVE_BUFFER_SIZE keyword [3-8](#)
 SOCKET_SEND_BUFFER_SIZE keyword [3-7](#)
endpoint.log [6-10](#)
error messages

- Windows CE endpoints 15-8
- F**
- failed assertion
 - HP-UX endpoint 5-9
 - IBM AIX endpoint 6-10
 - Linux 32-bit x86 endpoint 7-13
 - Mac OS X endpoint 13-5
 - Sun Solaris endpoint 17-10
 - Windows 32-bit endpoint 14-16
 - Windows CE endpoint 15-8
- FORCE_CLOCKSYNC keyword 3-8
- G**
- gsendw32.exe 3-10
- H**
- HP-UX endpoint 5-1
 - cleanup after errors 5-9
 - configuring 5-6
 - core dump 5-10
 - determining IP network address 5-7
 - disabling automatic startup 5-9
 - installing 5-1, 5-2, 5-5
 - messages 5-9, 5-10
 - running 5-8
 - safestore files 5-6
 - starting 5-8
 - stopping 5-8
 - support 5-10
 - TCP/IP 5-6, 5-7
 - unattended installation 5-5
 - uninstall 5-6
- I**
- IBM AIX endpoint 6-1
 - cleanup 6-8
 - configuring 6-6
 - determining if active 6-8
 - disabling automatic startup 6-9
 - installing 6-1, 6-2, 6-4
 - messages 6-10
 - README 6-4
 - removing 6-5
 - running 6-7
 - starting 6-7
 - stopping 6-8
 - support 6-10
 - TCP 6-6
- installation requirements 14-2
 - HP-UX endpoint 5-1
 - IBM AIX endpoint 6-1
 - Linux 32-bit x86 endpoint 7-2
 - Linux on Lexra endpoint 10-1
 - Linux PowerPC endpoint 12-1
 - Linux x86-64 endpoint 8-1
 - Linux-ARM endpoint 9-2
 - Linux-OpenWrt endpoint 11-2
 - Mac OS X endpoint 13-1
 - Sun Solaris endpoint 17-1
 - Windows 32-bit endpoint 14-2
 - Windows 64-bit 16-2
- installation, silent mode 14-8, 16-5
- installing
 - HP-UX endpoint 5-2
 - IBM AIX endpoints 6-2
 - Linux 32-bit x86 endpoint 7-5
 - Linux 64-bit endpoint 8-2
 - Linux on Lexra endpoint 10-1
 - Linux-ARM endpoint 9-2
 - Linux-OpenWrt endpoint 11-2
 - Mac OS X endpoint 13-1
 - Sun Solaris endpoints 17-2
 - Windows 32-bit endpoint 14-3
 - Windows 64-bit 16-3
 - Windows CE endpoints 15-3, 15-4
- IPKG distribution 11-1, 11-2, 11-3
- IPv6 support 2-4, 7-2, 8-2
- IPv6 Test Module
 - support for 2-4
- Ixia Performance Endpoint
 - about 4-1
 - logging 4-2
 - messages 4-2
 - starting and stopping 4-2
 - supported load modules 4-4
- L**
- Linksys WRT54G 11-1
- Linux 32-bit x86 endpoint 7-1
 - cleanup 7-12
 - configuring 7-9, 7-10
 - determining if active 7-12
 - endpoint.ini 7-7
 - installing 7-3, 7-5, 7-7
 - logging 7-13
 - running 7-11, 7-12, 7-13
 - starting 7-11
 - stopping 7-12
 - supported processors 7-1
 - uninstall 7-8
 - uninstalling 7-8
- Linux on Lexra endpoint 10-1
 - cleanup 10-5
 - configuring 10-2, 10-3
 - installation requirements 10-1
 - installing 10-2
 - running 10-4

- stopping 10-4
- Linux PowerPC endpoint 12-1
 - configuration 12-3
 - installation 12-1
 - starting 12-4
 - stopping 12-4
- Linux x86-64 endpoint 8-1
 - autostarting 8-8
 - cleanup 8-10
 - configuring 8-7, 8-8
 - disabling automatic startup 8-10
 - increasing number concurrent connections 8-10
 - installation requirements 8-1
 - installing 8-2, 8-4, 8-5
 - logging and messages 8-10, 8-11
 - running 8-8
 - starting 8-9
 - stopping 8-9
 - uninstalling 8-6
- Linux-ARM endpoint 9-1
 - configuration 9-3
 - endianness 9-2
 - installation 9-2
- Linux-OpenWrt endpoint 11-1
 - installing 11-2
 - removing 11-3
 - sockets interface support 11-3
 - starting 11-4
 - stopping 11-5
 - supported protocols 11-4
- little endian 9-2, 11-2

M

- Mac OS X endpoint 13-1
 - cleanup 13-5
 - configuring 13-3
 - determining if active 13-5
 - determining IP network address 13-3
 - endpoint.ini 13-2
 - installing 13-2
 - logging 13-5
 - running 13-4
 - stopping 13-5
 - support 13-6
 - supported processors 13-1
 - uninstall 13-2
- messages
 - HP-UX endpoint 5-9
 - IBM AIX endpoint 6-10
 - Linux 32-bit x86 endpoint 7-13
 - Mac OS X endpoint 13-5
 - Sun Solaris endpoint 17-10
 - Windows 32-bit endpoint 14-16
 - Windows CE endpoint 15-8
- Microsoft overlapped I/O 2-6

- Microsoft SNA Server 14-16
 - for Windows NT 14-2
- MIPS 11-1, 11-2
- MSS Option 2-4, 6-7

O

- Optixia 4-4

P

- PFS 5-2
- PKGADD command (Solaris) 17-5
- port number, management port 3-10, 5-7, 6-7, 7-9, 8-8, 9-4, 10-3, 11-4, 13-4, 14-10, 15-6, 16-8, 17-8
- Portable File System 5-2
- PowerPC, *See* Linux PowerPC Endpoint. 12-1

R

- Red Hat
 - autostart commands 7-10
 - disabling automatic startup 7-12
 - logging and messages 7-13
- response file 3-11
- restricting access to endpoints 3-3
- RISC System 6-1
- RPM
 - determining IP network address 7-9
 - endpoint installation for Linux 32-bit x86 7-5
 - removing endpoint 7-8

S

- SECURITY_AUDITING keyword 3-4
- SetAddr utility 14-13
- silent mode
 - install 14-8, 16-5
 - uninstall 14-9, 16-6
- Slackware
 - logging and messages 7-13
- SNA Server 14-16
 - for Windows NT 14-2
- SOCKET_RECEIVE_BUFFER_SIZE keyword 3-8
- SOCKET_SEND_BUFFER_SIZE keyword 3-7
- software requirements
 - protocol support 2-2, A-1
- SPX II
 - support on Windows NT 14-2
- Stack Manager 4-1
- Sun Solaris endpoint 17-1, 17-10
 - cleanup 17-9
 - configuring 17-7, 17-8
 - core dump 17-10

- determining if active 17-10
 - installing 17-1, 17-2, 17-5
 - IP network address 17-7
 - messages 17-10, 17-11
 - OS lockup 17-10
 - removing 17-7
 - running 17-8, 17-10
 - starting 17-9
 - stopping 17-9
 - support for OS 5-10
 - HP-UX 5-10
 - IBM AIX 6-10
 - Mac OS X 13-6
 - Sun Solaris 17-11
 - Windows 32-bit 14-16
- T**
- TAR
 - determining IP network address 7-9
 - endpoint installation for Linux 32-bit x86 7-3
 - unattended Linux 32-bit x86 install 7-5
 - TcpWindowSize 2-6
 - throughput 2-6
 - generating maximum 2-6
- U**
- uninstall
 - HP-UX endpoint 5-6
 - IBM AIX 6-5
 - Linux 32-bit x86 (RPM) endpoint 7-8
 - Linux-OpenWrt endpoint 11-3
 - Mac OS X endpoint 13-2
 - Sun Solaris 17-7
 - Windows 32-bit endpoint 14-9
 - Windows 64-bit endpoint 16-5
 - Windows CE endpoints 15-5
 - uninstall, silent mode 14-9, 16-6
- V**
- version
 - of endpoint 1-1
 - virtual addresses in Windows 14-13
 - VoIP Test Module
 - support for 2-3, 2-4, A-3
- W**
- Web-Based endpoint 18-1
 - compatibility with other endpoints 18-3
 - downloading 18-2
 - error handling 18-3
 - features not supported 18-2
 - restarting 18-3
 - running 18-2
 - running from location 18-2
 - saving 18-2
 - supported OSs 18-1
 - WhiteRussian 11-1
 - Windows 32-bit endpoint 14-1
 - configuring 14-10
 - disabling automatic startup 14-15
 - installing 14-3
 - IP address 14-12
 - IPX address 14-11
 - IPX/SPX 14-10
 - messages 14-16
 - running 14-13
 - SetAddr utility 14-13
 - starting 14-13
 - stopping 14-13
 - support 14-16
 - support for OS 14-16
 - TCP/IP 14-11, 14-12
 - uninstall 14-9
 - Windows 64-bit endpoint
 - configuring 16-6
 - disabling automatic startup 16-9
 - installing 16-3
 - IP address 16-7
 - logging 16-12
 - messages 16-12
 - SetAddr utility 16-9
 - starting 16-8
 - stopping 16-9
 - TCP/IP 16-6
 - uninstalling 16-5, 16-6
 - Windows Administrator permissions 14-4
 - Windows CE endpoint
 - error messages 15-8
 - installation requirements 15-2
 - installing 15-3, 15-4
 - IP address 15-5
 - limitations 15-8
 - messages 15-8
 - Microsoft Windows Mobile , support for 15-2
 - Performance Endpoints, list of 15-1
 - running 15-6
 - starting 15-6, 15-7
 - stopping 15-7
 - TCP and UDP 15-5
 - uninstall 15-5
 - Windows Mobile 15-2