IXIA®

IxChariot® Performance Endpoints

Release 7.10

913-0951 Rev. A
December 2009

Real World Traffic™
SOFTWARE

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 +1 818 595 2599 For the online support form, go to: http://www.ixiacom.com/support/inquiry/
EMEA	**Ixia Europe Limited** One Globeside, Fieldhouse Lane Marlow, SL7 1HZ United Kingdom +44 1628 405750 FAX +44 1628 405790 salesemea@ixiacom.com	**Support:** eurosupport@ixiacom.com +44 1628 405797 For the online support form, go to: http://www.ixiacom.com/support/inquiry/?location=emea
Asia Pacific	**Ixia Pte Ltd** 210 Middle Road #08-01 IOI Plaza Singapore 188994	**Support:** Support-AsiaPac@ixiacom.com +65 6332125 For the online support form, go to: http://www.ixiacom.com/support/inquiry/
Japan	**Ixia KK** Aioi Sampo Shinjuku Building, 16th Floor 3-25-3 Yoyogi Shibuya-Ku Tokyo 151-0053 Japan	**Support:** Support-Japan@ixiacom.com +81 3 5365 4690 For the online support form, go to: http://www.ixiacom.com/support/inquiry/
India	**Ixia Technologies Pvt Ltd** 2nd Floor, 19/1, Vithall Malya Road, Bangalore 560 001 India	**Support:** Support-India@ixiacom.com +91 80 22161000 For the online support form, go to: http://www.ixiacom.com/support/inquiry/?location=india

913-0951 Rev. A
December 14, 2009

Table of Contents

Chapter 1 IxChariot Performance Endpoints Overview

Chapter 2 Performance Endpoint Specifications

Chapter 3 Endpoint Initialization File

Chapter 4 Ixia Load Module

Chapter 5 Android

Chapter 6 iPhone 3G/iPod Touch

Chapter 8 Linux 64-Bit (x86-64)

Chapter 9 Linux 64-Bit (Itanium)

Chapter 10 Linux on ARM Processors

Chapter 11 Linux on Lexra

Chapter 12 Linux on OpenWrt (MIPS Platforms)

Chapter 13 Mac OS X

Chapter 14 Microsoft Windows 32-Bit

Chapter 15 Microsoft Windows 64-Bit

Chapter 17 Microsoft Windows CE 5.0 and Windows Mobile 5.0 and 6.0

Chapter 18 Microsoft Windows Embedded CE 6.0

Chapter 19 Sun Solaris

Chapter 20 Web-Based Performance Endpoint

1 IxChariot Performance Endpoints Overview

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

All the information you need to install, configure, and run the Performance Endpoints in your network is included in this guide. In addition to topics discussing issues common to all the endpoints, this guide also contain 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 1-2
- *Displaying Endpoint Configuration Settings* on page 1-2
- *List of Performance Endpoints* on page 1-2
- *What's New in IxChariot 7.10?* on page 1-4

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 instructs provides 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.

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.

List of Performance Endpoints

Table 1-1provides a listing of all the current active Performance Endpoint, organized by operating system and platform.

Table 1-1. Listing of Current Performance Endpoints

Performance Endpoint for …	See …
Android	Chapter 5, *Android*
Apple Mac OS X	Chapter 13, *Mac OS X*
iPhone 3G and iPod Touch	Chapter 6, *iPhone 3G/iPod Touch*
Linux / ARM (µClibC)	Chapter 10, *Linux on ARM Processors*
Linux / ARM (Big Endian)	Chapter 10, *Linux on ARM Processors*
Linux / ARM (Little Endian)	Chapter 10, *Linux on ARM Processors*
Linux / ARM (Statically-linked, Little-Endian)	Chapter 10, *Linux on ARM Processors*
Linux Lexra	Chapter 11, *Linux on Lexra*
Linux x86 / 32-Bit (i386)	Chapter 7, *Linux 32-Bit (x86)*
Linux x86 / 32-Bit (i386) RPM distribution	Chapter 7, *Linux 32-Bit (x86)*
Linux x86 / 64-Bit (x86_64)	Chapter 8, *Linux 64-Bit (x86-64)*
Linux x86 / 64-Bit (x86_64) RPM distribution	Chapter 8, *Linux 64-Bit (x86-64)*

Table 1-1. Listing of Current Performance Endpoints

Performance Endpoint for …	**See …**
Linux IA-64 (x86_64)	Chapter 9, *Linux 64-Bit (Itanium)*
Linux IA-64 (x86_64) RPM distributions	Chapter 9, *Linux 64-Bit (Itanium)*
Microsoft Windows 2000/2003/XP	Chapter 14, *Microsoft Windows 32-Bit*
Microsoft Windows 2000/2003/XP - x86 / 64-Bit	Chapter 15, *Microsoft Windows 64-Bit*
Microsoft Windows 2000/XP - web-based	Chapter 20, *Web-Based Performance Endpoint*
Microsoft Windows 2008 Server, 32-bit	Chapter 14, *Microsoft Windows 32-Bit*
Microsoft Windows 2008 Server, 64-bit	Chapter 15, *Microsoft Windows 64-Bit*
Microsoft Windows 2008 Server R2 for Itanium-Based Systems	Chapter 16, *Microsoft Windows Server 2008 R2 Itanium-Based System*
Microsoft Windows 7 Beta 32-Bit	Chapter 14, *Microsoft Windows 32-Bit*
Microsoft Windows 7 Beta 64-Bit	Chapter 15, *Microsoft Windows 64-Bit*
Microsoft Windows CE 5.0 / Mobile 5.0 / Mobile 6.0 (QoS support) for ARM	Chapter 17, *Microsoft Windows CE 5.0 and Windows Mobile 5.0 and 6.0*
Microsoft Windows CE 5.0 / Mobile 5.0 / Mobile 6.0 (QoS support) for ARM (Win CE Version)	Chapter 17, *Microsoft Windows CE 5.0 and Windows Mobile 5.0 and 6.0*
Microsoft Windows Embedded CE 6.0	Chapter 18, *Microsoft Windows Embedded CE 6.0*
Microsoft Windows Vista	Chapter 14, *Microsoft Windows 32-Bit*
Microsoft Windows Vista - x86 / 64-Bit	Chapter 15, *Microsoft Windows 64-Bit*
Open WRT Operating System (Linux/MIPS) - ipk image	Chapter 12, *Linux on OpenWrt (MIPS Platforms)*
Open WRT Operating System (Linux/MIPS) - tar image	Chapter 12, *Linux on OpenWrt (MIPS Platforms)*
Sun Solaris for SPARC	Chapter 19, *Sun Solaris*
Sun Solaris for x86 32-Bit	Chapter 19, *Sun Solaris*

What’s New in IxChariot 7.10?

IxChariot 7.10 includes the following changes to the Performance Endpoints library:

- The Mac OS X Performance Endpoint is now a universal binary application, supporting both PowerPC and Intel
- New Intel Intanium Architecture (IA-64) Performance Endpoints are available for Windows Server 2008 R2 and Linux Redhat Enterprise 5 (Itanium edition).

Refer to the *IxChariot User Guide* for a description of other new features and enhancements in release 7.10.

2 Performance Endpoint Specifications

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

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/endpoint_library.

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
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
Microsoft Windows 2000	Windows 2000	included	included	included
Microsoft Windows 7 (32-bit and 64-bit editions)	Windows 7	included	included	no
Microsoft Windows CE 5.0, Windows Mobile 5.0 and 6.0	Windows CE 5.0, Windows Mobile 5.0, Windows Mobile 6.0	included	included	no
Microsoft Windows Embedded CE 6.0	Windows Embedded CE 6.0	included	included	no
Microsoft Windows NT	Windows NT, SP4	included	SP3 (IGMPv1) SP4 (IGMPv2)	included
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
Microsoft Windows Server 2008 (32-bit and 64-bit editions)	Windows Server 2008	included	included	no

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

Endpoint	OS version	TCP, UDP, RTP	IP Multicast version	IPX/SPX stack
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 2008 R2	Microsoft Windows Server 2008 R2	included	included	no
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 2-5 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	IPTV Tests	IPv6 Tests	802.11 Statistics
Apple Macintosh OS X (G4 and G5 processors)	No	Yes	Yes	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
Microsoft Windows 2000	Yes	Yes	Yes	Yes	No	Yes[b]	Yes
Microsoft Windows 7	Yes	Yes	Yes	Yes	No	Yes[b]	Yes
Microsoft Windows CE 5.0, Windows Mobile 5.0 and 6.0	No	No[c]	Yes	Yes	No	No	Yes
Microsoft Windows Embedded CE 6.0	No	No[c]	Yes	Yes	No	No	Yes
Microsoft Windows NT 4	Yes	Yes	Yes	Yes	No	No	No

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

Endpoint OS	Trace-route	CPU Utilization	VoIP Tests	Video Pair Tests	IPTV Tests	IPv6 Tests	802.11 Statistics
Microsoft Windows Server 2003	Yes	Yes	Yes	Yes	No	Yes	Yes
Microsoft Windows Server 2008	Yes	Yes	Yes	Yes	No	Yes	Yes
Microsoft Windows Vista	Yes	Yes	Yes	Yes	No	Yes[b]	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 Server 2008 R2	Yes	No	Yes	Yes	No	Yes	No
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.

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 7, 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:

- 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
Apple Macintosh	**Yes**	No	No
IxOS Load Module	**Yes**	**Yes**	No
Linux (all)	**Yes**	**Yes**[a]	No
UNIX (all)	**Yes**	**Yes**	No
Microsoft Windows NT	**Yes**	No	No
Microsoft Windows 2000	**Yes**	**Yes**	**Yes**
Microsoft Windows Server 2003 (32-bit and 64-bit editions)	**Yes**	**Yes**	**Yes**
Microsoft Windows Server 2008 (32-bit and 64-bit editions)	**Yes**	**Yes**	**Yes**
Microsoft Windows XP (32-bit and 64-bit editions)	**Yes**	**Yes**	**Yes**
Microsoft Windows Vista (32-bit and 64-bit editions)[b]	**Yes**	**Yes**	**Yes**
Microsoft Windows 7 (32-bit and 64-bit editions)[c]	**Yes**	**Yes**	**Yes**
Microsoft Windows CE 5.0	No	**Yes**	No
Microsoft Windows Embedded CE 6.0	No	**Yes**	No

[a] integrated into 2.4 kernel.
[b] supported via qWave.
[c] supported via qWave.

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 2-8
- *IxOS Endpoint Pair Capacities* on page 2-9

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
Ixia Load Module	1320	57-74	89-105	65-84	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
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 5.0, Windows Mobile 5.0 and 6.0	277	44-70	196-436	700-773	n/a	n/a
Windows Embedded CE 6.0	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 7 (32-bit)	3500	203-356	240-396	205-340	n/a	n/a
Windows 7 (64-bit)	4700	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
Windows Server 2008 32-bit Edition	3500	203-356	240-396	205-340	n/a	n/a
Windows Server 2008 64-bit Edition	4700	203-356	240-396	205-340	n/a	n/a

These RAM usage numbers represent sending with the variable `send_datatype` set to `ZEROS`. Other `send_datatypes` require memory buffers roughly equivalent to the disk space of the `.cmp` file being used. Add 2 KBytes when using `send_datatype = NOCOMPRESS`. See the *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
IxOS	Refer to *IxOS Endpoint Pair Capacities* on page 2-9.				
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
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 5.0, Windows Mobile 5.0 and 6.0	56 MB	85	35	n/a	n/a
Windows Embedded CE 6.0	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 7	1 GB	3000	1000	n/a	n/a
Windows XP x64 Edition	768 MB	175	120	n/a	n/a
Windows Server 2003	1 GB	3000	1000	n/a	n/a
Windows Server 2008	1 GB	3000	1000	n/a	n/a

IxOS Endpoint Pair Capacities

Table 2-7 lists the maximum number of pairs supported by the IxOS Performance Endpoint running on various load modules.

Table 2-7. Ixia Load Module Pair Capacity - Maximum Pairs Supported

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
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
XMV16 and XMV12X[b]	1 GB	4,500	1,100[c]	n/a

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

[b] The maximum number of VoIP unidirectional pairs is 250, the maximum number of VoIP bidirectional pairs is 150, the maximum number of video pairs is 50, the maximum number of IPTV pairs is 65, and the maximum TCP throughput is 932.866 Mbps.

[c] The maximum number of UDP and RTP streaming pairs is 1,100. The maximum number of reliable UDP pairs is 200.

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 2-8. Endpoint Compatibility

	Ixia Product	
Performance Endpoint	Qcheck	IxChariot
Apple Macintosh (32-bit)	Yes	Yes
Ixia Load Module	Yes	Yes
Linux 32-bit (x86) – TAR	Yes	Yes

Table 2-8. Endpoint Compatibility (Continued)

	Ixia Product	
Performance Endpoint	**Qcheck**	**IxChariot**
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	Yes	Yes
Microsoft Windows CE	Yes	Yes
Microsoft Windows 64-Bit	Yes	Yes
Web-Based Performance Endpoint	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

Initialization File Description

An endpoint initialization file (endpoint.ini) 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).
- Specify the QoS TOS value that will be used by the endpoint on the management network until the first management command is received and executed.
- Sets values for the Ixia Discovery Server automatic registration process.
- Set the timeout value for sending timing records to the IxChariot Console.

endpoint.ini is an editable text file. There is a separate copy for each operating system.

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           10485760
USE_ENCRYPTION                       OFF
SOCKET_SEND_BUFFER_SIZE              0
SOCKET_RECEIVE_BUFFER_SIZE           0
FORCE_CLOCKSYNC                      1
MANAGEMENT_PORT                      80
INITIAL_MANAGEMENT_TOS               0
DISABLE_DISCOVERY                    OFF
DISCOVERY_SERVER_ADDRESS             IxDiscoveryServ
DISCOVERY_SERVER_PORT                10120
DISCOVERY_INTERVAL_BETWEEN_CONNECTS  0
REPORTING_TIMEOUT                    0
;-------------------------------------------------
```

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
INITIAL_MANAGEMENT_TOS	0
DISABLE_DISCOVERY	OFF
DISCOVERY_SERVER_ADDRESS	IxDiscoveryServ

Table 3-1. Endpoint Initialization File Defaults (Continued)

Keyword	Default Value
DISCOVERY_SERVER_PORT	10120
DISCOVERY_INTERVAL_BETWEEN_CONNECTS	0
REPORTING_TIMEOUT	0

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
- *INITIAL_MANAGEMENT_TOS* on page 3-10
- *DISABLE_DISCOVERY* on page 3-11
- *DISCOVERY_SERVER_ADDRESS* on page 3-11
- *DISCOVERY_SERVER_PORT* on page 3-11
- *DISCOVERY_INTERVAL_BETWEEN_CONNECTS* on page 3-12
- *REPORTING_TIMEOUT* on page 3-12

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 `ALLOW` address check are logged.
REJECTED	Only access attempts that failed the `ALLOW` address check are logged.
ALL	Both passed and rejected access attempts are logged.

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

AUDIT_FILENAME

This keyword specifies the filespec for the audit file. See *SECURITY_AUDITING* on page 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 `ALLOW` entries, this is the value to use for the protocol address.
Security Result	Whether this `SECURITY_AUDITING` “passed” or was “rejected.” If this is an entry for an “Ended” action, this field is reported as “n/a.”
Endpoint Partner Protocol	The network protocol used to run the test with our partner endpoint.
Endpoint Partner Address	The network address of our partner endpoint.

ENABLE_ PROTOCOL

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

```
ALL
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
```

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.

MANAGEMENT_ PORT

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.

INITIAL_ MANAGEMENT_ TOS

This keyword specifies the QoS TOS value that will be used by the endpoint on the management network until the first management command is received and executed.

Syntax:

```
INITIAL_MANAGEMENT_TOS <tos>
```

Example:

```
INITIAL_MANAGEMENT_TOS 32
```

Notes:

- <tos> must be a decimal number between 0 and 255.
- Any invalid <tos> value will be reset to 0.

Refer to the *IxChariot User Guide* for instructions for detailed information about setting QoS values for IxChariot management traffic.

DISABLE_ DISCOVERY

This keyword specifies whether the automatic endpoint registration with the Ixia Discovery Server should be disabled.

Syntax:

```
DISABLE_DISCOVERY <ON or OFF>
```

Example:

```
DISABLE_DISCOVERY ON
```

Notes:

- The default value is OFF, meaning that the automatic endpoint registration with the Ixia Discovery Server is enabled (that is, it is not disabled).
- This feature is supported by all of the Microsoft Windows performance endpoints (including Windows CE) and all of the Linux performance endpoints. None of the other performance endpoints support the feature.
- To disable the feature on a supported platform, un-comment the line and set it to ON.
- To enable the feature on a supported platform, simply leave the statement commented-out, or the value to OFF.
- For the unsupported platforms, leave this option commented-out.

DISCOVERY_ SERVER_ ADDRESS

This keyword specifies the Ixia Discovery Server IP address or name used in the automatic registration process.

Syntax:

```
DISCOVERY_SERVER_ADDRESS <address>
```

Example:

```
DISCOVERY_SERVER_ADDRESS IxDiscoveryServ
```

Notes:

- This feature is supported by all of the Microsoft Windows performance endpoints (including Windows CE) and all of the Linux performance endpoints. None of the other performance endpoints support the feature.
- For supported platforms, un-comment the line and set it to the desired address.
- For the unsupported platforms, leave this option commented-out.

DISCOVERY_ SERVER_PORT

This keyword specifies the port that the Ixia Discovery Server uses in the automatic registration process.

Syntax:

```
DISCOVERY_SERVER_PORT <port>
```

Example:

```
DISCOVERY_SERVER_PORT 10120
```

Notes:

- This feature is supported by all of the Microsoft Windows performance endpoints (including Windows CE) and all of the Linux performance endpoints. None of the other performance endpoints support the feature.
- For supported platforms, un-comment the line and specify the desired port number.
- For the unsupported platforms, leave this option commented-out.

DISCOVERY_INTERVAL_BETWEEN_CONNECTS

This keyword specifies the interval (in seconds) at which the endpoint will try to connect to the Ixia Discovery Server used in the automatic registration process.

Syntax:

```
DISCOVERY_INTERVAL_BETWEEN_CONNECTS <interval>
```

Example:

```
DISCOVERY_INTERVAL_BETWEEN_CONNECTS 30
```

Notes:

- The default value is 0, which means that the endpoint will try to connect only once.
- If the value is greater than 0, the endpoint will try to connect at the specified interval until the connection is successful.
- The maximum value is 2,000,000.
- This feature is supported by all of the Microsoft Windows performance endpoints (including Windows CE) and all of the Linux performance endpoints. None of the other performance endpoints support the feature.
- For supported platforms, un-comment the line and specify the desired value.
- For the unsupported platforms, leave this option commented-out.

REPORTING_TIMEOUT

This keyword specifies the interval (in seconds) after which the endpoint will abandon efforts to connect to its partner to send the timing record reports.

The valid range of values is from zero through 999,999 seconds. If the value is set to zero, IxChariot will use the default timeout interval (130 seconds).

Syntax:

```
REPORTING_TIMEOUT <interval>
```

Example:

```
REPORTING_TIMEOUT 180
```

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
- *Optimization* on page 4-3
- *Supported Load Modules* on page 4-3

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

Refer to the Release Notes for a list of the Ixia load modules that are supported in IxChariot tests.

5 Android

The following topics explain the installation, configuration, and operation of the Android performance endpoint. Android, developed by the Open Handset Alliance, is a software platform for mobile devices, such as smartphones and netbooks. The Android platform encompasses an operating system, middleware, an SDK, and mobile applications.

This chapter includes the following topics:

About the Android Performance Endpoint

This section identifies the requirements for installing and running the Android performance endpoint.

Requirements

Here is what you need to run the Android performance endpoint program:

- A device that supports the Android operating system, release 1.5. (which is based on Linux kernel version 2.6.27).
- Two MBytes of free RAM.
- The Android SDK installed on a Windows or Linux machine (to copy the performance endpoint files to the Android device).

Ixia has tested the performance endpoint using the Android Dev Phone 1, model G1, manufactured by HTC.

Distribution

The Android performance endpoint is provided as a single tar file. You can install it from the IxChariot CD-ROM or download it from the Ixia web site.

The performance endpoint file is named `pe_android_Mm.tar,` where "M" represents the major version and "m" represents the minor version. For example, the performance endpoint for Release 7.0 is named `pe_android_70.tar.`

Contents of the tar File

The performance endpoint tar file contains the following files:

- the performance endpoint executable.
- the `endpoint.ini` file. See Chapter 3, *Endpoint Initialization File* for information about tailoring this file for individual endpoints.
- the `echr.msg` file, which contains the text messages that will be displayed by the performance endpoint and by the IxChariot Console.
- the README file.
- the Ixia EULA (End-User License Agreement).
- the `cmpfiles` directory, containing files with the `.cmp` file extension. These files contain data of different types, such as typical text or binary data. They 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.

Little Endian

This performance endpoint is designed for Android devices with ARM little endian CPUs. (Little endian architectures store the most significant byte in a memory location with the highest address.)

Installing the Android Performance Endpoint

Installing the Android performance endpoint entails the following activities: extracting the contents of the tar file onto a Windows or Linux machine; then installing the extracted files onto the Android machine.

Extracting the Archive Contents

You use a Windows or Linux machine to extract the archive contents and copy the files to the Android machine. The following instructions assume that you obtained the performance endpoint tar file from the Ixia web site.

To decompress the performance endpoint files:

1. Log on to the Windows or Linux machine, using an account with administrator privileges. On a Linux machine, you will log in as *root*.
2. `cd` to the directory where you will extract the archive contents.
3. Extract the archive contents:

   ```
   tar -xvf pe_android_Mm.tar
   ```

   The files are extracted to a `/temp` directory. See *Contents of the tar File* on page 5-2 for a description of the files.

Installing the Performance Endpoint

Once you have extracted the contents of the tar file onto a Windows or Linux machine, you are ready to install those files onto the Android device.

To install and start the performance endpoint:

1. Install the Android SDK on a Windows or Linux machine.

   You obtain the SKD from this location: http://developer.android.com/sdk/.

2. Use a USB cable to connect the your Android device to the Windows or Linux machine.

3. Navigate to the following directory in a command line window:

   android-sdk-windows-1.5_r2\tools\

4. Enter the following commands:

   a: `adb push <path-to-the-temp-folder>\endpoint /data/local/temp/endpoint`

   b: `adb push <path-to-the-temp-folder>\endpoint.ini /data/local/temp/endpoint.ini`

   c: `adb push <path-to-the-temp-folder>\echr.msg /data/local/temp/echr.msg`

   d: `adb push <path-to-the-temp-folder>\cmpfiles\*.* /data/local/temp/cmpfiles`

   e: `adb shell`

   f: `cd /data/local/temp/`

   g: `su`

   h: `chmod 750 endpoint`

   i: `./endpoint &`

   For example:

```
adb push
D:\Ixia\IxChariot\7.0\7.0.28.63\Disk1\endpoint\android\
temp\endpoint  /data/local/7.0.28.63/endpoint

adb push
D:\Ixia\IxChariot\7.0\7.0.28.63\Disk1\endpoint\android\
temp\endpoint.ini  /data/local/7.0.28.63/endpoint.ini

adb push
D:\Ixia\IxChariot\7.0\7.0.28.63\Disk1\endpoint\android\
temp\echr.msg /data/local/7.0.28.63/echr.msg

adb push
D:\Ixia\IxChariot\7.0\7.0.28.63\Disk1\endpoint\android\
temp\cmpfiles\*.* /data/local/7.0.28.63/cmpfiles

adb shell

cd /data/local/7.0.28.63/

su

chmod 750 endpoint

./endpoint &
```

Uninstalling the Android Performance Endpoint

Follow these steps to uninstall the Android performance endpoint:

1. Log on to the Windows or Linux machine, using an account with administrator privileges. On a Linux machine, you will log in as *root*.
2. Navigate to the following directory in a command line window:

   android-sdk-windows-1.5_r2\tools\
3. Enter the following commands:

   a: `adb shell`

   b: `cd /data/local/temp/`

   c: `./endpoint -k`

   d: `rm endpoint`

   e: `rm endpoint.ini`

   f: `rm echr.msg`

   g: `rm cmpfiles/*.*`

Configuring the Android Performance Endpoint

The performance 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.

Supported Protocols

The performance endpoint for Android 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.

Configuring endpoint.ini

To modify the performance endpoint configuration file (endpoint.ini):

1. Manually modify the endpoint.ini file on Windows or Linux machine.
2. Then upload the file using the Android Debug Bridge (adb) shell on the Android device.

Refer to Chapter 3, *Endpoint Initialization File* for detailed information about the endpoint.ini options.

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.

Log Files

The default log directory is named /var/log/. If this directory is not available, the performance endpoint will use /data/local/ as an alternate log directory.

Starting and Stopping the Performance Endpoint

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

Starting the Android Performance Endpoint

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

1. Log on to the Windows or Linux machine, using an account with administrator privileges. On a Linux machine, you will log in as *root*.
2. Navigate to the following directory in a command line window:

   android-sdk-windows-1.5_r2\tools\

3. To start the endpoint, enter the following commands:

   a: `adb shell`

   b: `./endpoint &`

The optional "&" parameter indicates to Linux that the 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."

Stopping the Android Performance Endpoint

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

1. Log on to the Windows or Linux machine, using an account with administrator privileges. On a Linux machine, you will log in as *root*.
2. Navigate to the following directory in a command line window:

   android-sdk-windows-1.5_r2\tools\
3. To start the endpoint, enter the following command:

   a: `adb shell`

   b: `./endpoint -k`

The `-k` command-line option kills 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 the request to stop is not handled correctly by the running endpoint program, you may need to use the Linux "`kill -TERM`" command.

How to Tell if the Performance Endpoint is Active

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

```
ps axf | grep endpoint
```

If the performance 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
```

6

iPhone 3G/iPod Touch

The following topics explain the installation, configuration, and operation of the iPhone 3G/iPod Touch performance endpoint.

This chapter includes the following topics:

About the iPhone 3G/iPod Touch Performance Endpoint

This section identifies the requirements for installing and running the iPhone 3G/iPod Touch performance endpoint.

Requirements

Here is what you need to install and run the iPhone 3G/iPod Touch performance endpoint program:

- An iPhone 3G or an iPod Touch device, firmware version 2.2.1 or 3.0.
- Two MBytes of free RAM on the device.
- An iTunes Store account.
- iTunes running on a computer.

Ixia has tested the performance endpoint using an iPhone 3G and an iPod Touch, each with 128 MBytes of installed RAM.

Distribution

The iPhone 3G/iPod Touch performance endpoint is distributed through the Apple App Store, the same as any other iPhone application.

Installing the Performance Endpoint

This section provides instructions for downloading and installing the iPhone 3G/iPod Touch performance endpoint.

Prerequisites

To download and manage applications from the Apple App Store, you must have an iTunes Store account and you must have iTunes running on a computer.

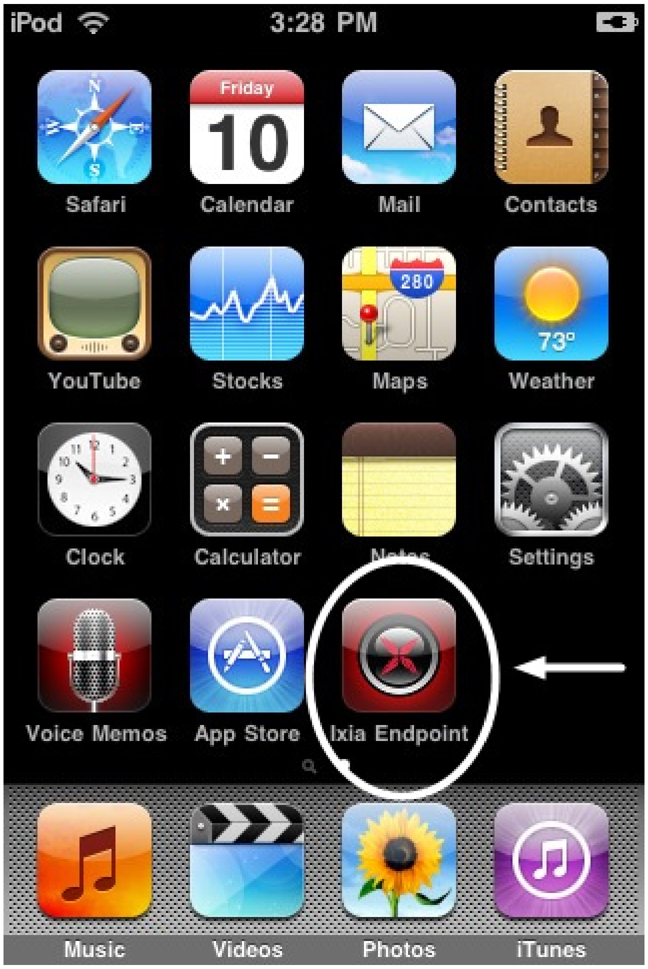
Installing Directly from the iPhone

Follow these steps to download the iPhone 3G/iPod Touch performance endpoint directly from an iPhone 3G device:

1. Connect to the Internet.
2. Start App Store.
3. Use “Ixia Endpoint” to search for the application.
4. Locate the application in the list, then tap **Get App**.
5. Sign in to your account, then tap **OK**.

App Store starts downloading the application to your iPhone. The performance endpoint icon appears on your Home screen and shows the status of the download and installation. Once the download is complete, the icon remains on the screen, as shown in Figure 6-1 on page 6-3.

Figure 6-1. iPod Screen Showing the *Ixia Endpoint* Application



When you connect the iPhone to your computer, iTunes automatically syncs the applications installed on iPhone to your iTunes library.

Installing from iTunes

You can also download the performance endpoint using iTunes on your computer. In this case, iTunes will automatically sync the application to your iPhone the next time that you connect the iPhone to your computer.

Uninstalling the iPhone 3G/ iPod Touch Performance Endpoint

You use the standard iPhone procedures to remove the iPhone 3G/iPod Touch performance endpoint from an iPhone 3G/iPod Touch device. To delete an App Store application:

1. Touch and hold the application icon on the Home screen until the icon starts to wiggle.

2. Tap the black dot with an “x” in the corner of the application.
3. Tap Delete
4. Press the Home button to save the changes.

You can reinstall any application and any associated data from your iTunes library if you previously backed up the application by syncing it to your computer.

Network Configuration

The performance 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.

Supported Protocols

The iPhone 3G/iPod Touch performance endpoint supports IPv4 over TCP, UDP, and RTP. It does not support IPv6, IPX, SPX, or APPC.

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.

Log Files

No log files are created for the iPhone 3G/iPod Touch performance endpoint.

Starting and Stopping the Performance Endpoint

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

Starting

To start the iPhone 3G/iPod Touch performance endpoint, tap its icon on the screen.

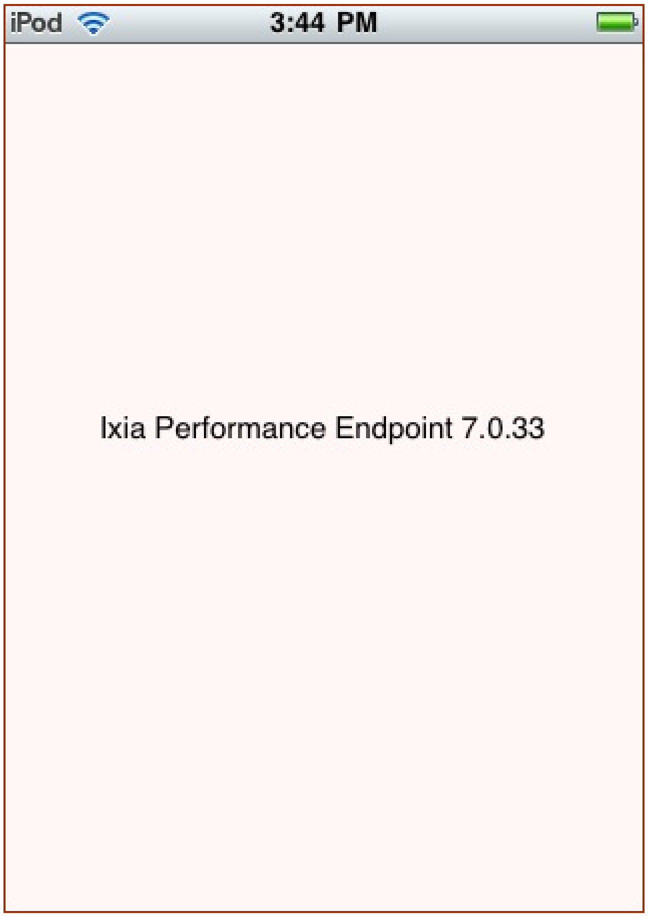
Stopping

To stop the iPhone 3G/iPod Touch performance endpoint, press the Home button. (Pressing the Home button exits from any application and displays the home screen.

How to Tell if the Performance Endpoint is Active

While the iPhone 3G/iPod Touch performance endpoint is running, it displays the performance endpoint version number, as shown in Figure 6-2.

Figure 6-2. Active iPhone 3G/iPod Touch Performance Endpoint



Ixia Performance Endpoint 7.0.33

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:

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/readstep.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

These performance endpoints are supported on the following x86-64 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 x86-64 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 *pthreads* 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 x86-64 Linux Performance Endpoints

This section provides installation instructions for both versions of the x86-64 Linux 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 uppercase 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 x86-64 Linux Endpoints

Instructions for uninstalling x86-64 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 x86-64 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 x86-64 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 an x86-64 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 an x86-64 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 an x86-64 Linux Endpoint Is Active

Use traditional Linux commands to determine if an x86-64 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 x86-64 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 64-Bit (Itanium)

The following topics explain the installation, configuration, and operation of the Performance Endpoint software for 64-bit Linux systems running on machines that use the Intel Itanium architecture (also referred to as the IA-64 architecture).

Topics in this chapter:

Supported Distributions

Ixia has tested this Performance Endpoint only on Red Hat Enterprise Linux 5.4.

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

Supported Network Protocols

The IA-64 Linux Performance Endpoint uses the Sockets interface to the TCP/IP support shipped with Linux. IPv6 is supported. IPX, SPX, or other network protocols are not supported in this version.

Installation Requirements

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

- A computer capable of running Linux Itanium edition. This includes an Itanium or Itanium 2 CPU.
- 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 8 MBytes of space available.
- 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 the IA-64 Linux Performance Endpoint

This section provides installation instructions for both versions of the IA-64 Linux Performance Endpoint (TAR-based and RPM-based):

- *TAR-Based Installation of the IA-64 Linux Endpoint* on page 9-2
- *RPM-Based Installation for the IA-64 Linux Endpoint* on page 9-4
- *What We Do During Installation* on page 9-6

TAR-Based Installation of the IA-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**.

For all the file names, *M* represents the major version and *m* represents the minor version.

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 tar command to extract the archive contents from the CD-ROM:

   ```
   cd /tmp
   rm -fr temp
   tar -zxvf ./pelinux_ia64_Mm.tar.gz
   ```

   (where *M* represents the major version and *m* represents the minor version.)

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."

6. 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 following command to extract the archive contents:

   ```
   tar -zxvf ./pelinux_ia64_Mm.tar.gz
   ```

   (where *M* represents the major version and *m* represents the minor version.)

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 IA-64 Linux Endpoint

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 uppercase 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**.

For all the file names, *M* represents the major version and *m* represents the minor version.

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 pelinux_ia64_Mm.rpm file from the CD-ROM drive to a local directory (for example, tmp).

   ```
   cp /cdrom/endpoint/linux_ia64/pelinux_ia64_Mm.rpm /tmp
   ```

   (where *M* represents the major version and *m* represents the minor version.)
4. Use the RPM command to install the endpoint:

   ```
   rpm -Uvh /tmp/pelinux_ia64_Mm.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 (this procedure uses `/tmp` as the install directory).

   ```
   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_ia64_Mm.rpm
   ```

   (where *M* represents the major version and *m* represents the minor version.)

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.

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 the IA-64 Linux Endpoint

Instructions for uninstalling the IA-64 Linux endpoint are 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 IA-64 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

It is recommended that you use Ping to verify connectivity to your endpoint computers. Before starting a test, 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 the test.

Running IA-64 Linux Performance 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 an IA-64 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 an IA-64 Linux Endpoint Is Active

Use traditional Linux commands to determine if an IA-64 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

You need to rebuild the Linux kernel to change the number of concurrent endpoint connections. Consult your IA-64 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.

10 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:

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 big 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_`*`Mn`*`.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 10-6). 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.

11 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 11-1
- *Configuring the Linux 32-bit on Lexra Endpoint* on page 11-2
- *Running the Linux 32-bit on Lexra Endpoint* on page 11-4

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.

12 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:

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 12-6). 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 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
```

13

Mac OS X

This chapter explains the installation, configuration, and operation of the Performance Endpoint software for the Mac OS X operating system.

Topics in this chapter:

Platforms Supported

The Mac OS X Performance Endpoint is a 32-bit universal binary endpoint. It runs natively, and at full performance, on both PowerPC and Intel based machines. It supports Mac OS 10.3.0 and higher.

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 10.3.0 or higher.
- 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.

Once the Performance Endpoint is installed, Mac OS will decide automatically which version of the endpoint (PowerPC or Intel) to run.

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_710.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
```

Determining CPU Type

To make it easier to determine which endpoint version is running (PowerPC or Intel), the IxChariot Endpoint Configuration dialog includes a CPU Architecture field (for the Mac OS X endpoint only).

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

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 Server 2008
- Windows Vista (32-Bit)
- Windows 7 (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 17, *Microsoft Windows CE 5.0 and Windows Mobile 5.0 and 6.0*; Chapter 18, *Microsoft Windows Embedded CE 6.0*; Chapter 15, *Microsoft Windows 64-Bit*; Chapter 20, *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 CE 4.x
- 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.ixiacom.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, Windows Server 2008, Windows Vista, or Windows 7:

- 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, Windows Server 2008, Windows Vista, or Windows 7.

  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, Windows Vista, or Windows 7 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, Windows Vista, and Windows 7.

  Quality of Service (QoS) support for TCP/IP is part of Microsoft Windows 2000, Windows XP, Windows Server 2003, Windows Vista., Windows 7, 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 14-5 or *Installing from a Downloaded Executable* on page 14-6.
- For unattended installation, refer to *Silent Mode Installation* on page 14-9.

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 the following 32-bit Windows operating systems:

  Windows NT

  Windows 2000

  Windows XP, and Windows XP Tablet PC Edition

  Windows Server 2003

- pevista32_*Mn*.exe

  Use this file for the following 32-bit Windows operating systems:

  Windows Server 2008

  Windows Vista

  Windows 7

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 7.0 is named pewindows32_70.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:

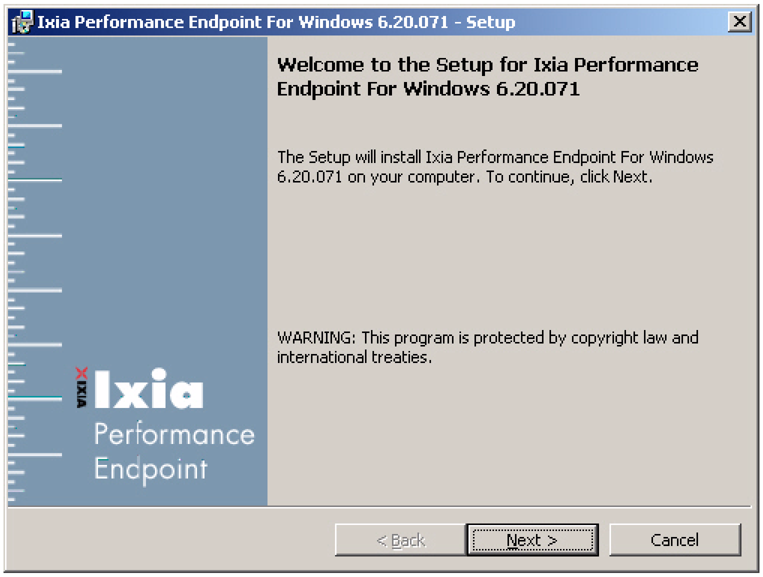
1. Stop the endpoint.
2. Delete the safestore files from the endpoint directory (or from the directory specified by the `SAFESTORE_DIRECTORY` keyword in `endpoint.ini`). Safestore files have an extension of `.q*`; you may delete them using the command delete `*.q*`.
3. Uninstall the current endpoint.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



Ixia Performance Endpoint For Windows 6.20.071 - Setup

Welcome to the Setup for Ixia Performance Endpoint For Windows 6.20.071

The Setup will install Ixia Performance Endpoint For Windows 6.20.071 on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

4. Click Next to continue.

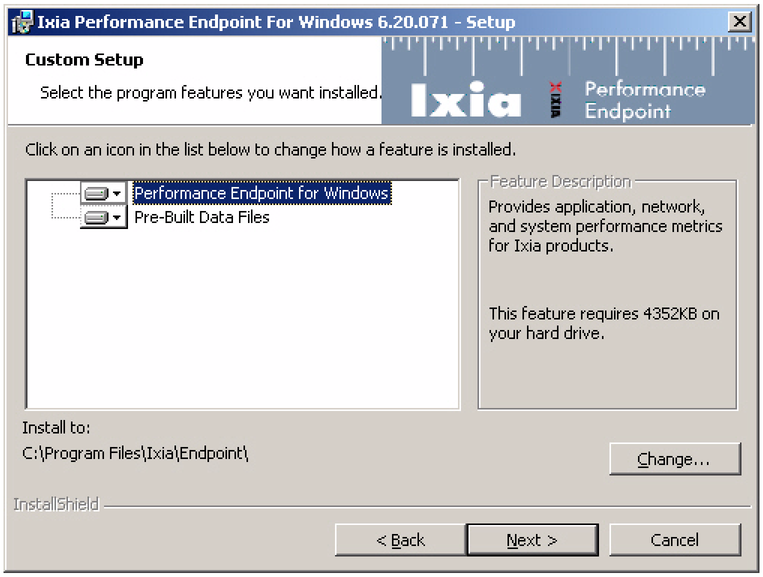
   The installer displays the Ixia Software End User License Agreement.

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.


Ixia Performance Endpoint For Windows 6.20.071 - Setup

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

- Performance Endpoint for Windows
- Pre-Built Data Files

Feature Description

Provides application, network, and system performance metrics for Ixia products.

This feature requires 4352KB on your hard drive.

Install to:
C:\Program Files\Ixia\Endpoint\

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.

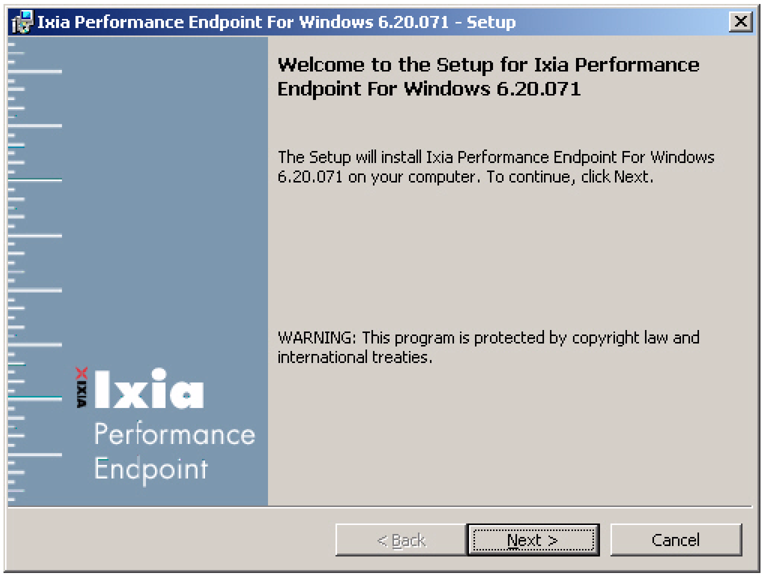
When you've completed installation, refer to *Configuring Windows Endpoints* on page 14-10 to make sure your endpoint is ready for testing and monitoring.

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:



Ixia Performance Endpoint For Windows 6.20.071 - Setup

Welcome to the Setup for Ixia Performance Endpoint For Windows 6.20.071

The Setup will install Ixia Performance Endpoint For Windows 6.20.071 on your computer. To continue, click Next.

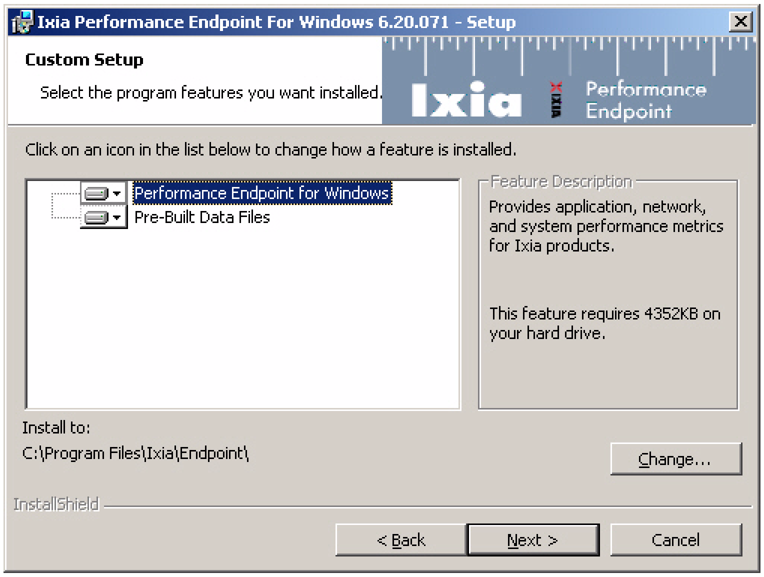
WARNING: This program is protected by copyright law and international treaties.

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.



Ixia Performance Endpoint For Windows 6.20.071 - Setup

Custom Setup

Select the program features you want installed.

Click on an icon in the list below to change how a feature is installed.

- Performance Endpoint for Windows
- Pre-Built Data Files

Feature Description

Provides application, network, and system performance metrics for Ixia products.

This feature requires 4352KB on your hard drive.

Install to:
C:\Program Files\Ixia\Endpoint\

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/qb!
```

where *filename* is the name of the specific Performance Endpoint. For example, for the release 7.0 Windows Vista Performance Endpoint, the command is:

```
C:\>pevista32_70.exe /s /v/qb!
```

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/qb!
```

where *filename* is the name of the specific Performance Endpoint. For example, for the release 7.0 Windows Vista Performance Endpoint, the command is:

```
C:\>pevista32_70.exe /s /x /v/qb!
```

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 14-11
- *Windows Configuration for TCP/IP* on page 14-12

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 mapping 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 loopback (that is, both endpoints have the same address). If the endpoint is on a Receive call, the protocol stack can pause for almost a minute before returning.

Windows 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, Windows Vista, or Windows 7 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. To keep your 32-bit Windows operating system up-to-date, you should use the Windows Update function available from your Start Menu.

15 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:

Separate Performance Endpoint executables are provided for 32-bit Windows systems, the Microsoft Windows Server 2008 R2 operating system running on Itanium-based systems, Windows CE, and the Web-based endpoint. For information about these endpoints, refer to the appropriate chapters in this manual.

Operating Systems and Processors Supported

This section identifies the operating systems and processors that support the Performance Endpoint software for 64-bit Microsoft Windows.

Supported Operating Systems

The Performance Endpoint software for 64-bit Microsoft Windows runs on the following operating systems:

- Windows 2000 64-bit
- Windows Server 2003 64-bit Edition
- Windows Server 2008 64-bit Edition
- Windows XP x64 Edition (Workstation and Server versions)
- Windows Vista (64-bit)
- Windows 7 (64-bit)

Supported Processors

These Performance Endpoints 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 these Performance Endpoints do not support the Intel Itanium processors. A separate Performance Endpoint is provided for the Microsoft Windows Server 2008 R2 operating system running on Itanium-based systems. Refer to Chapter 16, *Microsoft Windows Server 2008 R2 Itanium-Based System*, for detailed information.

Installation Requirements for the Windows 64-Bit Endpoint

The installation requirements for Microsoft Windows 64-Bit Performance Endpoint are:

- A computer equipped with one of the processor types identified in *Supported Processors* on page 15-2.
- One of the Microsoft Windows 64-bit operating systems identified in *Supported Operating Systems* on page 15-2.
- 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.
- Microsoft Windows qWAVE (on Windows Vista and Windows Server 2008).

  The qWAVE (Quality Windows Audio Video Experience) component is not installed by default in Windows Server 2008. Therefore, you need to install it prior to running the Performance Endpoint. Check your Windows Server 2008 documentation for installation instructions, or refer to *Installing Microsoft Windows qWAVE* on page 15-12 for a summary of the installation procedure.

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 15-4.
- For unattended installation, refer to *Silent Mode Installation* on page 15-5.

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 15-7 for more information.

Performance Endpoint Filenames

There are two 64-bit Windows Performance Endpoint files:

- pewindow64_*Mn*.exe

  Use this file for the following 64-bit Windows operating systems:

  Windows 2000

  Windows Server 2003 64-bit Edition

  Windows XP x64 Edition (Workstation and Server versions)

- pevista64_*Mn*.exe

  Use this file for the following 64-bit Windows operating systems:

Windows Vista

Windows 7

Windows Server 2008 64-bit Edition

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 7.0 is named pevista64_70.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 15-9 for instructions.

To prevent the endpoint from running automatically on startup, see *Disabling Automatic Startup* on page 15-10.

When you've completed installation, refer to *Configuring Windows 64-bit Performance Endpoints* on page 15-7 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/qb!
```

where *filename* is the name of the specific Performance Endpoint. For example, for the release 7.0 Windows Vista Performance Endpoint, the command is:

```
C:\>pevista64_70.exe /s /v/qb!
```

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 15-9.

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/qb!
```

where *filename* is the name of the specific Performance Endpoint. For example, for the release 7.0 Windows Vista Performance Endpoint, the command is:

```
C:\>pevista64_70.exe /s /x /v/qb!
```

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.

If You Receive an Error 1920 Message

If qWAVE is not installed on your Windows Server 2008 system, the Performance Endpoint will fail to start and will display an *Error 1920* message.

To resolve the problem, install qWAVE. Refer to *Installing Microsoft Windows qWAVE* on page 15-12 for more information.

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 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.

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

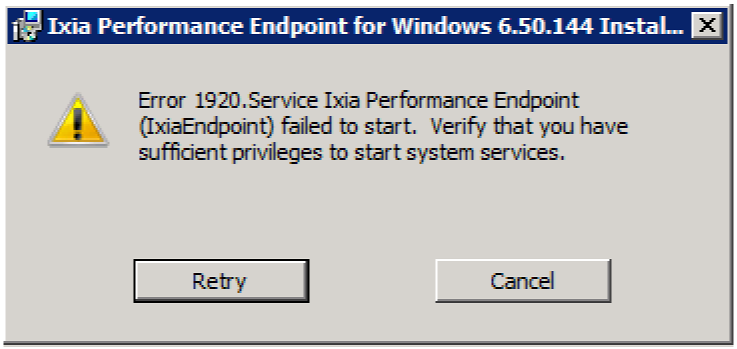
We've found that communications software is often fragile. Its developers are constantly working to make it more robust, as the software gets used in an ever-wider set of situations.

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.

Installing Microsoft Windows qWAVE

The qWAVE (Quality Windows Audio Video Experience) component is not installed by default in Windows Server 2008. Therefore, you need to install it prior to running the Performance Endpoint.

If you try to run the Performance Endpoint on a Windows Server 2008 system that does not have qWAVE installed, you will encounter Error 1920:

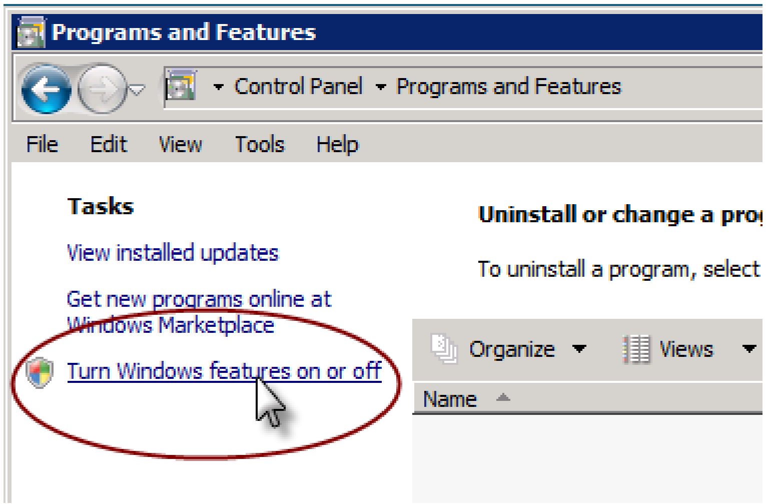


Ixia Performance Endpoint for Windows 6.50.144 Instal...

Error 1920.Service Ixia Performance Endpoint (IxiaEndpoint) failed to start. Verify that you have sufficient privileges to start system services.

The resolution of the error is to install qWAVE on the system. Check your Windows Server 2008 documentation for installation instructions, or use the following summary installation procedure:

1. Open the Control Panel and select Programs and Features.
2. Select the **Turn Windows features on or off** task on the left side of the window. For example:



Programs and Features
Control Panel ▸ Programs and Features
File Edit View Tools Help
Tasks
View installed updates
Get new programs online at Windows Marketplace
Turn Windows features on or off
Uninstall or change a pro
To uninstall a program, select
Organize
Views
Name

3. Click **Add Features**.
4. Find *Quality Windows Audio Video Experience* in the feature list, and select it using the checkbox.
5. Click **Install**.

   Windows Server 2008 installs the feature.
6. Select **Close** and then **Yes** to restart the server and complete the installation.

After the machine restarts, Windows Server 2008 displays the Installation Results window, verifying a successful installation of the feature.

16 Microsoft Windows Server 2008 R2 Itanium-Based System

This chapter explains the installation, configuration, and operation of the Performance Endpoint software for the 64-bit Microsoft Windows Server 2008 R2 operating system running on Itanium-based systems.

Topics in this chapter:

Separate Performance Endpoint executables are provided for 32-bit Windows systems, Windows CE, the Web-based endpoint, and the non-Itanium 64-bit Microsoft Windows operating systems. For the latter, refer to Chapter 15, *Microsoft Windows 64-Bit*.

Operating Systems and Processors Supported

This Performance Endpoint requires the Microsoft Windows Server 2008 R2 Itanium-Based System operating system running on a machine that uses an Intel Itanium 2 processor.

Protocols Supported

The Performance Endpoint for Microsoft Windows Server 2008 R2 Itanium-Based Systems supports the following protocols:

- IPv4 and IPv6
- TCP, UDP, and RTP

The IPX and SPX protocols are not supported.

To deactivate support for protocols you are not using, edit the ENABLE_PROTOCOL keyword in the ENDPOINT.INI file at each endpoint.

Hardware and Software Requirements

Following is a list of the installation requirements for the Microsoft Windows Server 2008 R2 Itanium-Based Systems Performance Endpoint:

- A minimum of 32 MBytes of random access memory (RAM).

  Total RAM requirement depends on RAM usage of protocol stack and number of concurrent connections or pairs. For very large tests, additional memory may be required.
- At least 8 MBytes of disk space.
- Microsoft Windows qWAVE (Quality Windows Audio Video Experience).

  The qWAVE component is not installed by default in Windows Server 2008 R2. Therefore, you need to install it prior to installing the Performance Endpoint. Check your Windows Server 2008 documentation for installation instructions, or refer to *Installing Microsoft Windows qWAVE* on page 16-7 for a summary of the installation procedure.

Installation Instructions

This section describes the requirements and procedures for installing the Microsoft Windows Server 2008 R2 Itanium-Based Systems Performance Endpoint.

Performance Endpoint Filename

The Microsoft Windows Server 2008 R2 Itanium-Based Systems Performance Endpoint file is named pewinia64_*Mm*.exe, where *M* represents the major release number and *m* represents the minor release number. For example, pewinia64_710.exe is the Performance Endpoint executable for IxChariot release 7.10.

User and System Permission Requirements

This Performance Endpoint is installed as and runs as a service. Therefore, 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 have installed the endpoint, plus any relevant subdirectories, if any.

Interactive Installation

To install the Microsoft Windows Server 2008 R2 Itanium-Based Systems Performance Endpoint interactively:

1. Log onto the target machine with a user ID that has Administrative privileges.
2. Either download the endpoint executable (pewinia64_*Mm*.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. Follow the instructions presented to complete the installation.

   Note: In the Custom Setup dialog, 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.

Silent Mode Installation

To install the Performance Endpoint using silent mode, enter the following command from the command line:

```
C:\>pewinia64_Mm.exe /s /v/qb!
```

where pewinia64_*Mm*.exe is the name of the specific Performance Endpoint. For example, for the release 7.10 Performance Endpoint, the command is:

```
C:\>pewinia64_710.exe /s /v/qb!
```

What We Do During Installation

The Performance Endpoint installer creates a directory (the default is `\Program Files\Ixia\Endpoint`) 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 `endpoint.ini` file for individual endpoints.

When you’ve completed installation, refer to *After Installation Is Complete* on page 16-4 to make sure your endpoint is ready for testing and monitoring.

The Performance Endpoint is installed as a service, and is configured to automatically start when the system reboots. Controlling the Performance Endpoint from the Services dialog box is described in *Running the Performance Endpoint* on page 16-4.

After Installation Is Complete

The Performance Endpoint program should be running at the completion of the installation. It runs in the background as a service, waiting for incoming test requests.

Testing the TCP Connection

Once the installation is complete, we recommend that you use Ping to verify connectivity to your endpoint computers. Before starting a test, 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 the test.

TCP Port Numbers

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:

- Either TCP port 10115 (the default),
- or a user-selected TCP 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. For more information on this, consult the *IxChariot User Guide*.

When runnings tests, ensure that the UDP and TCP ports used during the test are not blocked.

Running the Performance Endpoint

The following topics describe starting and stopping the Performance Endpoint for Microsoft Windows Server 2008 R2 Itanium-Based Systems, and also describe some of the messages and information that become available during testing. The Performance Endpoint for Microsoft Windows Server 2008 R2 Itanium-Based Systems 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 the Performance Endpoint

By default, the endpoint program is configured to start automatically. 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:

- At a command prompt, enter:

  ```
  net start IxiaEndpoint
  ```

- Or, 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 the 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.

If You Receive an Error 1920 Message

If qWAVE is not installed on your Windows Server 2008 system, the Performance Endpoint will fail to start and will display an *Error 1920* message.

To resolve the problem, install qWAVE. Refer to *Installing Microsoft Windows qWAVE* on page 16-7 for more information.

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 opens.
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.

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.

Uninstalling the Performance Endpoint

You can uninstall the Performance Endpoint using the Windows Control Panel, or using unattended (silent) mode.

Control Panel Uninstall

To use Control Panel to remove the endpoint package from your system, 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:\>pewinia64_Mm.exe /s /x /v/qb!
```

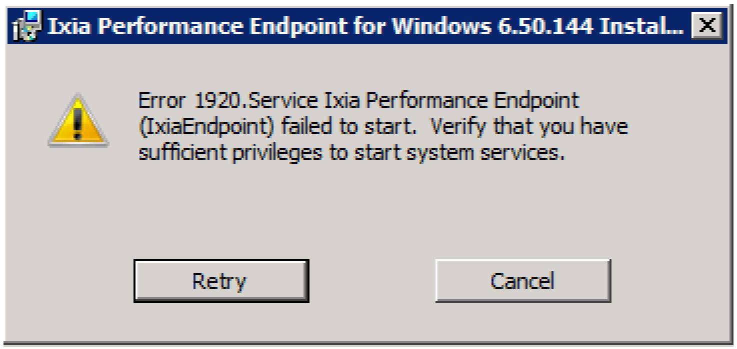
where pewinia64_*Mm*.exe is the name of the specific Performance Endpoint. For example, for the release 7.10 Performance Endpoint, the command is:

```
C:\>pewinia64_710.exe /s /x /v/qb!
```

Installing Microsoft Windows qWAVE

The qWAVE (Quality Windows Audio Video Experience) component is not installed by default in Windows Server 2008. Therefore, you need to install it prior to running the Performance Endpoint.

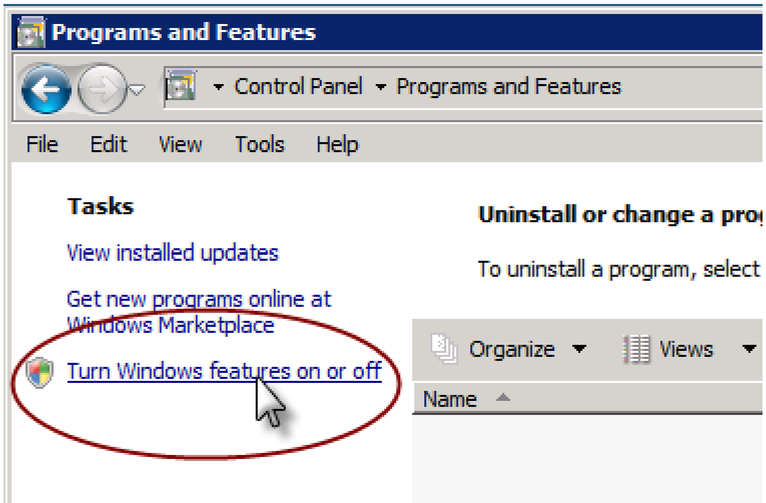
If you try to run the Performance Endpoint on a Windows Server 2008 system that does not have qWAVE installed, you will encounter Error 1920:



Error 1920.Service Ixia Performance Endpoint (IxiaEndpoint) failed to start. Verify that you have sufficient privileges to start system services.

The resolution of the error is to install qWAVE on the system. Check your Windows Server 2008 documentation for installation instructions, or use the following summary installation procedure:

1. Open the Control Panel and select Programs and Features.
2. Select the **Turn Windows features on or off** task on the left side of the window. For example:


Programs and Features
Control Panel ▾ Programs and Features
File Edit View Tools Help
Tasks
View installed updates
Get new programs online at Windows Marketplace
Turn Windows features on or off
Uninstall or change a pro
To uninstall a program, select
Organize ▾ Views ▾
Name

3. Click **Add Features**.

4. Find *Quality Windows Audio Video Experience* in the feature list, and select it using the checkbox.
5. Click **Install**.

   Windows Server 2008 installs the feature.
6. Select **Close** and then **Yes** to restart the server and complete the installation.

After the machine restarts, Windows Server 2008 displays the Installation Results window, verifying a successful installation of the feature.

17 Microsoft Windows CE 5.0 and Windows Mobile 5.0 and 6.0

This chapter describes the installation, configuration, and operation of the Performance Endpoint software for Microsoft Windows CE 5.0 and Windows Mobile 5.0 and 6.0.

Topics in this chapter:

You can run both streaming and non-streaming tests using the Performance Endpoint software for Microsoft Windows CE 5.0 and Windows Mobile 5.0 and 6.0. 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 the exceptions in *Limitations of the Windows CE Endpoint* on page 17-8.

Available Performance Endpoints

Table 17-1 on page 17-2 list the Performance Endpoints that Ixia provides for Windows CE 5.0 and Windows Mobile 5.0 and 6.0. The table shows the filename for each of the available Performance Endpoints.

Table 17-1. Performance Endpoints for Windows CE 5.0 and Windows Mobile 5.0 and 6.0

	Processor:
	ARM
Graphical User Interface	arm-ARMV4I-gui/pewce.exe
Command Line Interface	arm-ARMV4I-nogui/pewce_cl.exe
GUI with file storage	arm-ARMV4I-disk/pewce_disk.exe

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.

Installation Process

Installing a Performance Endpoint for a Windows CE 5.0, Windows Mobile 5.0, or Windows Mobile 6.0 device is a two-step process:

1. First, install the Performance Endpoint package to a Windows PC, as described in *Installing the Package* on page 17-3.

   The package includes the Performance Endpoints described in *Available Performance Endpoints* on page 17-2.

2. Then, install one of the Performance Endpoints to your target device, as described in *Installing a Performance Endpoint on a Device* on page 17-4.

Installing the Package

The Performance Endpoint package contains all of the Performance Endpoint files needed for IxChariot testing with Windows CE 5.0, Windows Mobile 5.0, or Windows Mobile 6.0.

Installation Requirements

Installation of the Performance Endpoint package requires a Windows PC with:

- 1,956 KB available disk space for the ARM Performance Endpoints.

The PC can be running Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008, Windows Vista, or Windows 7.

Administrator Privilege Requirements

You must be logged in with Administrator privileges to install the Performance Endpoint package. If you are installing the package 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.

Installation Procedure

To install the Microsoft Windows CE 5.0 / Windows Mobile 5.0 or 6.0 Performance Endpoint package to a Windows PC:

1. Log onto the PC to which you will install the package.
2. Access the package from the Ixia web site or from the product CD.

   The filename indicate the IxChariot version. For example, for pewce500armARMV4I_cl_70.exe, the *70* refers to IxChariot release 7.0.
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, select "I accept the terms of the license agreement", then click **Next**.

   The installer displays the Custom Setup dialog.
6. If you want to install the package in a folder other than the default folder (C:\Program Files\Ixia\IxChariot\wince), click **Change…**, then select the path.
7. Click **Next** to continue.

   The installer displays the Ready to Install the Program dialog.
8. Click **Install** to continue.

   The installer now copies the files and installs the Performance Endpoint package.

   Once the installation is complete, the installer displays the Setup Complete dialog.

9. Click **Finish** to complete the installation.

The installer creates three folders within the destination folder identified in step 6 above. These folders contain all the files needed for each of the Performance Endpoints described in *Available Performance Endpoints* on page 17-2.

Next Step

The next step is to install one of the Performance Endpoints on your target device, as described in *Installing a Performance Endpoint on a Device* on page 17-4.

Installing a Performance Endpoint on a Device

Once you have installed the package to a Windows PC, you can then install one of the specific Performance Endpoints to your target device.

Installation Requirements

Installation of any of Performance Endpoints requires a device that is running Windows CE 5.0, Windows Mobile 5.0, or Windows Mobile 6.0, configured with:

- 64 MB of RAM
- 530 KB available disk space

GUI Performance Endpoint Installation

To install the GUI version of the Performance Endpoint:

1. Ensure that the target Windows CE or Windows Mobile device is synched to your desktop computer (the computer on which you installed the Performance Endpoint package).
2. Navigate to the folder in which you installed the Performance Endpoint package (in *Installing the Package* on page 17-3), and within that folder, navigate to the arm_ARMV4I-gui folder.
3. Copy the Performance Endpoint executable (pewce.exe) to the Windows Clipboard using the Windows Explorer.
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 5.0 / Windows Mobile 6.0 Performance Endpoints* on page 17-6 for additional instructions.

CLI Performance Endpoint

To install the command line version of the Performance Endpoint:

1. Ensure that the target Windows CE or Windows Mobile device is synched to your desktop computer (the computer on which you installed the Performance Endpoint package).
2. Navigate to the folder in which you installed the Performance Endpoint package (in *Installing the Package* on page 17-3), and within that folder, navigate to the arm_ARMV4I-nogui folder.

3. Copy the Performance Endpoint executable (pewce_cl.exe) to your Windows CE or Windows Mobile device, using the tools available on the device.

Once you have copied the endpoint, it is ready for use. Refer to *Running Windows CE 5.0 / Windows Mobile 6.0 Performance Endpoints* on page 17-6 for additional instructions.

File-Storage Performance Endpoint

To install the file-storage version of the Performance Endpoint:

1. Ensure that the target Windows CE or Windows Mobile device is synched to your desktop computer (the computer on which you installed the Performance Endpoint package).
2. Navigate to the folder in which you installed the Performance Endpoint package (in *Installing the Package* on page 17-3), and within that folder, navigate to the arm_ARMV4I-disk folder.
3. Double-click the pewce_disk_zip.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:

   - the Performance Endpoint executable (pewce_disk.exe)
   - the End User License Agreement
   - endpoint.ini
   - echr.msg
   - the cmpfiles directory
4. Modify the endpoint.ini file, 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 5.0 / Windows Mobile 6.0 Performance Endpoints* on page 17-6 for additional instructions.

Removing the Endpoint Package (Uninstall)

The following installation instructions assume that the Windows CE or Windows Mobile device is synched to your desktop computer:

Delete the Performance Endpoint executable from the following directory on your desktop PC:

```
[Mobile Device]\My Pocket PC\Windows\Start Menu
```

Windows CE Configuration for TCP/IP

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

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 5.0 / Windows Mobile 6.0 Performance Endpoints

The following sections describe how to start, stop, and check the version of a Performance Endpoint.

Operations on GUI-Based Systems

Following are the basic operations for the Performance Endpoints running on graphical user interface (GUI)-based systems.

Starting the Performance Endpoint

On Windows CE or Windows Mobile devices that do not support a file system, tap **Start**, then tap the name of the Performance Endpoint executable.

On Windows CE or Windows Mobile devices that support a file system,, navigate to the directory where you have installed the Performance Endpoint files, then tap the name of the Performance Endpoint executable.

Stopping the Performance Endpoint (ARM Processors)

On ARM-based systems, use the following menu path on your Windows CE 5.0 or Windows Mobile 6.0 device to stop the Performance Endpoint:

1. Tap **Start** > **Settings** > **System** > **Memory** > **Running Programs**.
2. Select **Performance Endpoint** and then tap **Stop**.

Displaying the Performance Endpoint Version

The current version number is displayed on the Performance Endpoint main window.

Operations on CLI-Based Systems

Following are the basic operations for the Performance Endpoints running on command line interface (CLI)-based systems.

Starting the Performance Endpoint

Procedures for starting the command line versions 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.

Stopping the Performance Endpoint

Procedures for stopping the command line version of the Performance Endpoint depend on the tools available on the device. For example, for some devices you will use CTRL-C to stop the endpoint.

Displaying the Performance Endpoint Version

Procedures for displaying the Performance Endpoint version depend upon the tools available on the device.

Logging and Messages

All error messages encountered on a Windows CE 5.0 / Windows Mobile 6.0 Performance Endpoint are returned to the IxChariot or Qcheck Console.

For Performance Endpoints that support a file system, 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 Performance Endpoint writes a file named assert.err to the directory where you installed the endpoint.

Note that only the pewce500armARMV4I_disk_*Mm*.exe Performance Endpoint provides support for disk storage. The other Windows CE 5.0 / Windows Mobile 6.0 Performance Endpoints do not provide disk support.

Limitations of the Windows CE Endpoint

The Windows CE 5.0 / Windows Mobile 6.0 Performance Endpoints do not support the following IxChariot test facilities:

- IPTV testing.
- Disabling the UDP checksum.
- QoS templates for ToS or GQoS (only DiffServ QoS templates are supported).
- 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 5.0 and Windows Mobile 6.0 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 5.0 and Windows Mobile 6.0 will not support an IxChariot UDP 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 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.

18 Microsoft Windows Embedded CE 6.0

This chapter describes the installation, configuration, and operation of the Performance Endpoint software for Microsoft Windows Embedded CE 6.0. Windows Embedded CE 6.0 is a 32-bit real-time multitasking operating system that provides extensible CPU support.

Topics in this chapter:

You can run both streaming and non-streaming tests using the Performance Endpoint software for Microsoft Windows Embedded CE 6.0. 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 the exceptions in *Limitations of the Windows Embedded CE 6.0 Performance Endpoint* on page 18-8.

Available Performance Endpoints

Table 18-1 on page 18-2 list the Performance Endpoints that Ixia provides for Windows Embedded CE 6.0. The table shows the filename for each of the available Performance Endpoints.

Table 18-1. Performance Endpoints for Windows Windows Embedded CE 6.0

	Processor:
	ARM (Mainstone III)
Graphical User Interface	pewce600MainstoneIIIARMV4I.exe
Command Line Interface	pewce600MainstoneIIIARMV4I_cl.exe
GUI with file storage	pewce600MainstoneIIIARMV4I_disk.exe

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 Embedded CE 6.0. The Windows Embedded CE 6.0 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.

Installation Process

Installing a Performance Endpoint for a Windows Embedded CE 6.0 device is a two-step process:

1. First, install the Performance Endpoint package to a Windows PC, as described in *Installing the Package* on page 18-3.

   The package includes the Performance Endpoints described in *Available Performance Endpoints* on page 18-2.

2. Then, install one of the Performance Endpoints to your target device, as described in *Installing a Performance Endpoint on a Device* on page 18-4.

Installing the Package

The Windows Embedded CE 6.0 Performance Endpoint package contains all of the Performance Endpoint files needed for IxChariot testing with Windows Embedded CE 6.0.

Installation Requirements

Installation of the Performance Endpoint package requires a Windows PC with 1,956 KB available disk space for the ARM Performance Endpoints.

The PC can be running Windows 2000, Windows XP, Windows 2003, or Windows Vista.

Administrator Privilege Requirements

You must be logged in with Administrator privileges to install the Performance Endpoint package. If you are installing the package 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.

Installation Procedure

To install the Microsoft Windows Embedded CE 6.0 Performance Endpoint package to a Windows PC:

1. Log onto the PC to which you will install the package.
2. Access the package from the Ixia web site or from the product CD.

   The filename indicates the IxChariot version. For example, for pewce600MainstoneIIIARMV4I_cl_70.exe, the *70* refers to IxChariot release 7.0.
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, select "I accept the terms of the license agreement", then click **Next**.

   The installer displays the Custom Setup dialog.
6. If you want to install the package in a folder other than the default folder (C:\Program Files\Ixia\IxChariot\wince), click **Change…**, then select the path.
7. Click **Next** to continue.

   The installer displays the Ready to Install the Program dialog.
8. Click **Install** to continue.

   The installer copies the files and installs the Performance Endpoint package.

   Once the installation is complete, the installer displays the Setup Complete dialog.

9. Click **Finish** to complete the installation.

The installer creates three folders within the destination folder identified in step 6 above. These folders contain all the files needed for each of the Performance Endpoints described in *Available Performance Endpoints* on page 18-2.

Next Step

The next step is to install one of the Performance Endpoints on your target device, as described in *Installing a Performance Endpoint on a Device* on page 18-4.

Installing a Performance Endpoint on a Device

Once you have installed the package to a Windows PC, you can then install one of the specific Performance Endpoints to your target device.

Installation Requirements

Installation of any of Performance Endpoints requires a device that is running Windows Embedded CE 6.0, configured with:

- 64 MB of RAM
- 530 KB available disk space

GUI Performance Endpoint Installation

To install the GUI version of the Performance Endpoint:

1. Ensure that the target Windows Embedded CE 6.0 device is synched to your desktop computer (the computer on which you installed the Performance Endpoint package).
2. Navigate to the folder in which you installed the Performance Endpoint package (in *Installing the Package* on page 18-3), and within that folder, navigate to the MainstoneIII_ARMV4I-gui folder.
3. Copy the Performance Endpoint executable to the Windows Clipboard using the Windows Explorer.
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 Embedded CE 6.0 Performance Endpoints* on page 18-6 for additional instructions.

CLI Performance Endpoint

To install the command line version of the Performance Endpoint:

1. Ensure that the target Windows Embedded CE 6.0 device is synched to your desktop computer (the computer on which you installed the Performance Endpoint package).
2. Navigate to the folder in which you installed the Performance Endpoint package (in *Installing the Package* on page 18-3), and within that folder, navigate to the MainstoneIII_ARMV4I-nogui folder.

3. Copy the Performance Endpoint executable to your Windows Embedded CE 6.0 device, using the tools available on the device.

Once you have copied the endpoint, it is ready for use. Refer to *Running Windows Embedded CE 6.0 Performance Endpoints* on page 18-6 for additional instructions.

File-Storage Performance Endpoint

To install the file-storage version of the Performance Endpoint:

1. Ensure that the target Windows Embedded CE 6.0 device is synched to your desktop computer (the computer on which you installed the Performance Endpoint package).
2. Navigate to the folder in which you installed the Performance Endpoint package (in *Installing the Package* on page 18-3), and within that folder, navigate to the MainstoneIII_ARMV4I-disk or folder.
3. Double-click the pewce_disk_zip.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:

   - the Performance Endpoint executable
   - the End User License Agreement
   - endpoint.ini
   - echr.msg
   - the cmpfiles directory
4. Modify the endpoint.ini file, 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 Embedded CE 6.0 Performance Endpoints* on page 18-6 for additional instructions.

Removing the Performance Endpoint Package (Uninstall)

The following installation instructions assume that the Windows Embedded CE 6.0 device is synched to your desktop computer:

Delete the Performance Endpoint executable from the following directory on your desktop PC:

```
[Mobile Device]\My Pocket PC\Windows\Start Menu
```

Windows Embedded CE 6.0 Configuration for TCP/IP

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

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 Embedded CE 6.0 Performance Endpoints

The following sections describe how to start, stop, and check the version of a Performance Endpoint.

Operations on GUI-Based Systems

Following are the basic operations for the Performance Endpoints running on graphical user interface (GUI)-based systems.

Starting the Performance Endpoint

On Windows CE devices that do not support a file system, tap **Start**, then tap the name of the Performance Endpoint executable.

On Windows CE devices that support a file system,, navigate to the directory where you have installed the Performance Endpoint files, then tap the name of the Performance Endpoint executable.

Stopping the Performance Endpoint (ARM Processors)

On ARM-based systems, use the following menu path on your Windows Embedded CE 6.0 device to stop the Performance Endpoint:

1. Tap **Start** > **Settings** > **System** > **Memory** > **Running Programs**.
2. Select **Performance Endpoint** and then tap **Stop**.

Displaying the Performance Endpoint Version

The current version number is displayed on the Performance Endpoint main window.

Operations on CLI-Based Systems

Following are the basic operations for the Performance Endpoints running on command line interface (CLI)-based systems.

Starting the Performance Endpoint

Procedures for starting the command line versions 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.

Stopping the Performance Endpoint

Procedures for stopping the command line version of the Performance Endpoint depend on the tools available on the device. For example, for some devices you will use CTRL-C to stop the endpoint.

Displaying the Performance Endpoint Version

Procedures for displaying the Performance Endpoint version depend upon the tools available on the device.

Logging and Messages

All error messages encountered on a Windows Embedded CE 6.0 Performance Endpoint are returned to the IxChariot or Qcheck Console.

For Performance Endpoints that support a file system, 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 Performance Endpoint writes a file named `assert.err` to the directory where you installed the endpoint.

Note that only the pewce600MainstoneIIIARMV4I_disk_*Mm*.exe Performance Endpoint provides support for disk storage.

Limitations of the Windows Embedded CE 6.0 Performance Endpoint

The Windows Embedded CE 6.0 Performance Endpoints do not support the following IxChariot test facilities:

- IPTV testing.
- Disabling the UDP checksum.
- QoS templates for ToS or GQoS (only DiffServ QoS templates are supported).
- 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 Embedded CE 6.0 devices 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 Embedded CE 6.0 does not support an IxChariot UDP test with a datagram window of more than two datagrams. The test will time out with error message **CHR0216**. (This problem occurs only if you adjust

the `send_buffer_size` or Window Size parameter to include more than two UDP datagrams in a window.)

This 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.

19

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

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 names of the Solaris Performance Endpoint files are:

- pesun_*Mm*.tar.Z (Sun Solaris for SPARC)
- pes86_*Mm*.tar.Z (Sun Solaris for x86 32-Bit)

In each case, *Mm* indicates the major and minor IxChariot version number. For example. *670* for IxChariot release 6.70.

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 19-7 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 19-7 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 19-2, 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 S81endpoint 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/

20 Web-Based Performance Endpoint

This chapter describes the operation of the web-based Performance Endpoint.

Topics in this chapter:

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 25 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)

- Windows 7 (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 20-1. 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 `send_datatype` 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 **Microsoft Windows 2000/XP - web-based**.

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.

Index