Tivoli Management Framework

Firewall Security Toolbox User's Guide

Version 1.3

GC23-4826-00

Tivoli Management Framework

Firewall Security Toolbox User's Guide

Version 1.3

GC23-4826-00

Note
Before using this information and the product it supports, read the information in "Notices" on page 61.

ISO 9001 Certification

This product was developed using an ISO 9001 certified quality system.

Certification has been awarded by Bureau Veritas Quality International (BVQI)
(Certification No. BVQI - 92086 / A).

BVQI is a world leader in quality certification and is currently recognized by more than 20 accreditation bodies.

First Edition (September 2002)

This edition applies to version 1, release 3, of Tivoli Management Framework Firewall Security Toolbox (product number 5698-FRA) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Tables

Preface

Tivoli® Management Framework Firewall Security Toolbox, referred to as the Firewall Security Toolbox in the rest of this book, provides a solution for managing your Tivoli network across firewalls without compromising security. This guide explains how to install and configure this feature of Tivoli Management Framework.

Who Should Read This Guide

This guide is for administrators and system programmers who configure the firewalls in their networks.

This guide is also useful for network planners who organize the security configuration of their networks.

Readers should be familiar with the following:

- The UNIX® and Windows® operating systems
- Tivoli Management Framework

Publications

This section lists publications in the Tivoli Management Framework library and any other related documents. It also describes how to access Tivoli publications online, how to order Tivoli publications, and how to make comments on Tivoli publications.

Tivoli Management Framework Library

The following documents are available in the Tivoli Management Framework library:

- *Tivoli Management Framework: Planning for Deployment Guide*, GC32-0803

  Explains how to plan for deploying your Tivoli environment. It also describes Tivoli Management Framework and its services.

- *Tivoli Enterprise: Installation Guide*, GC32-0395

  Explains how to install and upgrade Tivoli Enterprise™ software within your Tivoli management region (Tivoli region) using the available installation mechanisms provided by Tivoli Software Installation Service and Tivoli Management Framework. Tivoli Enterprise software includes the Tivoli management region server (Tivoli server), managed nodes, gateways, endpoints, and RDBMS Interface Module (RIM) objects. This guide also provides information about troubleshooting installation problems.

- *Tivoli Management Framework: User's Guide*, GC32-0805

  Describes the concepts and procedures for using Tivoli Management Framework services. It provides instructions for performing tasks from the Tivoli desktop and from the command line.

- *Tivoli Management Framework: Maintenance and Troubleshooting Guide*, GC32-0807

  Explains how to maintain the Tivoli environment and troubleshoot problems that can arise during normal operations.

- *Tivoli Management Framework: Release Notes*, GI11-0890

  Provides late-breaking information about Tivoli Management Framework.
- *Tivoli Management Framework: Task Library Language Developers Guide*, SC32-0808

  Provides an overview of the task library language (TLL)
- *Tivoli Management Framework: Reference Manual*, SC32-0806

  Provides in-depth information about Tivoli Management Framework commands. This manual is helpful when writing scripts that are later run as Tivoli tasks. This manual also documents policy scripts provided with Tivoli Management Framework.

Related Publications

The following document also provides useful information related to Firewall Security Toolbox:

- *Tivoli Enterprise Console: Adapters Guide*, GC32-0668

  Provides detailed descriptions for the currently available Tivoli Enterprise Console® adapters.
- *IBM Tivoli Remote Control: User's Guide*, GC31-8437

  Provides information about using the Tivoli Firewall Security Toolbox with Remote Control.

The *Tivoli Glossary* includes definitions for many of the technical terms related to Tivoli software. The *Tivoli Glossary* is available, in English only, at the following Web site:

http://www.tivoli.com/support/documents/glossary/termsm03.htm

Conventions Used in This Guide

This guide uses several typeface conventions for special terms and actions. These conventions have the following meaning:

Bold	Commands, keywords, file names, authorization roles, URLs, names of windows and dialogs, other controls, or other information that you must use literally are in **bold**.
Italic	Variables and values that you must provide, new terms, and words and phrases that are emphasized are in *italics*.
`Monospace`	Code examples, output, and system messages are in a `monospace` font.

This guide uses the UNIX convention for specifying environment variables and for directory notation:

- When using the Windows command line, replace $variable with %variable% for environment variables and replace each forward slash (/) with a backslash (\) in directory paths.
- When using the bash shell on Windows, use the UNIX conventions.

Accessing Publications Online

When IBM® publishes an updated version of one or more online or hardcopy publications, they are posted to the Tivoli Information Center. You can access updated publications in the Tivoli Information Center from the following IBM Customer Support Web site for Tivoli products:

http://www.tivoli.com/support/documents/

The Tivoli Information Center contains the most recent version of the books in the product library in PDF or HTML formats, or both. Translated documents are also available for some products.

Note: If you print PDF documents on other than letter-sized paper, select the **Fit to page** check box in the Adobe Acrobat Print dialog (which is available when you click **File —> Print**) to ensure that the full dimensions of a letter-sized page are printed on the paper that you are using.

Ordering Publications

You can order many Tivoli publications online at the following Web site:

http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi

You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, for a list of telephone numbers, see the following Web site:

http://www.tivoli.com/inside/store/lit_order.html

Providing Feedback about Publications

We are very interested in hearing about your experience with Tivoli products and documentation, and we welcome your suggestions for improvements. If you have comments or suggestions about our products and documentation, contact us in one of the following ways:

- Send an e-mail to pubs@tivoli.com.
- Complete our customer feedback survey at the following Web site:
  http://www.tivoli.com/support/survey/

Accessibility

Accessibility features help users who have a physical disabilities, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

This guide provides descriptions of the graphics, which can be read by a screen reading program.

Contacting Customer Support

If you have a problem with any Tivoli product, you can contact Customer Support. See the *Customer Support Handbook* at the following Web site:

http://www.tivoli.com/support/handbook/

The handbook provides information about how to contact Customer Support, depending on the severity of your problem, and the following information:

- Registration and eligibility
- Telephone numbers and e-mail addresses, depending on the country you are in
- What information you should gather before contacting Customer Support

What This Guide Contains

This guide contains the following sections:

- Chapter 1, "Introduction" on page 1

  Provides an overview of the main concepts.
- Chapter 2, "Installing Tivoli Management Framework Firewall Security Toolbox" on page 7

  Provides instructions for installing the components.
- Chapter 3, "Configuring the Components" on page 27

  Explains how to configure the components.
- Chapter 4, "Using Firewall Security Toolbox" on page 43

  Explains how to perform various tasks, including starting and stopping the components.
- Appendix A, "Using the Command Line Interface" on page 49

  Provides usage information for the **wproxy** command.
- Appendix B, "Troubleshooting" on page 53

  Provides information to help identify and solve problems, including how to interpret the log files.

Chapter 1. Introduction

A simple Tivoli environment consists of the Tivoli management region server (Tivoli server), a gateway, and endpoints. The endpoints communicate with the Tivoli server through the gateway and the gateway communicates with the Tivoli server. See Figure 1.

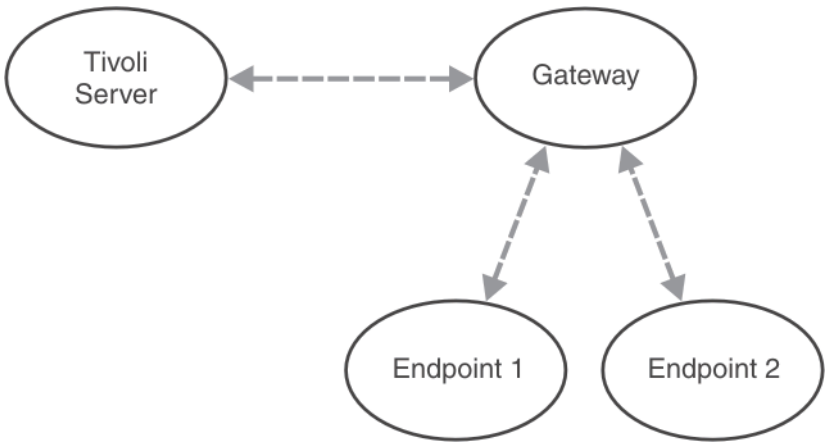


Figure 1. A simple Tivoli environment

Your Tivoli environment can be as simple or complex as your network demands. You can install multiple gateways in a Tivoli management region (Tivoli region) to manage large numbers of endpoints effectively.

When one or more firewalls exist between an endpoint and a gateway, the communication channels permitted by the firewall are limited. The Firewall Security Toolbox enables the endpoint and gateway to communicate across firewalls while respecting firewall restrictions.

Tivoli Environments with a Firewall

On the secure side of the firewall, the Firewall Security Toolbox provides an *endpoint proxy* that connects to the gateway as if it were the endpoints. On the less secure side of the firewall, the endpoints are connected to a *gateway proxy*, as if it were the gateway. The gateway proxy and endpoint proxy communicate with each other through the firewall. Figure 2 on page 2 shows a simple configuration with one gateway proxy and one endpoint proxy.

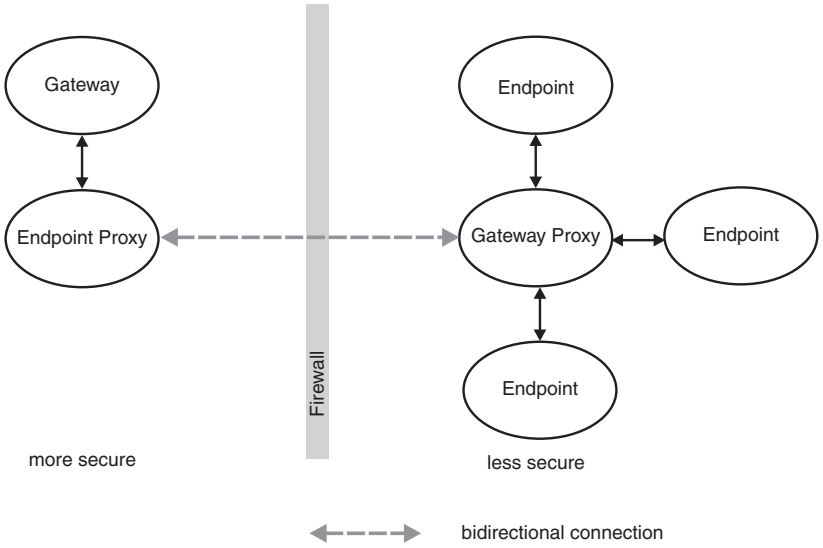


Figure 2. A Tivoli environment with an endpoint proxy and a gateway proxy connecting through a single firewall

Just as multiple endpoints can connect to a single gateway and multiple gateways to a single Tivoli server, multiple endpoints can connect to a single gateway proxy and multiple gateway proxies can connect to a single endpoint proxy. The endpoint proxy emulates all the endpoints to the gateway that manages them.

The communications between these Tivoli components is based on a Tivoli proprietary protocol over TCP/IP.

Tivoli Environments with Demilitarized Zones

When a network includes several firewalls that separate demilitarized zones (DMZs) of progressively lower security as they approach the Internet, the configuration becomes more complex. Although the gateway proxy and endpoint proxy continue to communicate with the endpoint and the gateway, respectively, they no longer communicate directly across the multiple firewalls, because this would defeat the purpose of having multiple firewalls in place.

Instead, the Firewall Security Toolbox provides *relays*, which are installed between the firewalls in DMZs. These relays pass on information to each other from one DMZ to another and, finally, to or from the endpoint proxy and gateway proxy. Figure 3 on page 3 shows an example of this configuration.

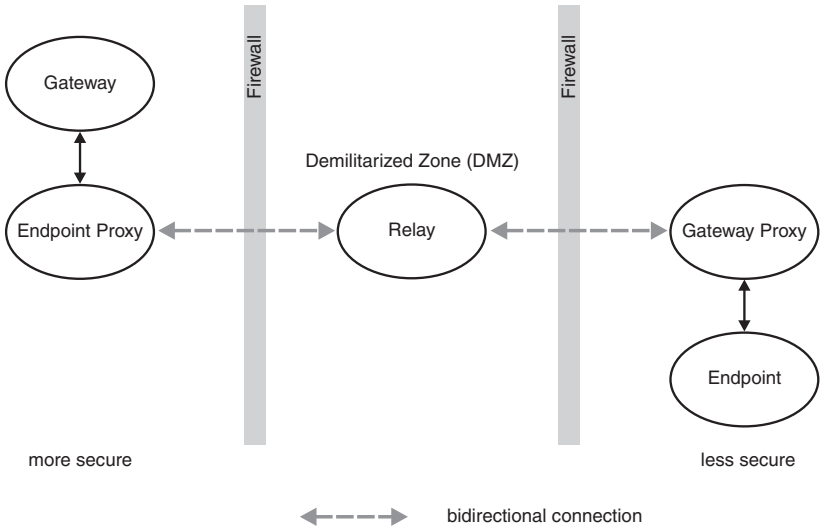


Figure 3. A Tivoli environment with the relay connecting the endpoint and gateway proxies through a DMZ

You can install a second instance of a relay on a single machine exclusively to work with the proxy components of Tivoli Remote Control Version 3.8. See the *IBM Tivoli Remote Control: User's Guide* for more details.

For all other components of the toolbox, you can install only one instance per machine.

Firewall Limitations on Connectivity

The firewall can allow connections between machines that have been initiated by either machine. These are known as *bidirectional* connections.

However, this can expose the Tivoli server to illicit connections by unauthorized machines posing as legitimate clients. To avoid such intrusions, each firewall can be configured to limit which machines can initiate a connection. This usually means that the machine on the more secure side initiates all connections with other machines on the less secure side. This machine is known as a client and becomes the *initiator*. The other machine is known as a server and become the *listener*. This type of connection is known as *unidirectional*. The Tivoli Management Framework Firewall Security Toolbox enables you to configure unidirectional connections among the endpoint proxy, gateway proxy, and relays in your Tivoli environment.

In a configuration composed of multiple firewalls and DMZs, you can set up a network that allows a mix of unidirectional and bidirectional connections. For example, a bank branch office, which operates within a secure intranet, connects to its main administrative office through a series of relays (see Figure 4 on page 4).

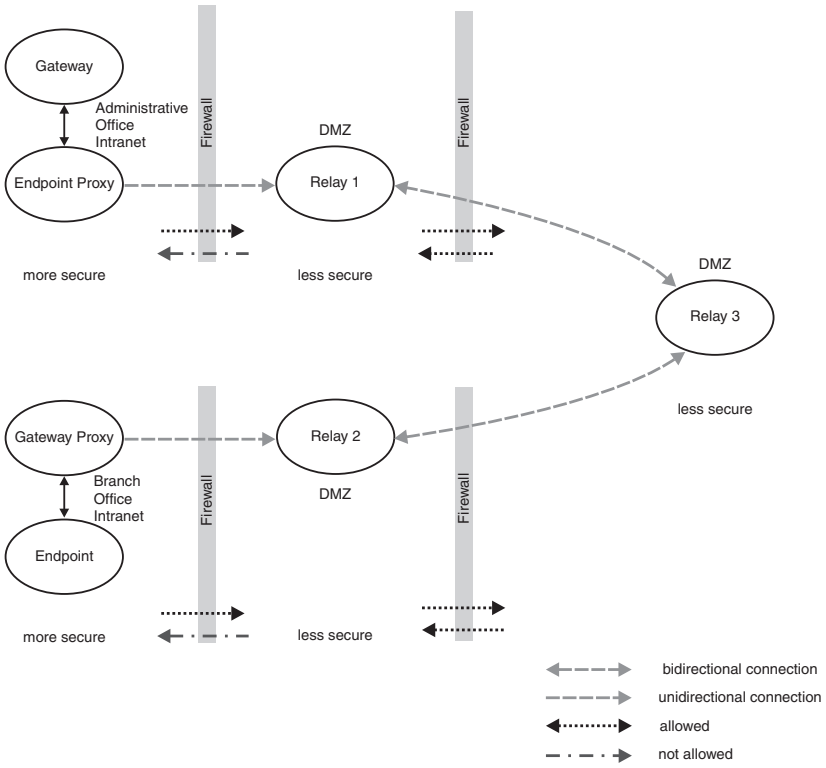


Figure 4. Example of an environment with DMZs, relays, and bidirectional and unidirectional connections

From the more secure sides, the offices use unidirectional connections to the relays 1 and 2. Relays 1 and 2 use bidirectional connections to relay 3 in the less secure areas. In the unidirectional connection between the gateway proxy and relay 2, the gateway proxy is the *initiator* and relay 2 is the *listener*.

Sending Events Across Firewalls

TME adapters use endpoints to send events to the Tivoli Enterprise Console server through Tivoli connections. When a firewall separates the endpoint from the Tivoli Enterprise Console server, the machines connect through the gateway and endpoint proxies.

Machines that are not part of the Tivoli environment use non-TME adapters to send events to the Tivoli Enterprise Console server through non-Tivoli connections. When a firewall separates the non-TME adapter machine from the gateway, the Tivoli Management Framework Firewall Security Toolbox provides the *event sink*, which sends the events to the Tivoli Enterprise Console server. The event sink, which is installed on an endpoint outside the firewall, collects events sent from non-TME adapters as if it were a Tivoli Enterprise Console server and sends them to the Tivoli Enterprise Console server as though they were TME events. The event sink can collect events from multiple non-TME adapters. See Figure 5 on page 5

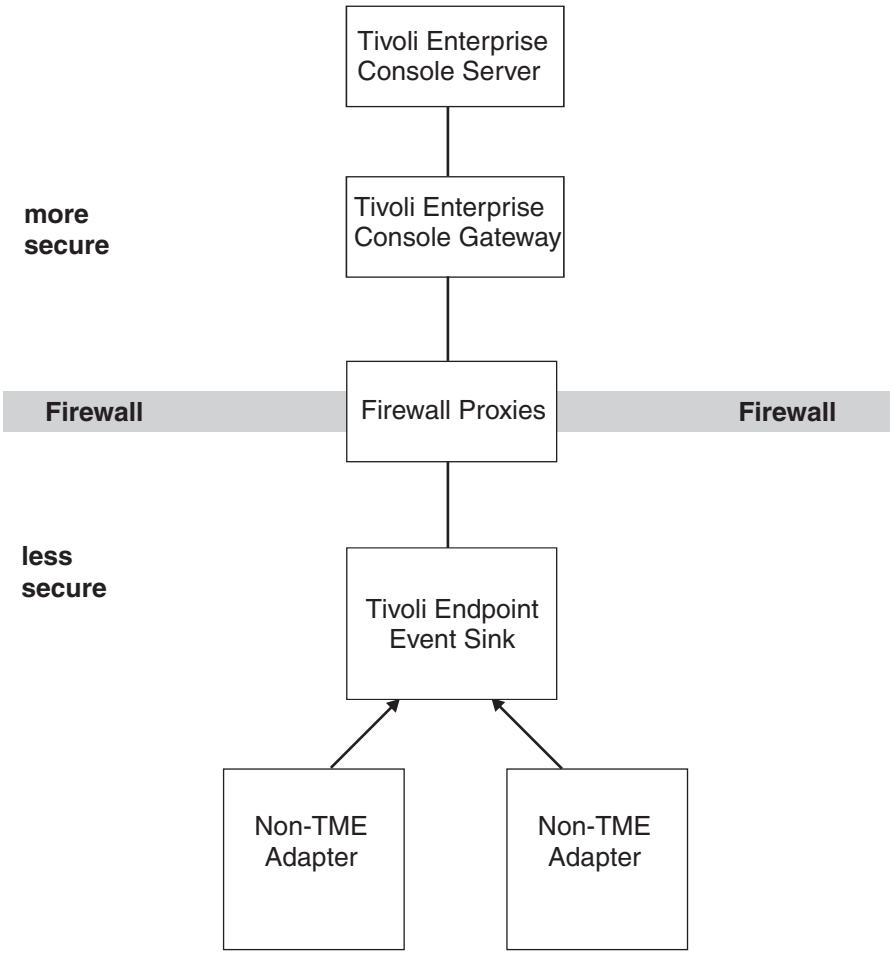


Figure 5. an event sink sends non-TME events to the Tivoli Enterprise Console server through the firewall proxies

The event sink can also collect Tivoli events that are sent from Tivoli Distributed Monitoring or the Tivoli Enterprise Console Availability Intermediate Manager Console and forward them to Tivoli Enterprise Console servers across firewalls.

In addition, you can configure the event sink to forward non-TME events to the Tivoli Enterprise Console server as if it were a Tivoli Enterprise Console adapter.

Identifying a Component as Parent or Child

The hierarchy of the components of the Tivoli Management Framework Firewall Security Toolbox is presented in terms of a *parent* and *children*. The endpoint proxy, the parent, can have one or more children, either relays or gateway proxies. It does not have a parent because it connects to the gateway.

A relay can have a parent (an endpoint proxy or another relay) and children (relays or gateway proxies). A gateway proxy has a parent but no children because it connects to endpoints.

For example, in Figure 3 on page 3, the endpoint proxy is the parent and the relay is its child. The relay is the parent of the gateway proxy and the gateway proxy is its child. The endpoint proxy and relays can have more than one child (relay or gateway proxy) but each component has only one parent. Because the gateway

proxy is at the bottom of this hierarchy, it has no children. In another example (Figure 4 on page 4), Relay 1 is the child of the endpoint proxy and the parent of Relay 3. Relay 3 is the parent of Relay 2. Relay 2 is the parent of the gateway proxy. Understanding this hierarchy is important when you install and configure the components.

Chapter 2. Installing Tivoli Management Framework Firewall Security Toolbox

This chapter explains how to install and configure the components of the Firewall Security Toolbox.

Prerequisite Software

Tivoli Management Framework Firewall Security Toolbox is a feature of the Tivoli Management Framework, Version 3.7.1 or 4.1.

The event sink must be have Tivoli Management Framework 3.7.1 or 4.1 (endpoint) installed. It must also have Tivoli Enterprise Console 3.6.2 or 3.7.

All other components of the Tivoli Management Framework Firewall Security Toolbox do *not* require any Tivoli software.

Table 1 lists the operating system software required to run the toolbox components:

Table 1. Prerequisite software

Component	Supported Operating Systems
Endpoint proxy	• Windows NT®, Version 4.0 • Windows XP Professional • Windows 2000 • AIX®, Version 4.3.3 or 5.1 • HP-UX Version 11.0 or 11i • Red Hat Linux for Intel, interpreter i386, Version 7.1 (Kernel 2.4.2-2) or 7.2 (Kernel 2.4.7-10smp) • Sun Solaris Operating System, Version 2.7 or 2.8
Gateway proxy	• Windows NT, Version 4.0 • Windows XP Professional • Windows 2000 • AIX, Version 4.3.3 or 5.1 • HP-UX Version 11.0 or 11i • Red Hat Linux for Intel, interpreter i386, Version 7.1 (Kernel 2.4.2-2) or 7.2 (Kernel 2.4.7-10smp) • Sun Solaris Operating System, Version 2.7 or 2.8
Relay	• Windows NT, Version 4.0 • Windows XP Professional • Windows 2000 • AIX, Version 4.3.3 or 5.1 • HP-UX Version 11.0 or 11i • Red Hat Linux for Intel, interpreter i386, Version 7.1 (Kernel 2.4.2-2) or 7.2 (Kernel 2.4.7-10smp) • Sun Solaris Operating System, Version 2.7 or 2.8

Table 1. Prerequisite software (continued)

Component	Supported Operating Systems
Event sink	• Windows NT, Version 4.0 • Windows XP Professional • Windows 2000 • AIX, Version 4.3.3 or 5.1 • HP-UX Version 11.0 or 11i • Red Hat Linux for Intel, interpreter i386, Version 7.1 (Kernel 2.4.2-2) or 7.2 (Kernel 2.4.7-10smp) • Sun Solaris Operating System, Version 2.7 or 2.8

Planning Where to Install the Components

You can install as many gateways and endpoint proxies as you need in the firewall region. A recommended ratio is one endpoint proxy for every 500 endpoints.

Install multiple gateway proxies in a DMZ to provide backup gateway proxies when the main gateway proxy is unavailable. See "Configuring Backup Gateway Proxies" on page 39 for configuration details.

You cannot run different components on the same machine and use them to connect to the same component on another machine. For example, you cannot have a relay and a gateway proxy on a machine and use them to connect to the same endpoint proxy. You can connect the gateway proxy to the relay on the same machine and the relay to an endpoint proxy on another machine.

Install a few endpoints first to test connectivity from the Tivoli region to an endpoint through the proxy. Enter the following command from the management region server (Tivoli server) or managed node:

```
wadminep endpoint view_config_info
```

where *endpoint* is the label of the endpoint.

If you cannot reach the endpoint, follow the instructions in "Testing Proxy Configuration" on page 53.

Because the toolbox requires a particular configuration of the Tivoli region, keep machines that are in less secure zones in separate Tivoli regions. Set up a separate Tivoli region to manage resources that are in a DMZ. To manage your endpoints as if they were in a single Tivoli region, you can interconnect the firewall region to non-firewall regions.

Getting Started

You need to do the following to get started:

- Ensure that the components of the Tivoli Management Framework Firewall Security Toolbox that will communicate with each other directly have IP visibility of each other. Depending on your configuration, these components can be the endpoint proxy and gateway proxy, a proxy and a relay, or two relays. You can use DNS if you have DNS configured. However, there is no requirement to use DNS host names. The TCP/IP address works as well. TCP/IP connectivity is required. If you use the DNS name of the machine, ensure that the DNS name of the destination machine is resolved into its IP address.

- Before installing the software, ensure that you have the following information:
  - The port number of the gateway that the endpoint proxy will use to communicate
  - The host name or IP address of all the components that you are installing
- Decide on some additional ports that the components will use to communicate with each other:
  - The endpoint proxy requires a range of ports used to emulate the endpoints logged in through the Firewall Security Toolbox.
  - Gateway proxies require one port to receive traffic from the endpoint proxy or relay and another port to listen for traffic from the endpoints.
  - Relays require ports to receive traffic from the components with which they connect, one for the parent and one for the children.

  Use the following criteria to choose the port number:
  - The port must not be used by other applications
  - The user account that you specify must have the privileges to use the port

Components on Multihomed Hosts

When a machine has more than one network interface and address, it is known as a *multihomed* host. Multihomed hosts might need to connect to one component in one subnet and another component in another subnet. For example, an endpoint proxy machine might connect to a gateway in one subnet and relays or gateway proxies in another subnet (see Figure 6).

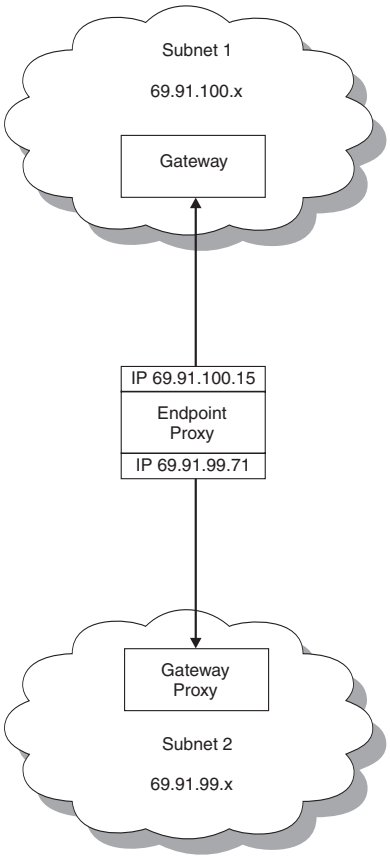


Figure 6. Example of a multihomed host

The endpoint proxy can connect to the gateway using the DNS name or IP address of one network interface, for example, 69.91.100.15, and to the relays or gateway proxies using the DNS name or IP address of another network interface, for example, 69.91.99.71. When configuring the components with multihomed hosts, you need to specify the correct DNS name or IP address. See Chapter 3, "Configuring the Components" on page 27 for more details.

Decompressing the Installation Files

To install Tivoli Management Framework Firewall Security Toolbox, decompress the 1.3-TFS-0001.tar file. Under the main Proxy directory, the file creates directories for each component and copies installation scripts to subdirectories for each platform.

Installing on UNIX Systems

The following sections describe how to install the components on UNIX systems. These operations need to be run as root user.

Installing the Endpoint Proxy on UNIX Systems

To install the endpoint proxy, follow these steps:

1. From the `EndpointProxy` directory, go to the directory for the platform on which the proxy will run.
2. Run the ./install.sh script.
3. Provide the following information:
   a. `To install the Tivoli Endpoint Proxy, you must accept the agreement written in the License file. If you accept the agreement, enter Y. [Y/N]:`

      Enter `Y` to accept the license agreement and continue the installation.

   b. `Installation directory [default=/usr/epp]:`

      Specify the directory where you want to install the endpoint proxy. If the directory does not exist, you are prompted to create it.

   c. `Run the endpoint proxy as the following user:`

      Specify the account name to use to run the proxy process. This account must exist and it is recommended that you use an unprivileged account.

   d. `Gateway address:`

      Specify the host name or IP address of the gateway with which the endpoint proxy communicates. The endpoint proxy can communicate with only one gateway in a Tivoli region.

   e. `Gateway port [default=9494]:`

      Specify the TCP/IP port number of the gateway on which it will listen for communication from the endpoint proxy as if it were the endpoint. This is normally port 9494. Do not change this value unless the gateway is known to be using a different listening port with the endpoint.

   f. `Endpoint Proxy Port:`

      Specify the port number of the endpoint proxy machine from which it listens for connections with the relay or gateway proxy.

   g. `Relay or gateway proxy hostname:`

      Specify the host name of the relay or gateway proxy with which the endpoint connects.

h. `Relay or gateway proxy port:`

   Specify the port number from which the relay or gateway proxy listens for connections from the endpoint proxy.

i. `Enter more destinations? [Y/N] [default=N]:`

   Enter `Y` to specify additional relays or gateway proxies with which the endpoint connects. You are asked to specify the host name and port number of the additional destination machines. When you are done adding destination machines and are ready to continue with installation, enter `N`.

j. `Specify how endpoint proxy connects to destinations [0=bidirectional, 1=unidirectional initiator, 2=unidirectional listener]:`

   Enter `0` to permit connections that are initiated by either machine.

   Enter `1` to permit connections initiated by the endpoint proxy only.

   Enter `2` to permit connections initiated by the destination machine only.

k. `Start the endpoint proxy? [Y/N] [default=Y]:`

   Enter `Y` to start the endpoint proxy. Otherwise, enter `N`. To start the component at another time, see "Starting and Stopping the Components" on page 43.

Installing the Gateway Proxy on UNIX Systems

To install the gateway proxy, follow these steps:

1. From the `GatewayProxy` directory, go to the directory for the platform on which the proxy will run.
2. Run the ./install.sh script.
3. Provide the following information:
   a. `To install the Tivoli Gateway Proxy, you must accept the agreement written in the License file. If you accept the agreement, enter Y. [Y/N]:`

      Enter `Y` to accept the license agreement and continue the installation.

   b. `Installation directory [default=/usr/gwp]:`

      Specify the directory where you want to install the gateway proxy. If the directory does not exist, you are prompted to create it.

   c. `Run the gateway proxy as the following user:`

      Specify the account name to use to run the proxy process. This account must exist and it is recommended that you use an unprivileged account.

   d. `Name (label) for this proxy [default=localhost]:`

      Optionally, enter a name to identify the gateway proxy.

   e. `Port to listen on for TMA traffic [default=9494]:`

      Enter the port number on the gateway proxy that represents the gateway to the endpoints. The default is 9494.

   f. `Gateway proxy port:`

      Specify the port number that the gateway proxy uses to listen for connections from the relay or endpoint proxy.

   g. `Relay or endpoint proxy hostname:`

      Specify the host name of the machine that the gateway proxy will connect up the chain toward the Tivoli gateway.

h. `Relay or endpoint proxy port:`

Enter the port number of the machine that the gateway proxy will connect up the chain toward the Tivoli gateway.

i. `Specify how gateway proxy connects to destination [0=bidirectional, 1=unidirectional initiator, 2=unidirectional listener]:`

Specify how the gateway proxy connects to the destination relay or endpoint proxy.

Enter `0` to permit connections that are initiated by either machine.

Enter `1` to permit connections initiated by the gateway proxy only.

Enter `2` to permit connections initiated by the parent machine only.

j. `Start the gateway proxy? [Y/N] [default=Y]:`

Enter `Y` to start the gateway proxy. Otherwise, enter `N`. To start the component at another time, see "Starting and Stopping the Components" on page 43.

Installing the Relay on UNIX Systems

To install the relay, follow these steps:

1. From the `Relay` directory, go to the directory for the platform on which the relay will run.
2. Run the ./install.sh script.
3. Provide the following information:
   a. `To install the Tivoli Relay, you must accept the agreement written in the License file. If you accept the agreement, enter Y. [Y/N]:`

   Enter `Y` to accept the license agreement and continue the installation.

   b. `Installation directory [default=/usr/relay-1]:`

   Specify the directory where you want to install the relay. If the directory does not exist, you are prompted to create it.

   Note: The default directory ends with a directory named for the numbered instance of the relay. You must launch all operations (start, stop, and uninstall) on an instance of the relay from the directory in which you install the instance.

   If you change the default directory, ensure that it is different from the directory where you have installed any previous instances of the relay.

   c. `Run the relay as the following user:`

   Specify a user name to use to run the relay. This account must exist and it is recommended that you use an unprivileged account.

   d. *Relay-Parent Connection Options*

   `Relay port:`

   Enter the port number for the relay to communicate with the parent machine.

   e. `Relay or endpoint proxy hostname:`

   Enter the host name for the parent relay or endpoint proxy with which the relay will communicate.

   f. `Parent Remote Port:`

   Enter the port number for the parent relay or endpoint proxy with which the relay will communicate.

g. `Specify how relay connects to parent relay or endpoint proxy [0=bidirectional, 1=unidirectional initiator, 2=unidirectional listener]:`

   Specify how the relay connects to the destination relay or gateway proxy.

   Enter `0` to permit connections that are initiated by either machine.

   Enter `1` to permit connections initiated by the relay only.

   Enter `2` to permit connections initiated by the parent machine only.

h. *Relay-Children Connection Options*

   `Relay port:`

   Enter the port number for the relay to communicate with the children machines.

i. `Relay or gateway proxy hostname:`

   Specify the host name of the child machine, relay or gateway proxy, with which the relay connects.

j. `Relay or gateway proxy port:`

   Enter the port number of the machine with which the relay connects.

k. `Enter more destinations? [Y/N] [default=N]:`

   Enter `Y` to specify additional relays or gateway proxies with which the relay connects. You are asked to specify the host name and port number of the additional destination machines. When you are done adding destination machines and are ready to continue with installation, enter `N`.

l. `Specify how the relay connects to the child destination [0=bidirectional, 1=unidirectional initiator, 2=unidirectional listener]:`

   Specify how the relay connects to the parent relay or endpoint proxy.

   Enter `0` to permit connections that are initiated by either machine.

   Enter `1` to permit connections initiated by the relay only.

   Enter `2` to permit connections initiated by the destination machine only.

m. `Start the relay? [Y/N] [default=Y]:`

   Enter `Y` to start the relay. Otherwise, enter `N`. To start the component at another time, see "Starting and Stopping the Components" on page 43.

Installing the Event Sink on UNIX Systems

To install the event sink, follow these steps:

1. From the `EventSink` directory, go to the directory for the platform on which the proxy will run.
2. Run the ./install.sh script.
3. Provide the following information:
   a. `To install the Tivoli Event Sink, you must accept the agreement written in the License file. If you accept the agreement, enter Y. [Y/N]:`

      Enter `Y` to accept the license agreement and continue the installation.

   b. `Installation directory [default=/usr/eventsink]:`

      Specify the directory where you want to install the event sink. If the directory does not exist, you are prompted to create it.

   c. `LCF_DATDIR directory:`

      Specify the LCF_DATDIR directory of the endpoint on which you are installing the event sink.

d. `Run the event sink as the following user:`

   Specify a user name to use to run the event sink. This account must exist and it is recommended that you use an unprivileged account.

e. Listening Port [default=9444]:

   Enter the port number on the endpoint where the event sink will receive events.

f. Maximum Number of Events in Package [default=50]:

   Enter the maximum number of events that the event sink will send to the Tivoli Enterprise Console server in a single package.

g. Maximum Buffer Size [default=40000]:

   Enter the maximum buffer size, in bytes, of the package that the event sink will send to the Tivoli Enterprise Console server.

h. `Start the event sink? [Y/N] [default=Y]:`

   Enter `Y` to start the event sink. Otherwise, enter `N`. To start the component at another time, see "Starting and Stopping the Components" on page 43.

Installing on Windows Systems

The Firewall Security Toolbox provides a self-extracting EXE file to install each component on Windows systems. The installation files are unpacked into a default directory, which you can change. You need to specify this directory only the first time you run this file. You can use these files for any future installations.

You can install in one of the following ways:

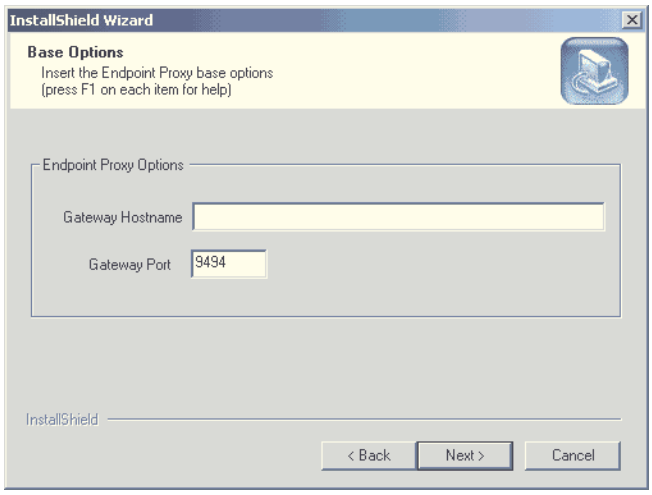
- By running the self-extracting EXE file
- By running setup.exe if you are using a disk image

Installing the Endpoint Proxy on Windows Systems

To install the endpoint proxy, do the following:

1. From the directory that contains the Tivoli Endpoint Proxy\w32-ix86\ subdirectory, double-click the Tivoli Endpoint Proxy.exe file. The Tivoli Endpoint Proxy InstallShield Wizard starts.
2. Click **Next**.
3. On the next dialog, click **Yes** to accept the license agreement.

4. On the next dialog, enter the installation directory and click **Next**. The dialog for Endpoint Proxy Options is displayed.

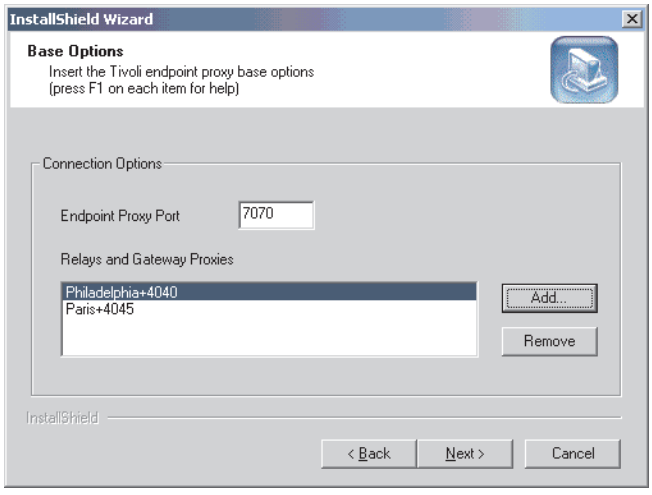


5. Complete the following fields:

   Gateway Hostname
   Enter the host name or IP address of the gateway with which the endpoint proxy communicates. The endpoint proxy can communicate with only one gateway in a Tivoli region.

   Gateway Port
   Enter the TCP/IP port number of the gateway on which it will listen for communication from the endpoint proxy as if it were the endpoint. The default is 9494 and should not be changed unless the gateway is known to be using a different listening port with the endpoint.

   Click **Next**. The dialog for Connection Options is displayed.

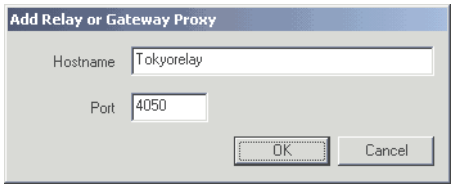


6. Complete the following fields:

   Endpoint Proxy Port
   Enter the port number of the endpoint proxy machine from which it listens for connections with the relay or gateway proxy.

   Relays and Gateway Proxies
   Lists the relays and gateway proxies with which the endpoint proxy connects.

To add a relay or gateway proxy to the list of destination machines, click **Add**. The Add Relay or Gateway Proxy dialog is displayed.



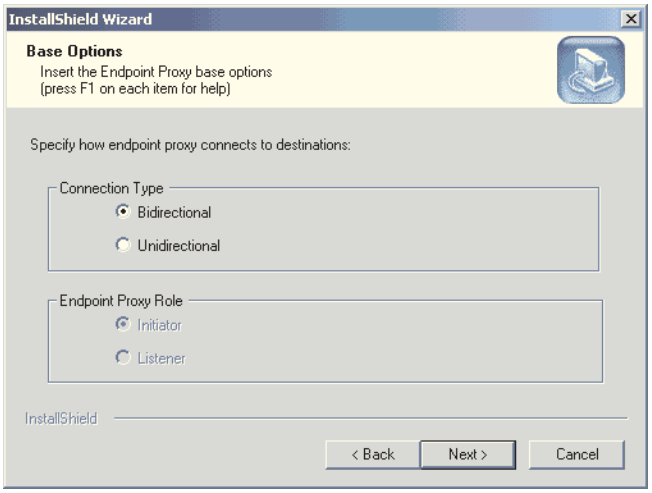
Complete the fields and click **OK**:

Hostname
Enter the host name of the relay or gateway proxy with which the endpoint connects.

Port Enter the port number from which the relay or gateway proxy listens for connections from the endpoint proxy.

To remove a machine, select it and click **Remove**.

Click **Next**. The dialog for the type of endpoint proxy connection is displayed.



7. Specify how the endpoint proxy connects to the destination relay or gateway proxy:
   - Select **Bidirectional** to permit connections that are initiated by either machine.
   - Select **Unidirectional** to permit connections initiated by only one machine.

     If you select this option, the Endpoint Proxy Role box is enabled:

     Initiator
     Select to specify that the endpoint proxy machine can start the connection with the destination machines.

     Listener
     Select to specify that the destination machines can start the connection with the endpoint proxy machine.
8. Click **Next**. The next dialog shows a summary of your input.
9. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the endpoint proxy.

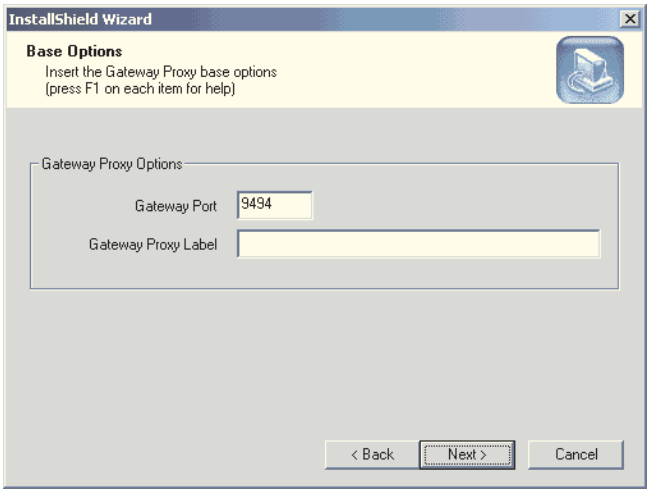
10. A message asks you whether or not you want to start the endpoint proxy. Click **Yes** to start it. Otherwise, click **No**. To start the component at another time, see "Starting and Stopping the Components" on page 43.

Installing the Gateway Proxy on Windows Systems

The gateway proxy needs to be installed on a host that is in the DMZ where the endpoints will be located.

To install the gateway proxy, do the following:

1. From the directory that contains the gateway Proxy\w32-ix86\ subdirectory, double-click the Tivoli Gateway Proxy.exe file. The Tivoli Gateway Proxy InstallShield Wizard starts. Click **Next**.
2. On the next dialog, click **Yes** if you accept the license agreement.
3. On the next dialog, enter the installation directory and click **Next**. The dialog for Gateway Proxy Options is displayed.

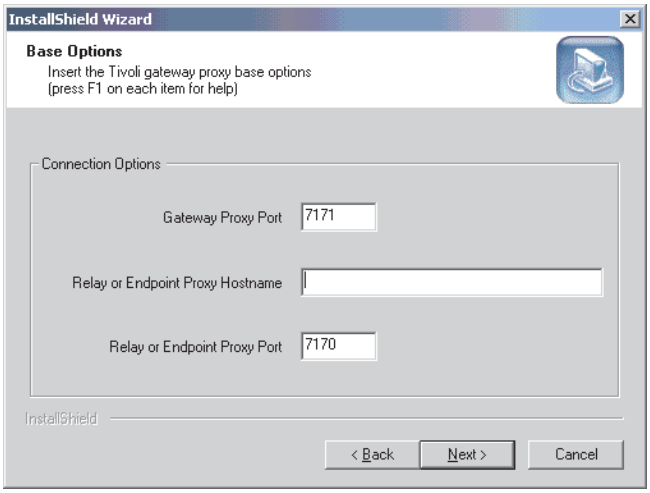


4. Complete the following fields:

Gateway Port
Enter the port number on the gateway proxy that represents the gateway to the endpoints. The default is 9494.

Gateway Proxy Label
Optionally, enter a name to identify the gateway proxy.

Click **Next**. The dialog for Gateway Proxy-Parent Connection Options is displayed.



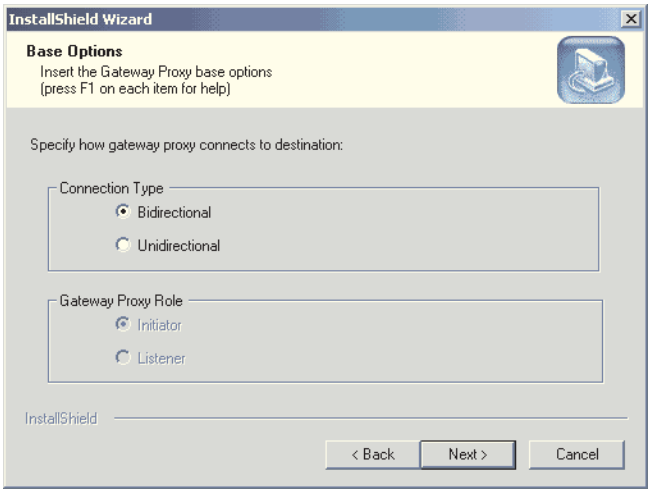
5. Complete the following fields:

   Gateway Proxy Port
   Enter the port number that the gateway proxy uses to listen for connections from the relay or endpoint proxy.

   Relay or Endpoint Proxy Hostname
   Enter the host name of the machine that the gateway proxy will connect to up the chain toward the gateway.

   Relay or Endpoint Proxy Port
   Enter the port number of the machine that the gateway proxy will connect to up the chain toward the gateway.

   Click **Next**. The dialog for the type of gateway proxy connection is displayed.



6. Specify how the gateway proxy connects to the destination relay or endpoint proxy:
   - Select **Bidirectional** to permit connections that are initiated by either machine.
   - Select **Unidirectional** to permit connections initiated by only one machine.

If you select this option, the Gateway Proxy Role box is enabled:

Initiator
Select to specify that the gateway proxy machine can start the connection with the parent endpoint proxy or relay machine.

Listener
Select to specify that the parent machine can start the connection with the gateway proxy machine.

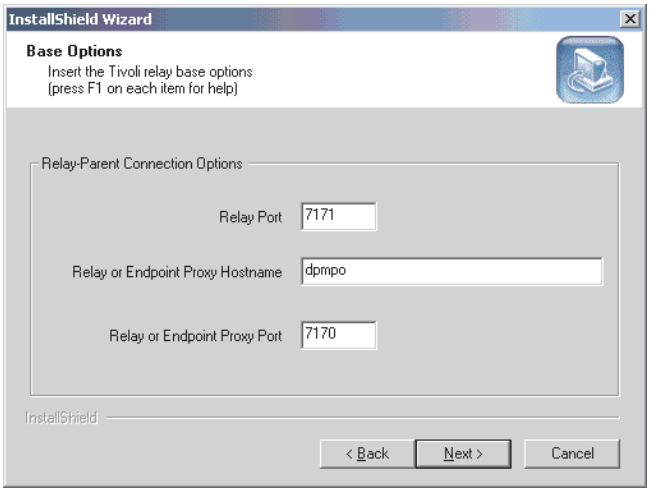
Click **Next**. The next dialog shows a summary of your input.

7. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the gateway proxy.
8. A message asks you whether or not you want to start the gateway proxy. Click **Yes** to start it. Otherwise, click **No**. To start the component at another time, see "Starting and Stopping the Components" on page 43.

Installing the Relay on Windows Systems

You can install multiple instances of a relay on a single machine. To install the first relay, do the following:

1. From the directory that contains the Tivoli Relay installation images, double-click the setup.exe file. The Tivoli Relay InstallShield Wizard starts. Click **Next**.
2. On the next dialog, click **Yes** if you accept the license agreement.
3. On the next dialog, enter the installation directory and click **Next**. The dialog for Relay-Parent Connection Options is displayed.



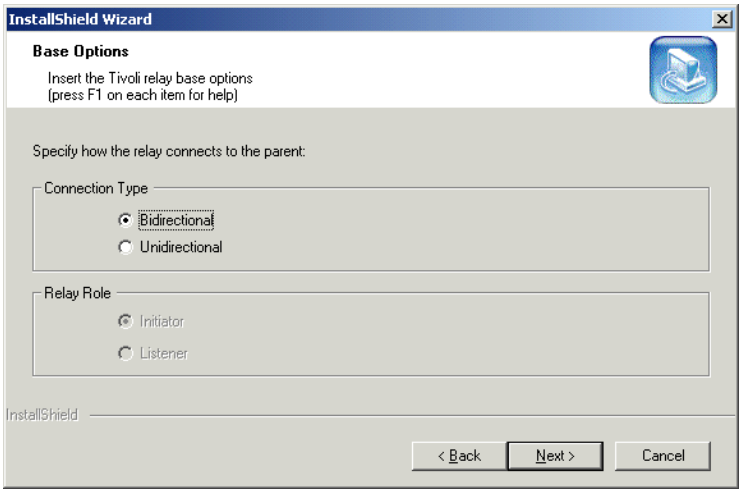
4. Complete the following fields regarding the connection between the relay and a parent machine, which can be either another relay or the endpoint proxy:

Relay Port
Enter the port number for the relay to communicate with the parent machine.

Relay or Endpoint Proxy Hostname
Enter the host name for the parent relay or endpoint proxy with which the relay will communicate.

Relay or Endpoint Proxy Port
Enter the port number for the parent relay or endpoint proxy with which the relay will communicate.

Click **Next**. The dialog for the type of relay-parent proxy connection is displayed.



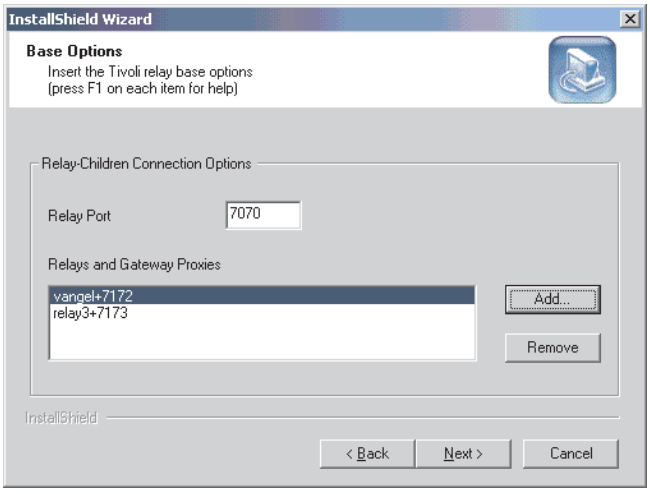
5. Specify how the relay connects to the parent relay or endpoint proxy:
   - Select **Bidirectional** to permit connections that are initiated by either machine.
   - Select **Unidirectional** to permit connections initiated by only one machine.

     If you select this option, the Relay Role box is enabled:

     Initiator
     The relay machine can start the connection with the parent machine.

     Listener
     The parent machine can start the connection with the relay machine.

   Click **Next**. The dialog for relay-child connection options is displayed.

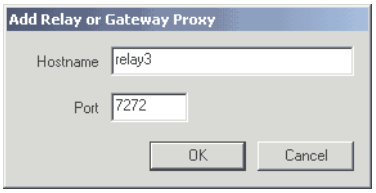


6. Complete the following fields:

   Relay Port
   Enter the port number for the relay to communicate with the children machines.

   Relays and Gateway Proxies
   Lists the relays and gateway proxies with which the relay connects.

To add a relay or gateway proxy to the list of destination machines, click **Add**. The Add Relay or Gateway Proxy dialog is displayed.



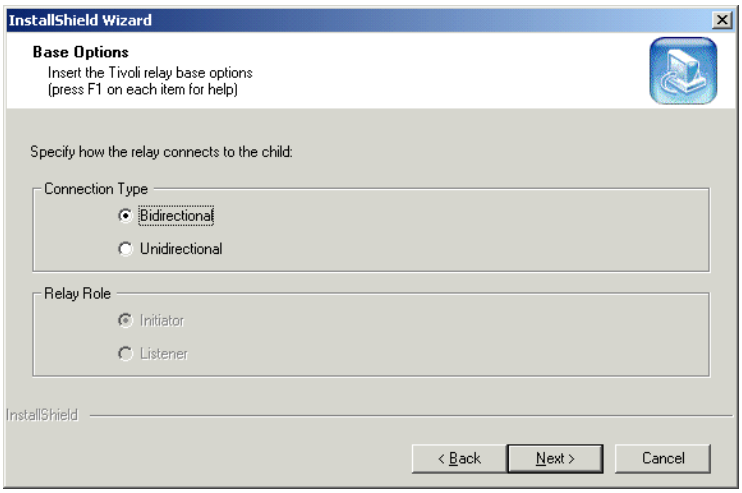
Complete the fields and click **OK**:

Hostname
Enter the host name of the child machine relay or gateway proxy with which the relay connects.

Port Enter the port number of the machine with which the relay connects.

To remove a machine, select it and click **Remove**.

Click **Next**. The dialog for the type of relay-child connection is displayed.



7. Specify how the relay connects to the child relay or gateway proxy:
   - Select **Bidirectional** to permit connections that are initiated by either machine.
   - Select **Unidirectional** to permit connections initiated by only one machine.

     If you select this option, the Relay Role box is enabled:

     Initiator
     The relay machine can start the connection with the child machines.

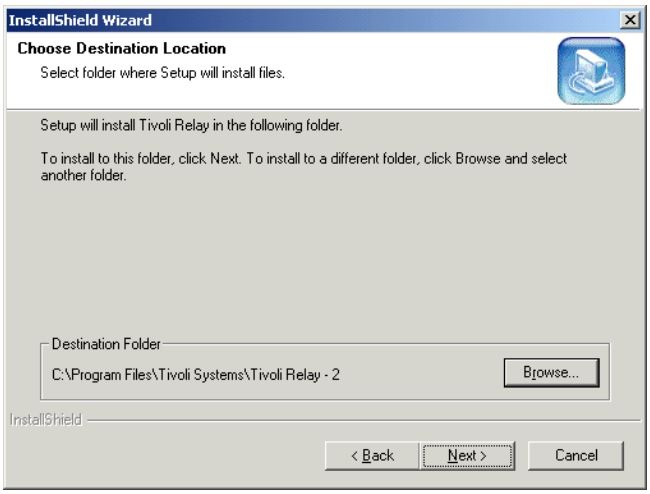
     Listener
     The child machines can start the connection with the relay machine.

   Click **Next**. The next dialog shows a summary of your input.
8. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the relay.
9. A message asks you whether or not you want to start the relay. Click **Yes** to start it. Otherwise, click **No**. To start the component at another time, see "Starting and Stopping the Components" on page 43.

Installing Additional Relays on the Same Machine

To install additional instances, do the following:

1. From the directory that contains the Tivoli Relay installation images, double-click the setup.exe file. InstallShield Wizard starts. Select **Install** and click **Next**. The Choose Destination Location dialog is displayed.



2. Enter the installation directory. Note that the default directory ends with a folder named for the numbered instance of the relay. This numbered name identifies this instance of the relay when you want to start, stop, and uninstall it.

   If you change the default directory, ensure that it is different from the directory where you have installed previous instances of the relay.

   Click **Next**.

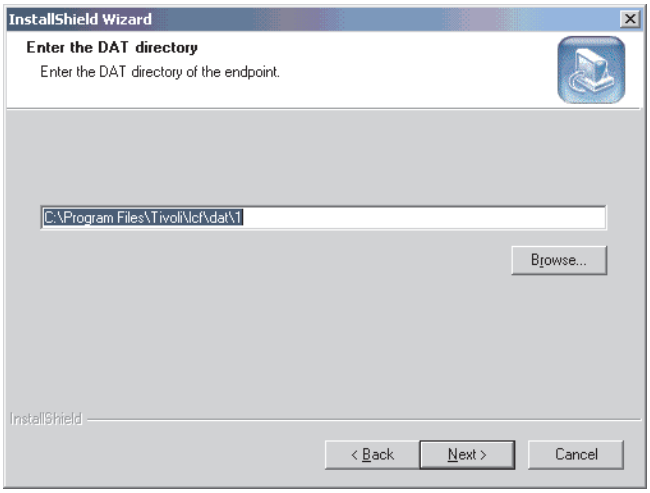
3. Continue with step 4 on page 19 in the previous procedure.

Installing the Event Sink on Windows Systems

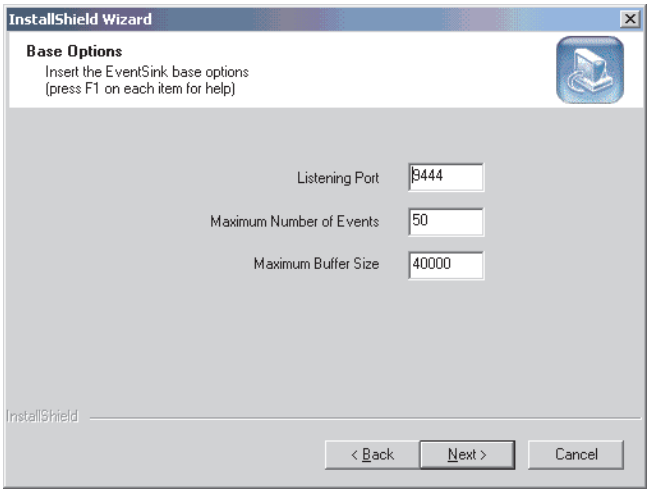
You must install the event sink on a endpoint. To install the event sink, do the following:

1. From the directory that contains the Event Sink\w32-ix86\ subdirectory, double-click the Tivoli EventSink.exe file. The Tivoli Event Sink InstallShield Wizard starts.
2. Click **Next**.

3. On the next dialog, click **Yes** if you accept the license agreement. The Destination Location dialog is displayed.



4. Specify the lcf_datdir directory of the endpoint on which you are installing the event sink. The event sink is installed in the lcf_datdir\..\..\bin\w32-ix86\mrt\ directory. Click **Next**. The dialog for Event Sink Options is displayed.



5. Complete the following fields:

   Port Enter the port number on the endpoint where the event sink will receive events. The default is 9444.

   Maximum Number of Events
   Enter the maximum number of events that the event sink will send to the Tivoli Enterprise Console server in a single package. The default is 50.

   Maximum Buffer Size
   Enter the maximum buffer size, in bytes, of the package that the event sink will send to the Tivoli Enterprise Console server. The default is 40000.

   Click **Next**. The next dialog shows a summary of your input.

6. To go back and make changes, click **Back**. Otherwise, click **Next** to continue. The program proceeds to install the event sink.

Upgrading the Components

The following sections describe how to upgrade the components from a previous version to the current version. There is no compatibility between versions 1.2 and 1.3 of the toolbox. When a component from version 1.3 detects a connecting component from version 1.2, it rejects the connection and logs the following error message:

```
ERROR multiplex.parseFrame: connecting peer is version 1.2.0
```

Upgrading on Windows Systems

To upgrade the component on a Windows machine, start the installation for the new version. The configuration and all the data of the previous version are saved and used by the new version.

Upgrading on UNIX Systems

To upgrade the component on a UNIX machine, run the script **upgrade.sh** from the directory that contains the installation images.

You are asked to enter the directory in which the previous version is installed. The configuration and all the data of the previous version are backed up to the directory that you specify with a suffix of **.1.2**. For example, if you specify `/usr/epp`, the backup directory becomes:

```
/usr/epp.1.2
```

Uninstalling the Components

The following sections describe how to uninstall the components.

Note: Under normal circumstances, you should not delete and reinstall an endpoint proxy. If the endpoint proxy is removed, all the dynamic configuration maintained in the `epproxy.bdb` file is also removed and lost. The reinstallation of the endpoint proxy cannot restore this information that is created during initial logins of endpoints. When you remove and reinstall the endpoint proxy, all endpoints that are connected to the endpoint proxy must do an initial login to the Endpoint Manager database as if they were new endpoints.

Make a backup copy of the `epproxy.bdb` file before uninstalling the endpoint proxy. After you reinstall the endpoint proxy, you can replace the `epproxy.bdb` file with your backup copy of the file. If you reinstall from the beginning, then you do not need to make a backup copy of the file. For more information, see "Backing Up and Restoring the Endpoint Manager Database" on page 46.

The installation files are not removed when you uninstall the components. If you want to remove them, delete the files by hand.

Uninstalling from UNIX Systems

To uninstall the component from a UNIX machine, run the script **uninstall.sh** from the directory in which the component is installed.

Uninstalling All Components except the Relay from Windows Systems

You can uninstall all the components except the relay from a Windows NT or Windows 2000 system in one of the following ways:

- If you are using the disk image, run setup.exe and then choose **Remove**.
- If you installed the component using InstallShield, double-click **Add/Remove Programs** from the **Control Panel**. Select the component to remove, for example, Tivoli Endpoint Proxy, from the list of currently installed programs, and click **Add/Remove** or **Change/Remove**.
- Start the InstallShield Wizard that you used to install. Select **Remove** and click **Next**.

  Note: If you are uninstalling one or more instances of a relay from the same machine, use this method.

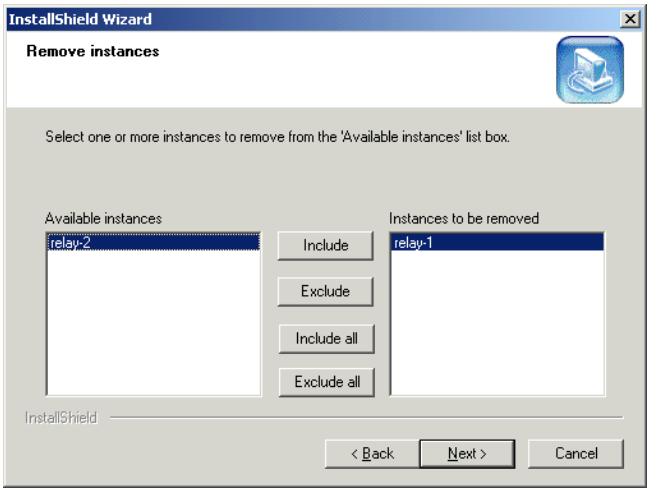
In addition, delete the CFG and LOG file from the directory in which the component is installed. The following table lists the name of each file for each component:

Endpoint Proxy	Event Sink	Gateway Proxy	Relay
epproxy.cfg	eventsink.cfg	gwproxy.cfg	relay.cfg
epp.log	eventsink.log	gwp.log	relay.log

Uninstalling the Relay from Windows Systems

To uninstall one or more instances of a relay from a Windows machine, perform the following steps:

1. Start the InstallShield Wizard that you used to install it. Select **Remove** and click **Next**. The Remove instances dialog is displayed. It lists the numbered relay instances that are installed on the machine.



2. From the Available instances list, select the instances that you want to remove and click **Include**, or click **Include all** to remove all the instances from the machine.

3. Click **Next** and confirm the deletion. The selected instances are deleted.

Note: The number of an instance that was uninstalled is recycled when you reinstall it. For example, if you uninstall relay-2, the next time you install an instance on the machine, the new instance is named relay-2.

Chapter 3. Configuring the Components

This chapter explains how to configure the components of the Tivoli Management Framework Firewall Security Toolbox.

Configuring the Endpoint Proxy

After you install the endpoint proxy, the configuration file epproxy.cfg is created in the folder in which you installed the proxy. It contains the configuration input that you supplied during installation. In addition, to configure other options, edit the epproxy.cfg file with a text editor.

Stop and start the component to make your changes effective.

The following sections are divided by the sections in the configuration file. Each section provides a table of the keywords and comments. Enter the values in the format:

```
keyword=value
```

The section titles are case-sensitive.

Endpoint-proxy

The [endpoint-proxy] section lists the main options for the endpoint proxy. Table 2 lists the keywords and a description.

Table 2. The endpoint-proxy section

Keyword	Description
gateway-host	The address and port number of the gateway on which it will listen for communication from the endpoint proxy as if it were the endpoint. The default port is 9494 and should not be changed unless the gateway is known to be using a different listening port with the endpoint. Use the format: `address+port_number`
gateway-interface	This option is used for multihomed endpoint proxies. It is the Domain Name System (DNS) name or IP address of the network interface of the endpoint proxies that is used to communicate with the gateway network interface.
accept-timeout	The timeout interval, in seconds, that the endpoint proxy waits for connections expected by the gateway. The default is 300.
max-sessions	The number of connections that the endpoint proxy can manage at the same time. The default is 75.
tcpip-timeout	The timeout interval, in seconds, for TCP/IP operations to succeed or fail. This controls outbound and inbound connection attempts between the proxy and children or parent. This timeout ensures that the proxy cannot be overloaded by denial-of-service attacks by clients who open connections, but never close them. The default is 240.

Table 2. The endpoint-proxy section (continued)

Keyword	Description
port-range	The port ranges to use when allocating endpoint ports and when connecting with the gateway. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: `6060,7000-7070,9050,8000-8080` The default is 6000-8000.
database-path	The full path to a directory where the endpoint proxy database is installed. The endpoint proxy database records information about endpoints that it manages, and it must remain in existence even when the endpoint proxy is restarted. The default is the directory where the component is installed.
disable-udp	Enables the endpoint proxy to forward login requests via TCP instead of datagram protocol (UDP). To enable TCP, specify 1. To enable UDP, specify 0. The default is 0.

Log

The [log] section lists log options. Table 3 lists the keywords and a description.

Table 3. The log section

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommended for normal operation. Levels higher than 3 should only be used when recommended by customer support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. The range is 0-11. The default is 3.
log-file	The full path of the log file where endpoint proxy messages are written. The installation default is epp.log. If no file specified, the default is standard error.
max-size	The maximum size, in megabytes, that the log file can reach. When the log file reaches the maximum size, it is renamed to *filename*.log.bak and a new log file is started. For no limit, specify 0. The default is 1.

Communication-layer

The [communication-layer] section lists options for how the endpoint proxy connects to its relays or gateway proxies. Table 4 lists the keywords and a description.

Table 4. The communication-layer section

Keyword	Description
children-local-host	The network interface (DNS name or IP address) on the endpoint proxy to listen for communication from its relays or gateway proxies.
children-local-port	The port number of the endpoint proxy machine from which it listens for connections with relays or gateway proxies.

Table 4. The communication-layer section (continued)

Keyword	Description
children-remote-list	The list of children hosts (relays or gateway proxies) to which the endpoint proxy connects. Separate the entries with a semi-colon (;) but leave the end of the line without the delimiter. For example, a relay with address 69.99.99.71 and port 7071 and a gateway proxy with address 69.99.99.80 and port 7073: `69.99.99.71+7071;69.99.99.80+7073`
children-remote-file	The name of the optional file containing a list of children hosts (relays or gateway proxies) to which the endpoint proxy connects. This keyword can be listed in addition to or instead of the children-remote-list keyword.
children-cm-type	The communication interface (TCP/IP) and the connectivity (unidirectional or bidirectional) that the endpoint proxy uses with its relays or gateway proxies. The values are cm-tcp-bidirectional, cm-tcp-unidirectional.

Notes:

1. You must include at least one children host, specified either in the children-remote-list keyword or in the file listed for the children-remote-file keyword.
2. In the file specified for the children-remote-file keyword, list the children hosts on separate lines in the *host_name+port_number* format. Precede comments in the file with the number sign (#). You can leave blank lines as shown in the following example:

```
#Add these children hosts

luna+10023
sol+29993
```

Children-cm-info

The [children-cm-info] section lists further options about connectivity between the endpoint proxy and its children (relays or gateway proxies). Table 5 lists the keywords and a description.

Table 5. The children-cm-info section

Keyword	Description
connection-mode	The role of the endpoint proxy in unidirectional connections only. The values are initiator or listener. The default is listener. **Note:** The values from Version 1.2 client (initiator) and server (listener), are still valid values.
local-port-range	The range of local ports to be used when connecting to the other peer. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: `6060,7000-7070,9050,8000-8080`
receive-buffer-size	The size (in kilobytes) of the buffer that is used to receive communications from a TCP/IP socket. The minimum value is 1. The default is 17.
connect-timeout	The timeout interval, in seconds, after which a TPC/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0.

Table 5. The children-cm-info section (continued)

Keyword	Description
send-timeout	The timeout interval, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. The default is 120.
log-mode	The buffer that is sent or received during peer connection. Specifying a value other than 0 can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. The default is 0.
drop-timeout	For bidirectional connections only, the number of seconds after which an inactive connection is closed. To close the connection as soon as the usage counter drops to zero, specify 0. The default is 5.
polling-interval	In unidirectional connections for initiator components only, the initiator polls the listener periodically to check if it needs to establish a connection. This option is the interval, in seconds, after which an initiator automatically connects to the listener. The default is 2.
drop-interval	In unidirectional connections for listener components only, the interval, in seconds, after which the listener suspends an inactive connection. The default is 5.

Configuring the Gateway Proxy

After you install the gateway proxy, the configuration file gwproxy.cfg is created in the folder in which you installed the proxy. It contains the configuration input that you supplied during installation. In addition, to configure other options, edit the gwproxy.cfg file with a text editor.

Stop and start the component to make your changes effective.

The following sections are divided by the sections in the configuration file. Each section provides a table of the keywords and comments. Enter the values in the format:

```
keyword=value
```

The section titles are case-sensitive.

Gateway-proxy

The [gateway-proxy] section lists the main options for the gateway proxy. Table 6 lists the keywords and a description.

Table 6. The gateway-proxy section

Keyword	Description
gateway-port	The port number on the gateway proxy that represents the gateway to the endpoints. The default is 9494.
gateway-interface	This option is used for multihomed gateway proxies. It is the DNS name or IP address of the gateway proxies network interface used to communicate with the Tivoli endpoints.

Table 6. The gateway-proxy section (continued)

Keyword	Description
tcpip-timeout	The timeout interval, in seconds, for TCP/IP operations to succeed or fail. This controls outbound and inbound connection attempts between the proxy and parent or endpoints. This timeout ensures that the proxy cannot be overloaded by denial-of-service attacks by clients who open connections, but never close them. The default is 240.
proxy-label	Optional name to identify the gateway proxy instance. The default is the *host_name*.
max-sessions	Number of connections that the gateway proxy can manage at the same time. The default is 75.
port-range	The port ranges to use when allocating ports to connect to endpoint ports. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: `6060,7000-7070,9050,8000-8080` The default is 6000-8000.

Log

The [log] section lists log options. Table 7 lists the keywords and a description.

Table 7. The log section

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommended for normal operation. Levels higher than 3 should only be used when recommended by Tivoli Customer Support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. The range is 0-11. The default is 3.
log-file	The full path of the log file where gateway proxy messages are written. The installation default is gwp.log. If no file specified, the default is standard error.
max-size	The maximum size, in megabytes, that the log file can reach. When the log file reaches the maximum size, it is renamed to *filename*.log.bak and a new log file is started. For no limit, specify 0. The default is 1.

Communication-layer

The [communication-layer] section lists options for how the gateway proxy connects to its relay or endpoint proxy. Table 8 lists the keywords and a description.

Table 8. The communication-layer section

Keyword	Description
parent-local-host	The DNS name or IP address of the gateway proxy from which it listens for connections from its relay or endpoint proxy.
parent-local-port	The port number of the gateway proxy machine from which it listens for connections with relay or endpoint proxy.
parent-remote-host	The DNS name or IP address of the relay or endpoint proxy.

Table 8. The communication-layer section (continued)

Keyword	Description
parent-remote-port	The port number of the parent relay or endpoint proxy from which it listens for connections with gateway proxy.
parent-cm-type	The communication interface and the connectivity that the gateway proxy uses with its relay or endpoint proxy. Values: cm-tcp-bidirectional, cm-tcp-unidirectional.

Parent-cm-info

The [parent-cm-info] section lists further options about connectivity between the gateway proxy and its parent (relay or endpoint proxy). Table 9 lists the keywords and a description.

Table 9. The parent-cm-info section

Keyword	Description
connection-mode	The role of the gateway proxy in unidirectional connections only. The values are initiator or listener. The default is listener. **Note:** The values from Version 1.2 client (initiator) and server (listener) are still valid.
local-port-range	The range of the local ports to be used when connecting to the other peer. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: `6060,7000-7070,9050,8000-8080`
receive-buffer-size	The size (in kilobytes) of the buffer that is used to receive communications from a TCP/IP socket. Minimum value: 1. The default is 17.
connect-timeout	Timeout, in seconds, after which a TPC/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0.
send-timeout	Timeout, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. The default is 120.
log-mode	The buffer that is sent or received during peer connection. Specifying a value other than 0 can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. The default is 0.
drop-timeout	For bidirectional connections only, number of seconds after which an inactive connection is closed. To close the connection as soon as the usage counter drops to zero, specify 0. The default is 5.
polling-interval	In unidirectional connections for initiator components only, the initiator polls the listener periodically to check if it needs to establish a connection. Interval, in seconds, after which an initiator automatically connects to the listener. The default is 2.
drop-interval	In unidirectional connections for listener components only, interval, in seconds, after which the listener suspends an inactive connection. The default is 5.

Configuring the Relay

After you install the relay, the configuration file relay.cfg is created in the folder in which you installed the component. It contains the configuration input that you supplied during installation. In addition, you can configure other options. To change these or configure other options, edit the relay.cfg file with a text editor.

Stop and start the component to make your changes effective.

The following sections are divided by the sections in the configuration file. Each section provides a table of the keywords and comments. Enter the values in the format:

```
keyword=value
```

The section titles are case-sensitive.

Relay

The [relay] section is required at the top of the file, even when you do not specify any keywords. Table 10 lists the keyword and a description.

Table 10. The relay section

Keyword	Description
tcpip-timeout	The timeout interval, in seconds, for TCP/IP operations to succeed or fail. This controls outbound and inbound connection attempts between the relay and children or parent. This timeout ensures that the relay cannot be overloaded by denial-of-service attacks by clients who open connections, but never close them. The default is 240

Log

The [log] section lists log options. Table 11 lists the keywords and a description.

Table 11. The log section

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommended for normal operation. Levels higher than 3 should only be used when recommended by Tivoli Customer Support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. The range is 0-11. The default is 3.
log-file	The full path of the log file where relay messages are written. Installation The default is relay.log. If no file specified, the default is standard error.
max-size	The maximum size, in megabytes, that the log file can reach. When the log file reaches the maximum size, it is renamed to *filename*.log.bak and a new log file is started. For no limit, specify 0. The default is 1.

Communication-layer

The [communication-layer] section lists options for how the relay connects to its parent and children, relays, endpoint proxy, or gateway proxy. Table 12 on page 34 lists the keywords and a description.

Table 12. The communication-layer section

Keyword	Description
parent-local-host	The DNS name or IP address of relay from which it listens for connections from its parent relay or endpoint proxy.
parent-local-port	The port number of the relay machine from which it listens for connections with parent relay or endpoint proxy.
parent-remote-host	The DNS name or IP address of the parent relay or endpoint proxy.
parent-remote-port	The port number of the parent relay or endpoint proxy from which it listens for connections with relay.
parent-cm-type	Communication interface and the connectivity that the relay uses with its parent relay or endpoint proxy. Values: cm-tcp-bidirectional, cm-tcp-unidirectional.
children-local-host	DNS name or IP address of relay from which it listens for connections from children relays or gateway proxies.
children-local-port	The port the relay listens on for traffic from children relays or gateway proxies.
children-remote-list	List of children hosts (relays or gateway proxies) to which the relay connects. Separate the entries with a semi-colon (;) but leave the end of the line without the delimiter. For example, a relay with address 69.99.99.71 and port 7071 and a gateway proxy with address 69.99.99.80 and port 7073: `69.99.99.71+7071;69.99.99.80+7073`
children-cm-type	Communication interface and the connectivity that the relay uses with its relays or gateway proxies. Values: cm-tcp-bidirectional, cm-tcp-unidirectional.
children-remote-file	The name of the optional file containing a list of children hosts (relays or gateway proxies) to which the relay connects. This keyword can be listed in addition to or instead of the children-remote-list keyword.

Notes:

1. You must include at least one children host, specified either in the children-remote-list keyword or in the file listed for the children-remote-file keyword.
2. In the file specified for the children-remote-file keyword, list the children hosts on separate lines in the *host_name+port_number* format. Precede comments in the file with the number sign (#). You can leave blank lines as shown in the following example:

```
#Add these children hosts

luna+10023
sol+29993
```

Children-cm-info

The [children-cm-info] section lists further options about connectivity between the relay and its children (relays or gateway proxies). Table 13 on page 35 lists the keywords and a description.

Table 13. The children-cm-info section

Keyword	Description
connection-mode	The role of relay in unidirectional connections only. The values are initiator or listener. The default is listener. **Note:** The values from Version 1.2 client (initiator) and server (listener) are still valid.
local-port-range	The range of local ports to be used when connecting to the other peer. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: `6060,7000-7070,9050,8000-8080`
receive-buffer-size	The size (in kilobytes) of the buffer that is used to receive communications from a TCP/IP socket. The minimum value is 1. The default is 17.
connect-timeout	The timeout interval, in seconds, after which a TPC/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0.
send-timeout	The timeout interval, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. The default is 120.
log-mode	The buffer that is sent or received during peer connection. Specifying a value other than 0 can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. The default is 0.
drop-timeout	For bidirectional connections only, number of seconds after which an inactive connection is closed. To close the connection as soon as the usage counter drops to zero, specify 0. The default is 5.
polling-interval	In unidirectional connections for initiator components only, the initiator polls the listener periodically to check if it needs to establish a connection. Interval, in seconds, after which an initiator automatically connects to the listener. The default is 2.
drop-interval	In unidirectional connections for listener components only, interval, in seconds, after which the listener suspends an inactive connection. The default is 5.

Parent-cm-info

The [parent-cm-info] section lists further options about connectivity between the relay and its parent (relay or endpoint proxy). Table 14 lists the keywords and a description.

Table 14. The parent-cm-info section

Keyword	Description
connection-mode	Role of relay in unidirectional connections only. Values: initiator or listener. The default is listener. **Note:** The values client (initiator) and server (listener), which were accepted values for Version 1.2, remain valid.
local-port-range	Range of local ports to be used when connecting to the other peer. For example, ports 6060, 7000 to 7070, 9050, and 8000 to 8080: `6060,7000-7070,9050,8000-8080`
receive-buffer-size	Size of buffer, in kilobytes, used to receive from a TCP/IP socket. Minimum value: 1. The default is 17.

Table 14. The parent-cm-info section (continued)

Keyword	Description
connect-timeout	Timeout, in seconds, after which a TPC/IP connect operation fails. To disable the timeout and use the default of the TCP/IP library, specify 0.
send-timeout	Timeout, in seconds, after which a TCP/IP send operation fails. To disable the timeout, specify 0. The default is 120.
log-mode	The buffer that is sent or received during peer connection. Specifying a value other than 0 can lower performance. Values: 0=none 1=sent data only 2=received data only 3=all transmitted data. The default is 0.
drop-timeout	For bidirectional connections only, number of seconds after which an inactive connection is closed. To close the connection as soon as the usage counter drops to zero, specify 0. The default is 5.
polling-interval	In unidirectional connections for initiator components only, the initiator polls the listener periodically to check if it needs to establish a connection. Interval, in seconds, after which an initiator automatically connects to the listener. The default is 2.
drop-interval	In unidirectional connections for listener components only, interval, in seconds, after which the listener suspends an inactive connection. The default is 5.

Configuring the Event Sink

After you install the event sink, the configuration file eventsink.cfg is created in the folder in which you installed the component. It contains the configuration input that you supply during installation.

Note: You must also configure every generator of non-secure events in your environment to send events to the event sink and not to the Tivoli Enterprise Console server. To configure non-TME adapters, see "Configuring Non-TME Adapters for the Event Sink" on page 38. To configure Availability Intermediate Manager Console server see "Processing Events from the Tivoli Enterprise Console Availability Intermediate Manager Console" on page 47.

To configure other options, edit the eventsink.cfg file with a text editor.

Stop and start the component to make your changes effective.

The following sections are divided by the sections in the configuration file. Each section provides a table of the keywords and comments. Enter the values in the format:

```
keyword=value
```

The section titles are case-sensitive.

SENDING

The [SENDING] section lists options for sending events to the Tivoli Enterprise Console server. Table 15 on page 37 lists the keywords and a description.

Table 15. The SENDING section

Keyword	Description
lcf-datdir	The `dat` directory of the endpoint.
max-size-buffer	The maximum buffer size, in bytes, of the package that the event sink sends to the Tivoli Enterprise Console server. The default is 40000.
max-num-events-to-send	The maximum number of events that the event sink sends to the Tivoli Enterprise Console server in a single package. The default is 50.
delay-time	The minimum interval, in seconds, between sending packages of events to the Tivoli Enterprise Console server. To send packages immediately, specify 0. The default is 1.
caching-timeout	The timeout interval, in seconds, by which the event sink sends events if neither the maximum buffer size nor the maximum number of events is reached. The default is 30.

RECEPTION

The [RECEPTION] section lists options for receiving events from non-TME adapters. Table 16 lists the keywords and a description.

Table 16. The RECEPTION section

Keyword	Description
port	The port number on the endpoint where the event sink receives events. The default is 9444.
max-sessions	The maximum number of threads that the event sink can have with the non-TME adapters. Set this value at least to equal the number of non-TME adapters with which the event sink communicates. The default is 100.
tcpip-timeout	The timeout interval, in seconds, for TCP/IP operations to succeed or fail. This controls outbound and inbound connection attempts between the event sink and non-TME adapters. This timeout ensures that the event sink cannot be overloaded by denial-of-service attacks by clients who open connections, but never close them. The default is 240.
max-ram-cache	The amount of memory, RAM, in kilobytes, in which events get held on the event sink machine. The event sink stops receiving events when this value is reached, until it sends the events to the Tivoli Enterprise Console server. The default is 1024.
caching-timeout	The timeout value, in seconds, by which the event sink sends events if neither the maximum buffer size nor the maximum number of events is reached. The default is 30.

EIF

The [EIF] section lists options for the Tivoli Enterprise Integration Facility.

You can optionally have the event sink forward non-TME events to the Tivoli Enterprise Console server as if it were a Tivoli Enterprise Console adapter. To do this, you configure the EIF parameters here in this section as you would for a normal adapter. See the *Tivoli Event Integration Faciliy: User's Guide* for the keywords, formats, and values that apply.

Table 17 lists only the keywords and a description for the parameters that are required for the Tivoli Management Framework Firewall Security Toolbox, except for the ServerLocation parameter, which you must specify only if you are configuring the event sink to be a Tivoli Enterprise Console adapter.

Table 17. The EIF section

Keyword	Description
BufEvtMaxSize	The maximum size, in kilobytes, of the eventsink.cache file. The default is 64.
BufEvtPath	The file in which the event sink saves events that it temporarily cannot send. By default, this file is created in the installation directory on UNIX and in the dat directory on Windows machines. The default is eventsink.cache.
ServerLocation	Optional. Enables you to bypass the Firewall Security Toolbox proxies by specifying the name of the host on which an event server is installed. To configure the event sink to work as a Tivoli Enterprise Console adapter, enter the *IP_address*+*port* of the Tivoli Enterprise Console server. To forward events through the normal Tivoli channels, use the default value. The default is @EventServer.

LOG

The [log] section lists log options. Table 18 lists the keywords and a description.

Table 18. The log section

Keyword	Description
debug-level	The level of detail in the log. Levels 0 and 1 are recommended for normal operation. Levels higher than 3 should only be used when recommended by customer support for diagnosing problems. A level higher than 6 will noticeably impact the performance of the service and should be used with discretion. The range is 0-11. The default is 3.
log-file	The full path of the log file where event sink messages are written. The default for Windows systems is: */DAT_directory*/eventsink.log. The default for UNIX systems is: */installation_directory*/eventsink.log. If no file is specified, the default is standard error.
max-size	The maximum size, in megabytes, that the log file can reach. For no limit, specify 0. The default is 1.

Configuring Non-TME Adapters for the Event Sink

To configure the non-TME adapter to send events to the event sink and not to the Tivoli Enterprise Console server, edit the configuration file on the non-TME adapter and change the following parameters:

```
ServerLocation=host_name
ServerPort=port
```

Where:

host_name
: The host name of the endpoint on which the event sink is installed

port
: The port on which the event sink listens for events.

Migrating Endpoints to Connect to a Gateway Proxy

To migrate an endpoint from a Tivoli gateway *to* a gateway proxy, change the login interfaces of the endpoint by specifying gateway proxies in the list of interfaces. Do one of the following:

- Use the **wep set interfaces** command:
  1. From the Tivoli management region server (Tivoli server) or managed node, enter the command:

     ```
     wep set interfaces -e ep_label host_name_gwp+port
     ```

     Where:

     ep_label
     Is the label of the endpoint

     host_name_gwp
     Is the host name of the gateway proxy

     port Is the port number of the gateway proxy
  2. Enter the command:

     ```
     wep sync_gateways
     ```
  3. Stop the endpoint.
  4. Put up the firewall between the endpoint and the gateway that managed it previously.
  5. Restart the endpoint.
- Use the HTTP interface of the endpoint:
  1. Update the login interfaces and gateway to point to one or more gateway proxies.
  2. Use a Web browser to update each endpoint.
  3. From the Network Address Configuration menu, use the **lcfd –g** command to set the gateway. Additionally or alternatively, use the **–D lcs.login_interfaces** argument. Note that you must know the HTTP user name and password for the endpoint. See the *Planning for Deployment Guide* for details.
  4. Put up the firewall between the endpoint and the gateway that managed it previously.
  5. Restart the endpoint.

The following commands are *not* supported for endpoints and gateways that have the firewall proxies between them:

- **wep migrate**
- **select_gateway_policy**

To migrate an endpoint from a gateway proxy *to* a Tivoli gateway, use the **wep migrate** command as you would normally.

Configuring Backup Gateway Proxies

You can set up the components to include more than one gateway proxy in a DMZ so that if a gateway proxy is down, the endpoint proxy can use an alternative gateway proxy to reach an endpoint. This process of looking for an alternative gateway proxy is called a *gateway proxy failover*.

To set up alternative gateway proxies, you create groups of gateway proxies that the endpoint proxy tries to use in the order that you specify.

Do the following:

1. Create a file named proxy.grp in the directory where the *endpoint proxy* is installed. The account with which the endpoint proxy runs must have the permissions to read the file.
2. In the proxy.grp file, include a single line entry for each group of gateway proxies that you want to create. For example:

   ```
   group1: a b c
   group2: a d e f
   ```

   Where `group1` and `group2` are the names of groups of gateway proxies. The letters `a` through `f` are the labels of gateway proxies.

   Follow each group name with a colon (:). The group names can be whatever you like as long as each name is unique in the file.

   List each gateway proxy in the order in which the endpoint proxy should search for it. Use the gateway proxy label that is specified either at installation or in the configuration file of the gateway proxy. If gateway proxy 'a' is down, the endpoint proxy tries 'b'. If 'b' is down, it tries 'c'.

   You can specify the same gateway proxy in more than one group, for example, `a` in both group1 and group2. When gateway proxy `a` fails, the endpoint proxy will try all the gateway proxies in the groups that contain gateway proxy `a`.

Configuring Endpoints for Backup Gateway Proxies

In addition to specifying the list of backup gateway proxies for the endpoint proxy, you must configure the endpoint to connect to a specific list of gateway proxies when the main gateway proxy is unavailable. Edit the login_policy script to configure it to work with the toolbox. Refer to the *Reference Manual* for more information about the login_policy script.

Add the following to the login_policy script:

- List the gateway proxies for the endpoint in the following format:

  gateway_proxy1`+`*port1*`:`*gateway_proxy2*`+`*port2*

  gateway_proxy1+*port1*
  : Indicates the host name or IP address of the main gateway proxy and its port number.

  gateway_proxy2+*port2*
  : Indicates the host name or IP address of the alternative gateway proxy and its port number.

- The **wep set interfaces** command

  For example:

  `wep set interfaces -e $1` *`gwp_list`*

  Where:

  $1 Is the endpoint label as defined in the login policy.

gwp_list
Is the list of gateway proxies as you defined them in your logic in the login policy.

For new endpoints: When you first install an endpoint, specify the gateway proxy and its ports.

The login policy that you defined will be run and the login interfaces will be updated to add backup gateways.

For existing endpoints: To start the login policy for the endpoints, enter the following command:

```
wadminep endpoint_label reexec_lcfd
```

Where *endpoint_label* is the label of the endpoint.

Setting the Endpoint Proxy Login Interval on All Platforms

The Tivoli endpoint manager prevents multiple logins from the same IP address within a specified interval. The gateways in the Tivoli management region (Tivoli region) share the interval. Using a single IP address and multiple ports, the endpoint proxy represents each endpoint to the gateway. To allow endpoints (and therefore the endpoint proxy) to login with the same IP address, set the login interval value to 0 seconds in the Tivoli endpoint manager configuration.

You need to have authorization to run this command and you need to set up the environment by running the `setup_env.sh` script from a Bourne shell prompt (`$BINDIR/tools/bash.exe` on Windows NT).

For Tivoli Management Framework Versions 3.7 and 4.1, do the following:

1. Type:
   ```
   wepmgr set login_interval=0
   ```
2. Restart the Tivoli region:
   ```
   odadmin reexec all
   ```

Chapter 4. Using Firewall Security Toolbox

This chapter describes how to work with the Firewall Security Toolbox in your environment.

Starting and Stopping the Components

If you did not start the components when you installed them, you need to start them to use the Firewall Security Toolbox.

Starting and Stopping the Components on Windows Systems

On Windows systems, do the following:

1. Open Services from the Control Panel.
2. Select the component from the list of services:
   - Tivoli Endpoint Proxy
   - Tivoli Event Sink
   - Tivoli Gateway Proxy
   - Tivoli Relay. If you have more than one instance of the relay installed on the machine, select the numbered instance that you want to start.
3. Select **Start** from the pop-up menu.

To stop the components on Windows, select **Stop** from the pop-up menu instead.

Starting and Stopping the Components on UNIX Systems

On UNIX systems, to start the components, do the following:

1. Go to the directory in which the component is installed.
2. Enter the command:

   ```
   ./component.sh start
   ```

   Where *component* stands for:

   epproxy
   The endpoint proxy

   eventsink
   The event sink

   gwproxy
   The gateway proxy

   relay The relay. If you have more than one instance of the relay installed on the machine, specify the directory numbered for the instance of the relay to be started or stopped.

To stop the components on UNIX, enter the command from the directory in which the component is installed:

```
./component.sh stop
```

Working with Endpoints Logged in through the Proxy

Endpoints that log in with the Tivoli management region server (Tivoli server) through the endpoint proxy are recorded in the endpoint proxy database (epproxy.bdb). To work with these endpoints, use the command `wproxy`. Before using the `wproxy` command, ensure the following:

- That you have logged on with the account with which the endpoint proxy runs. You must use the same account that was specified to run the endpoint proxy.
- That you have set up the shell environment by running setup_env.sh from the directory in which the endpoint proxy is installed.

Listing the Endpoints in the Database

To list the endpoints in the endpoint proxy database, enter the following command:

```
wproxy db [-d db_directory] ls [odnum...]
```

Where `db_directory` indicates the directory in which the database is stored.

The results appear in the following format:

```
odnum=identifier address=IP_address proxy_port=port1 real_port=port2  \
gwp_label=gateway_proxy
```

where:

identifier
: Indicates the number assigned to the endpoint by the Tivoli Management Framework.

IP_address
: Indicates the address of the endpoint.

odnum
: Indicates the number assigned to the endpoint by Tivoli Management Framework.

port1
: Indicates the port that the endpoint proxy assigns to the endpoint after the endpoint logs in for the first time. It is a port that the endpoint proxy uses to pose as the endpoint to the gateway.

port2
: Indicates the endpoint port that the gateway proxy uses to communicate with the endpoint.

gateway_proxy
: Indicates the label of the gateway proxy with which the endpoint connects.

See “wproxy” on page 50 for more details.

Modifying the Attributes of an Endpoint

You can modify the attributes of an endpoint, for example, the gateway proxy with which it connects.

To modify one or more attributes of an endpoint, enter the following command:

```
wproxy db edit odnum attribute=value...
```

Where:

odnum
: Indicates the number assigned to the endpoint by Tivoli Management Framework.

attribute

Indicates the attributes that you can modify:

proxy_port

Indicates the port that the endpoint proxy assigns to the endpoint after the endpoint logs in for the first time. It is a port that the endpoint proxy uses to pose as the endpoint to the gateway.

real_port

Indicates the endpoint port that the endpoint uses to listen for connections from the gateway proxy. It is the same port as the lfcd_port in the last.cfg file.

gwp_label

Indicates the label of the gateway proxy with which the endpoint connects.

value Indicates the new value that you want to assign to the attribute.

See "wproxy" on page 50 for more details.

Reassigning an Endpoint to a New Gateway Proxy

By modifying the gateway proxy label for an endpoint, you can reassign an endpoint to a new gateway proxy to balance the load on the gateway proxies in your configuration. Perform the following main steps:

1. Run the **wep set interface** command.
2. Run the **wproxy db edit** command (see "Modifying the Attributes of an Endpoint" on page 44 for the syntax) and specify the new gwp_label value.
3. To update the gateway settings in the lcf.dat file of the endpoint, run the **wep ep_label status** command.

See "wproxy" on page 50 for more details.

Changing the Port on the Endpoint

If you change the port that is used by the endpoint to listen for connections from the gateway (for example, if you restarted the endpoint after editing the lcfd_port value in the last.cfg file) the real port value could get out of sync with the lcfd_port value. You can bring the ports back in sync by modifying the real port value.

Run the **wproxy db edit** command and specify the new real_port value.

See "wproxy" on page 50 for more details.

Removing an Endpoint from the Database

When endpoints are deleted from a Tivoli management region (Tivoli region), they are not automatically deleted from the endpoint proxy database. You must remove them manually. To remove one or more endpoints from the endpoint proxy database, enter the following command:

```
wproxy db -d db_directory remove odnum
```

Where:

db_directory

Indicates the directory in which the database is stored.

odnum Indicates the number assigned to the endpoint by Tivoli Management Framework.

See “wproxy” on page 50 for more details.

Backing Up and Restoring the Endpoint Manager Database

When you back up the Endpoint Manager database using the **wbkupdb** command, the endpoint proxy database is not backed up. To keep a version of the endpoint proxy database that reflects the state of the endpoints when you backed up the Endpoint Manager database, make a copy of the epproxy.bdb file and store it with the output file of the **wbkupdb** command. If you restore the Endpoint Manager database, stop the endpoint proxy, copy the backup epproxy.bdb file to the directory in which the endpoint proxy is installed, then restart the endpoint proxy.

Installing Endpoints in a DMZ

The following sections describe how to install or configure endpoints in a DMZ.

Installing the Endpoints from Scratch

Installing UNIX endpoints using the winstlcf command across a firewall is supported only if the firewall allows rexec traffic to pass. This is probably not enabled in a production environment. Recommendations for alternative methods follow:

- Shut down the firewall for the time necessary to install and configure the endpoints using the winstlcf command, and then enable it again when the endpoints are ready to run.
- Open the firewall to permit the winstlcf command to be sent through the rexec port on UNIX endpoints.
- Create a separate Tivoli region in the DMZ for installing endpoints only. Shut down the Tivoli region after installation and leave it dormant unless you need to install other endpoints. Then, configure the endpoints to communicate with the gateway proxy.

Connecting Endpoints that Are Already Present in the Tivoli Region

To connect endpoints that were connected to a gateway to a gateway proxy instead, do the following:

1. Delete the endpoint from the Tivoli region to which it is connected using the **wdelep** command.
2. Stop the endpoint.
3. Reconfigure the endpoint:
   a. From the endpoint DAT directory, delete all *except* the following files:
      - last.cfg
      - lcf_env.csh
      - lcf_env.sh
      - lcfd.sh
   b. Edit the last.cfg file:
      - Change the `gateway_port` entry to the following:

        `gateway_port=`*`gateway_proxy_port`*

        Where *gateway_proxy_port* is the port number of the gateway proxy to which you are migrating the endpoint.

- Add the following entry:

  `lcs.login_interfaces=`*`gateway_proxy_host_name+port`*

  Where *gateway_proxy_host_name+port* is the host name and the port number of the gateway proxy.

4. Save and close the file, and restart the endpoint.

Existing installation methods that are not based on remote access work as well.

Processing Events from the Tivoli Enterprise Console Availability Intermediate Manager Console

If you use the Tivoli Enterprise Console Availability Intermediate Manager Console, you need to configure it to work with the Firewall Security Toolbox. When there is a firewall between the machine with the Tivoli Enterprise Console Availability Intermediate Manager Console and the Tivoli Enterprise Console server, and you need to send events to Tivoli Enterprise Console, you must send events to the event sink instead of to the Tivoli Enterprise Console server directly. The event sink then forwards the events to the Tivoli Enterprise Console server across the firewalls.

If the Tivoli Enterprise Console Availability Intermediate Manager Console is set up to send the events to the Tivoli Enterprise Console server to be processed, do the following:

1. Customize the action "Send a TEC Event to a TEC Server." The Customize Action dialog is displayed.
2. In the **IP Address or Hostname** text box, enter the host name or address of the event sink.
3. In the **Server Port** text box, enter the port of the event sink.
4. Click **Save Event Action**.
5. Distribute the Event Action Plan to the Tivoli Enterprise Console Availability Intermediate Manager.

If you have a rule base that processes the events on the Tivoli Enterprise Console Availability Intermediate Manager Console and forwards them to the Tivoli Enterprise Console server, do the following:

1. In the /dat/default_rb/TEC_Rules/ directory where the console is installed, edit the tec_forward.conf file:
   - For the ServerLocation entry, specify the host name of the event sink.
   - For the ServerPort entry, specify the port number of the event sink.
   - For the TestMode entry, specify **No**.

   Save and close the file.
2. Customize the action "Send a Tivoli Enterprise Console Event to a Tivoli Enterprise Console Server." The Customize Action dialog is displayed.
3. In the **IP Address or Hostname** text box, enter the host name or address of the Tivoli Enterprise Console Availability Intermediate Manager.
4. In the **Server Port** text box, enter the port of the Tivoli Enterprise Console Availability Intermediate Manager.
5. Click **Save Event Action**.
6. Distribute the Event Action Plan to the Tivoli Enterprise Console Availability Intermediate Manager.

Viewing Endpoint Properties

You can view the properties of an endpoint using a Web browser by entering the host name and port number of the endpoint in the **Location** text box.

When the endpoint is connected to the Tivoli environment through the Tivoli Firewall Security Toolbox proxies, enter the following URL in the **Location** text box:

`http://host_name:port_number`

Where:

host_name
: Indicates the host name of the endpoint proxy, not the endpoint.

port_number
: Indicates the port number that the endpoint proxy uses to pose as the endpoint. To get the port number, you can, for example, view the list of endpoints for the gateway from the gateway list and select the endpoint.

Appendix A. Using the Command Line Interface

This appendix describes the **wproxy** command.

Command Line Syntax

The commands described in this book use the following special characters to define the syntax of commands:

[ ] Identifies optional attributes. Attributes not enclosed in brackets are required.

... Indicates that you can specify multiple values for the previous attribute.

| Indicates mutually exclusive information. You can use the attribute to the left of the separator or the attribute to its right. You cannot use both attributes in a single use of the command.

{ } Delimits a set of mutually exclusive attributes when one of the attributes is required. If the attributes are optional, they are enclosed in square brackets ([ ]).

**** Indicates that the syntax in an example wraps to the next line.

The options for each command are listed alphabetically in the Options section, unless the options must be used in a specific order to implement the command.

wproxy

Works with endpoints in the endpoint proxy database and displays the version of the component installed.

Syntax

wproxy db [**–d** *db_directory*] **edit** *odnum* [*attribute*=*value*]

wproxy db [**–d** *db_directory*] **ls** [*odnum* ...]

wproxy db [**–d** *db_directory*] **remove** [*odnum* ...]

wproxy –v

Description

The **wproxy** command is run at the endpoint proxy. It lists endpoints, modifies attributes of an endpoint, and removes one or more endpoints from the endpoint proxy database.

Before launching the **wproxy** command on UNIX operating systems, move to the directory where the endpoint proxy is installed and enter the following command:

```
. ./setup_env.sh
```

Options

The options of the **wproxy** command follow:

attribute
: Indicates the attributes that you can modify:

  proxy_port
  : Indicates the port that the endpoint proxy assigns to the endpoint after the endpoint logs in for the first time. It is a port that the endpoint proxy uses to pose as the endpoint to the gateway.

  real_port
  : Indicates the endpoint port that the endpoint uses to listen for connections from the gateway proxy. It is the same port as the lfcd_port in the last.cfg file. Set this attribute using **wproxy db edit** if the lcfd_port and the real_port get out of sync.

  gwp_label
  : Indicates the label of the gateway proxy with which the endpoint connects.

  value Indicates the new value that you want to assign to the attribute.

db Indicates that you are performing actions on an endpoint proxy database.

–d *db_directory*
: Indicates the directory in which the database is located.

edit Edits an endpoint record in the endpoint proxy datebase.

ls Lists one or more endpoints in the endpoint proxy database.

odnum Indicates the number assigned to the endpoint by Tivoli Management Framework.

remove

Removes one or more endpoints from the endpoint proxy database.

–v Displays the version of the component that is installed on the machine from which it is run.

Authorization

Since the machine that the **wproxy** command runs on is probably not a machine with Tivoli Management Framework installed on it, authorization roles do not apply. On UNIX machines, run the **wproxy** command as the user account that was specified during installation of the endpoint proxy.

Examples

To list the endpoints in the endpoint proxy database that is stored in the folder /usr/epp, enter the following command:

```
wproxy db -d /usr/epp ls
```

To reassign endpoint with odnum 15763 to a new gateway proxy x3gateway, enter the following command:

```
wproxy db edit 15763 gwp_label=x3gateway
```

Appendix B. Troubleshooting

This appendix provides information about solving problems and gathering information to solve problems.

Testing Proxy Configuration

When the components are started, they try to exchange signals, called a *handshake*. The parent component sends its children a *who* request. The children reply. Similarly, the children send their parents a *tell* message and the parents reply. These exchanges enable the components in a chain of communication to establish the labels of all the components in the chain.

When one of the components is not running, the handshake fails. A message in the log file of each component lists the component with which the handshake failed.

Because components do not take the same amount of time to start, the log file might record a failure. The order of startup does not matter, but the order in which components are started affects the messages in the log file.

To test the components, set the log level to 3. The messages in the following example assume that you start the gateway proxy first. In the gateway proxy log file, look for lines of the type:

```
01/12/18 17:38:06 3 161 routingManager: WHO command received [l=null]
```

If the gateway proxy has problems connecting to its parent (relay or endpoint proxy), the log file records a message. For example, on Windows NT or Windows 2000, the entry is similar to the following:

```
01/12/18 17:43:07 1 130 ERROR multiplex.newsessopen: cannot open connection (-1)
```

To verify that the gateway proxy handshake reached its parent, check the log file on the parent machine for an entry similar to the following:

```
01/12/18 17:53:34 3 248 routingManager: WHO reply command received [l=ascotti_gwp1]
```

Do a similar check with the parent (endpoint proxy or relay). Start the parent component first. When the components communicate, the log file show entries for each child similar to the following:

```
01/12/18 17:40:16 3 179 routingManager: TELL reply command received
```

If you do not see an entry similar to this, the components are not communicating, and the log file shows a message similar to the following:

```
01/12/18 17:43:07 1 130 ERROR multiplex.newsessopen: cannot open connection (-1)
```

To verify that the endpoint proxy or relay handshake reached its child, check the log file on the child machine for an entry similar to the following:

```
01/12/18 17:51:27 3 47 routingManager: TELL command received [l=ascotti_gwp1]
```

Debug each component individually to ensure that each is operating correctly. Do the following for each component:

1. Stop the component.
2. Restart the component you are testing and check the log file.

Debugging Application Errors

If you have a problem accessing an endpoint or running a management operation, verify that all the links in your communication path from the Tivoli management region server (Tivoli server) to the endpoint work correctly. Use the following checklist to diagnose problems:

__ 1. Ensure that the Tivoli server is running. The Tivoli server and the Endpoint Manager must be available for any transaction with an endpoint.

__ 2. Ensure that the gateways that manage endpoints in the DMZ are running. Use the wgateway command to verify this or use the Endpoint Manager user interface on the Tivoli desktop to check the gateway status. Check the gateway log (`$DBDIR/gatelog`) on the gateway machine for messages indicating problems.

__ 3. Ensure that before starting your proxies, all machines that are involved in your deployment have DNS or IP visibility.

__ 4. Ensure that your firewall is correctly configured.

__ 5. Ensure that the endpoint proxy is configured to communicate with the correct gateway (address and port). When the endpoint proxy process starts, it logs a message stating the gateway IP and port with which it will communicate. See "Testing Proxy Configuration" on page 53 to ensure that the endpoint proxy is working correctly.

__ 6. Ensure that the gateway proxies and relays are configured to communicate with the correct endpoint proxy or relay. See "Testing Proxy Configuration" on page 53 to ensure proxy communication is working correctly.

__ 7. Ensure that the endpoints are running. Use the **wep** command to check the status of the endpoint. Check the lcfd.log file on the endpoint for warnings and errors.

Using the Log Files for Troubleshooting

When you install the component, a log file is created in the directory in which you installed it:

epp.log
Logs messages for the endpoint proxy.

eventsink.log
Logs messages for the event sink.

gwp.log
Logs messages for the gateway proxy.

relay.log
Logs messages for the relay.

You can adjust the level of detail that you want reported in the logs. See Chapter 3, "Configuring the Components" on page 27 for instructions about setting the log level for each component.

In a production environment, use the default proxy logging level of 3. The range is 0–11. Values higher than 3 lower performance significantly.

Providing More Detail in the Log Files

If you need to troubleshoot or to contact customer support, set the log levels of the components to a higher level of detail as follows:

- Set the gateway level to 7 by entering the following commands:

  `wgateway gateway_name set_debug_level 7`

  `wgateway gateway_name restart`
- Set the gateway proxy level to 8.
  - For UNIX:
    1. Edit the gwp.cfg file and change the debug-level entry to 8.
    2. Enter the command: **./gwproxy.sh stop**
    3. Enter the command: **./gwproxy.sh start**
  - For Windows:
    1. Stop the gateway proxy.
    2. Edit the gwproxy.cfg file and change the debug-level entry to 8.
    3. Start the gateway proxy.
- Set the endpoint proxy level to 8.
  - For UNIX:
    - Edit the epp.cfg file and change the debug-level entry to 8.
    - Enter this command: **./epproxy.sh stop**
    - Enter this command: **./epproxy.sh start**
  - For Windows:
    1. Stop the endpoint proxy.
    2. Edit the epproxy.cfg file and change the debug-level entry to 8.
    3. Start the endpoint proxy.
- Set the event sink level to 8.
  - For UNIX:
    - Edit the eventsink.cfg file and change the debug-level entry to 8.
    - Enter this command: **./eventsink.sh stop**
    - Enter this command: **./eventsink.sh start**
  - For Windows:
    1. Stop the event sink.
    2. Edit the eventsink.cfg file and change the debug-level entry to 8.
    3. Start the event sink.
- Set the relay level to 8.
  - For UNIX:
    - Edit the relay.cfg file and change the debug-level entry to 8.
    - Enter this command: **./relay.sh stop**
    - Enter this command: **./relay.sh start**
  - For Windows:
    1. Stop the relay.
    2. Edit the relay.cfg file and change the debug-level entry to 8.
    3. Start the relay.
- Set the endpoint level to 3

  Using the Web interface edit `Endpoint config` to change **log_threshold** to 3.

Alternatively, edit the `last.cfg` file on the endpoint machine and change **log_threshold** to 3. Stop and restart the endpoint.

Interpreting the Log Files

The log files present information in the following format:

```
01/11/22 16:03:27 1 2144 ERROR tcpunidir.createServerSocket: \
cannot bind socket (10049)
```

The following table explains each column in the log file message:

Column	Description
1	Date
2	Time
3	Debug level of the message is logged
4	Thread ID
5	Message description

The debug level determines which messages are logged. You should debug problems that are logged at levels 0-3. Do not try to analyze messages at levels 5-11, because they are intended for customer support personnel. The following list defines the severity of each debug level:

0 Fatal Error. During or after startup, an application cannot continue.

1 Error. A single operation has failed.

2 Warning

3 Informational

4 Verbose. A trace of all the information exchanged between the endpoint and the gateway.

5 Light Debug. Shows function entries and exits.

6 This severity is currently not in use.

7 Debug

8 Communication Library

9 Intensive Communication Library

10 All Communication Library

11 Intensive Debug

Providing Details to Customer Support

After you recreate the problem, provide the following information to Tivoli Customer Support:

- The error or exception message displayed and a description of the problem.
- The version of Tivoli Management Framework, applications, and patches installed. Use the **wlsinst** command.
- A description of the configuration of all the Tivoli components installed.
- Details about the firewall and its configuration.

- Log files that you have gathered, including the log file from the Endpoint Manager ($DBDIR/epmgrlog). See "Providing More Detail in the Log Files" on page 55.
- The lcfd.log with debug level 3
- The startup and configuration files of the components of the Tivoli Management Framework Firewall Security Toolbox:

  epproxy.sh, epproxy.cfg
  Script and configuration files for the endpoint proxy.

  eventsink.sh, eventsink.cfg
  Script and configuration files for the event sink.

  gwproxy.sh, gwproxy.cfg
  Script and configuration files for the gateway proxy.

  relay.sh, relay.cfg
  Script and configuration files for the relay.
- Optionally, if this indicates that there are errors: `odstat output`
- Optionally, if this indicates that there are errors: `wtrace output`

Tuning

When complex configurations, numerous endpoints, or long response times cause your distributions to time out, you can change some of the timeout values to try to fix the problem. The following sections describe some timeout values that you can adjust to optimize the connections in your Tivoli environment.

Timeout Values for Tivoli Management Framework

You can adjust the timeout values for Tivoli Management Framework to optimize the communication between the gateway and endpoints. For requests from the gateway to the endpoint, use the **wgateway** command to set the *session timeout*. This setting determines the amount of time (in seconds) that a gateway waits for a response from an endpoint after sending a request. The default is 300 seconds (5 minutes). Because responses from the endpoint might take longer when there are proxies between it and the gateway, a higher value would enable the gateway to wait longer for a response.

Timeout Values for the Firewall Security Toolbox

You can adjust timeout values to optimize the communication between the proxies. The **tcpip-timeout** value is the interval, in seconds, within which each component tries to complete a single operation with another component. The **tcpip-timeout** affects how long the endpoint proxy and gateway proxy wait to connect with their Tivoli Management Framework counterparts. For example, when the endpoint proxy connects to the gateway, it times out after the **tcp-timeout** interval is finished. Tune this parameter to give it time to connect to the gateway.

The **connect-timeout** value is the interval, in seconds, within which each component tries to connect to another component. Ensure that this value is not so low that the component does not have time to get a response from the other component. Tune this value to a value that is slightly longer than the longest a connection can take. For example, if it usually takes 10 seconds for the components to connect, set this value at 15 seconds. In unidirectional connections, the initiator sends the listeners a higher number of requests than in bidirectional connections, so the response time will be higher. However, if you make the value too high, the endpoint proxy takes longer to discover that the gateway proxy is down and to try

a backup gateway proxy. For example, an endpoint proxy, which is an initiator in a unidirectional connection, has a gateway proxy `a` and 2 backup gateway proxies `b` and `c` and the **connect-timeout** is 30 seconds. If `a` and `b` are down, it will take 60 seconds (30 plus 30) for the endpoint proxy to try `c`.

Connecting Components from Different Versions

If you try to use different versions of the Tivoli Management Framework Firewall Security Toolbox together, you get an error. Ensure that all the components in your configuration are from version 1.3.

Rescuing Lost Endpoints from the Gateway

Rescuing lost endpoints from the gateway Web page is not supported because the Web page does not go through the proxies.

Error on UNIX Systems When Installing as User Nobody

Problem: The following message is logged when you install a component on UNIX as user nobody and allocate a reserved port:

```
01/12/13 15:55:43 1 1 ERROR tcpbidir.createServerSocket: cannot bind socket (13)
01/12/13 15:55:43 0 1 FATAL tcpbidir.constructor: cannot create server socket
01/12/13 15:55:43 1 1 initRoutedSessionsManager: failure creating the connection
manager for child 0
01/12/13 15:55:43 0 1 routed sessions manager initialization failed
[cfg=epproxy.cfg;label=null]
```

Solution: Allocate port numbers that have permissions for the account being used to run the component.

NAT Not Supported

The network address translation (NAT) feature does not work with the Tivoli Management Framework Firewall Security Toolbox.

Wake on LAN Not Supported

The Wake on LAN® feature is not supported with the Tivoli Management Framework Firewall Security Toolbox.

Gateway Proxy Label Might Be Displayed Incorrectly

Problem: In the output of the **wproxy** command, the gateway proxy label is displayed incorrectly when the endpoint proxy and gateway proxy machines use different locales, in particular if the machines use a double-byte character set (DBCS).

Solution: Avoid using DBCS characters in the gateway proxy label. Although this does not cause functional problems, it can cause the label to be displayed incorrectly in command line output.

Multicast Feature Not Supported

The multicast feature of Tivoli Management Framework Version 4.1 and IBM Tivoli Configuration Manager Version 4.2 is not supported for the toolbox.

Port Conflicts

The port range 6000–8000 is usually reserved for communication from endpoints. It is recommended that you use ports outside of that range for communication between proxy components.

Gateway Times Out before Distribution Complete

Problem: When you have a complex configuration with multiple DMZs, your network can slow down significantly and some applications might take longer to distribute profiles.

Solution: To ensure that distribution takes place, increase the gateway timeout using the wgateway set_session_timeout command. See the *Reference Manual* for the command usage.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents.You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/10111400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information in softcopy form, the photographs and color illustrations might not appear.

Trademarks

IBM, Tivoli, Tivoli Enterprise, Tivoli Enterprise Console, Wake on LAN, and AIX are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

G

I

L

M

N

O

P

R

S

T

U

W

Printed in Denmark by IBM Danmark A/S

GC23-4826-00