Lucent Technologies
Bell Labs Innovations

Lucent Security Management Server

Release 9.1

Tools and Troubleshooting Guide

260-100-020R9.1
Issue 1
August 2006

Contents

About this information product

Purpose xv

Reason for reissue xv

How this document is organized xv

How to comment xvi

1 Introduction to the LSMS CLI

Overview 1-1

How to Access the LSMS Command Line 1-3

How the Command Line Works 1-5

2 LSMS CLI Commands

Overview 2-1

add applicationfilter 2-5

add brick 2-6

add brickruleset 2-25

add clienttunnel 2-26

add dependencymasks 2-27

add hostgroup 2-28

add lan2lantunnel 2-29

add servicegroup 2-30

apply brick 2-31

apply brickruleset 2-32
apply group 2-33
boot brick 2-34
bootgroup 2-35
delete applicationfilter 2-36
delete brick 2-37
delete brickruleset 2-38
delete clienttunnel 2-39
delete dependencymasks 2-40
delete hostgroup 2-41
delete lan2lantunnel 2-42
delete servicegroup 2-43
disable clienttunnel 2-44
disable lan2lantunnel 2-45
download brick 2-46
downloadgroup 2-48
enable clienttunnel 2-49
enable lan2lantunnel 2-50
failover brick 2-51
gotogrp 2-52
help 2-53
list applicationfilter 2-57
list brick 2-58
list brickruleset 2-60
list brickteps 2-62
list clientlicenselimits 2-63
list clienttunnel 2-66

list clienttunneldefaults 2-67
list current group 2-68
list dependencymasks 2-69
list groups 2-70
list hostgroup 2-71
list lan2lantunnel 2-72
list lan2lantunneldefaults 2-73
list servicegroup 2-74
list unusedclientteps 2-75
logout 2-76
lsmslogon 2-77
refreshmac brick 2-79
rehome brick 2-80
save brick 2-81
save brickruleset 2-82
save clientlicenselimits 2-83
save clienttunnel 2-84
save clienttunneldefaults 2-85
save dependencymasks 2-86
save hostgroup 2-87
save lan2lantunnel 2-88
save lan2lantunneldefaults 2-89
save servicegroup 2-90

3 **LSMS CLI Files**

Overview 3-1
brick File 3-3
brickruleset File 3-12

client license limits File 3-20

client tunnel defaults File 3-24

client tunnel File 3-34

hostgroups File 3-49

lan2lan tunnel defaults File 3-51

lan2lan tunnel File 3-57

servicegroups File 3-72

dependency masks File 3-74

4 **LSMS CLI Error Codes**

Overview 4-1

Codes 4-3

5 **Archive Recovery**

Overview 5-1

Creation of Archived Entities 5-2

Recovery of Archived Entities 5-3

6 **Database Utilities**

Overview 6-1

allowSecondarySetup 6-2

backup 6-3

changeInactiveBrickVersion 6-4

changeIP 6-5

changeName 6-6

convertToPrimary 6-7

dbsetup 6-8

restore 6-10

validateHash 6-11

7 **LSMS Service Status**

Overview 7-1

Displaying LSMS Service Status 7-2

Summary View 7-3

Individual Service View 7-5

8 **Troubleshooting Resources**

Overview 8-1

Online Help and Documentation 8-2

Log Files and Reports 8-3

Brick Problems 8-5

VPN Tunnel Problems 8-6

9 **Introduction to the Lucent VPN Firewall *Brick*® CLI**

Overview 9-1

Remote Console Connection via the Navigator 9-3

Remote Console Connection from the Command Line 9-5

Remote Console Connection from Windows 9-6

Remote Console Connection from Solaris 9-8

Set up a Local Connection 9-10

Command List Introduction 9-11

10 **Lucent VPN Firewall *Brick*® Device Display Commands**

Overview 10-1

display arptable 10-3

display clientpolicy 10-4

display configuration 10-5

display encapsulation 10-7

display failover 10-8

display files 10-9
display hostgroups 10-11
display icm 10-12
display interfacestatus 10-13
display lantolantunnels 10-16
display lsms 10-17
display mactable 10-18
display mempools 10-19
display nat 10-20
display partitions 10-21
display policy 10-22
display remoteconsole 10-24
display routes 10-25
display sa 10-27
display servicegroups 10-28
display sessions 10-29
display slamon 10-31
display time 10-32
display vlans 10-33
display zonetable 10-34

11 Lucent VPN Firewall *Brick*® Device Trace Commands

Overview 11-1
trace arp 11-3
trace audit delete 11-4
trace audit filter 11-5
trace audit list 11-7
trace audit modify 11-8

trace audit off 11-10
trace audit on 11-11
trace nonip 11-12
trace packet delete 11-13
trace packet filter 11-14
trace packet list 11-16
trace packet modify 11-17
trace packet off 11-19
trace packet on 11-20

12 Lucent VPN Firewall*Brick*® Device Set Commands

Overview 12-1
set baudrate 12-2
set errors 12-3
set printing 12-4
set screensize 12-5
set throttle 12-6

13 Lucent VPN Firewall *Brick*® Device General Commands

Overview 13-1
failover yield 13-2
help 13-3
login 13-4
logout 13-5
modem 13-6
reboot 13-7
refresh 13-8
repeat 13-10

A Set up a Remote Dial-In Connection

Overview A-1
Modem Setup on the Brick A-3
Modem Setup on a Remote Computer A-4
Create a Serial Port Access Password A-9
Dial Up and Log Into a Brick A-11

B Set up a Direct Serial Port Connection

Overview B-1
Set Up a Local Serial Port Connection B-2
Create a Serial Port Access Password B-5
Log In to a Brick B-7

Index

List of tables

3 LSMS CLI Files

3-1 e12e2esp EncryptionKey Field Values 3-67

3-2 e12e2esp AuthKey Field Values 3-68

3-3 e12e2ah AuthKey Field Values 3-68

3-4 e22e1esp EncryptionKey Field Values 3-68

3-5 e22e1esp AuthKey Field Values 3-68

3-6 e22e1ah AuthKey Field Values 3-69

List of figures

7 LSMS Service Status

7-1 LSMS Status — Total 7-3

7-2 LSMS Status — Single 7-5

9 Introduction to the Lucent VPN Firewall *Brick*® CLI

9-1 Brick Console Window 9-3

9-2 Windows Brick Remote Console Session 9-7

9-3 Local Connection 9-10

A Set up a Remote Dial-In Connection

A-1 Remote Dial-in Connection A-2

A-2 Entering Name for Hyperterminal Connection A-4

A-3 Connect To Window in Hyper Terminal A-5

A-4 Connect Window in HyperTerminal A-6

A-5 Configuring Port Settings for Remote Computer Modem A-7

A-6 Setting Emulation Type for Remote Computer Modem A-8

B Set up a Direct Serial Port Connection

B-1 Entering Name for Hyperterminal Connection B-2

B-2 Configuring Port Settings for Remote Computer Modem B-3

About this information product

Purpose

This is the *Tools and Troubleshooting Guide* for the Lucent Security Management Server (LSMS) and the Lucent VPN Firewall *Brick*® device. While the heart of this manual explains how to use the command line interface for the LSMS and the Brick, we have expanded it to provide the reader more information on the other tools included with the product as well as a basic introduction to troubleshooting the system.

Reason for reissue

Updated with information for Release 9.1.

How this document is organized

There are three sections to this guide:

- LSMS Command Line Interface (CLI)
- LSMS Tools and Troubleshooting Resources
- Brick CLI

The LSMS command line interface provides administrators with an alternative to the LSMS graphical user interface (GUI). It allows administrators to perform many - but not all - of the tasks they ordinarily perform from the GUI by executing typed commands and scripts instead.

In the tools and troubleshooting section, we explore recovery of individual items in the database, database utilities as well as a summary of the troubleshooting tools available within the product.

The Brick CLI provides administrators with a means of directly querying the Brick for diagnostic and troubleshooting purposes. This interface is especially useful when communication between the Brick and the LSMS is severed. There are a variety of methods available to the administrator to access the Brick console, both directly and remotely.

How to comment

To comment on this information product, go to the Online Comment Form (http://www.lucent-info.com/comments/enus/) or email your comments to the Comments Hotline (comments@lucent.com).

1 Introduction to the LSMS CLI

Overview

Purpose

This chapter provides an introduction to the Lucent Security Management Server (LSMS) Command Line Interface (CLI).

Purpose of the CLI

The CLI is an alternative to the LSMS graphical user interface (GUI). It enables administrators to log into the LSMS and perform administrative tasks using typed commands instead of the graphical components of the GUI.

If you decide to use the Command Line Interface, you will not be able to perform all the tasks that an administrator has to perform. However, administrators will be able to:

- Make changes to existing Lucent VPN Firewall *Brick*® device Policy Assignments
- Review their group security policy and create, edit, and delete all of the following:
  - Brick zone rulesets
  - Host groups
  - Service groups
  - Dependency masks
  - Application filters
- Run commands to create, edit, and delete a Brick, as well as commands to control a Brick, such as apply a policy, reboot, download software, rehome, failover, and refreshmac.
- Make VPN changes to create, edit, delete, enable, and disable Client and LAN-to-LAN tunnels, and saving tunnel information (output of CLI commands) into files for subsequent reuse.

  Important! This manual explains how to use the command line interface to perform specific administrative tasks. It does not describe these tasks in detail or explain when or why they need to be performed.

If the significance of a task is not clear, you need to consult the appropriate manual for a more comprehensive discussion of the task.

Supported Brick devices

The following available Brick models are supported by the current LSMS release:

- Model 20 Brick device
- Model 50 Brick device
- Model 150 Brick device
- Model 350 Brick device
- Model 1100/1100A Brick device
- Model 700 Brick device
- Model 1200 Standard and HS Brick devices

Some of the above Brick device models require a specific patch of the current LSMS release in order to be fully supported. For details about the LSMS patch release required for a specific Brick device model, refer to the *User's Guide* for the Brick device model or check out the **VPN Firewall Portfolio** link on the following website for specific LSMS patch release information: http://www.lucent.com/security/

Contents

How to Access the LSMS Command Line	1-3
How the Command Line Works	1-5

How to Access the LSMS Command Line

Overview

The LSMS command line interface can be accessed directly from the machine running the LSMS application, or by logging into the machine remotely using a utility such as Telnet or SSH. The appropriate service (telnet or ssh) needs to be started on the LSMS host.

> **Important!** Telnet service is not as secure, as all of the information is transmitted in clear text. The use of a more secure communication method like SSH is highly recommended.

Before you Begin

Before accessing the command line interface, you should make certain that the appropriate paths and permissions have been set so that the commands execute properly from any directory in your system.

The procedure for doing this differs on machines running the Windows and Solaris operating systems.

Windows

On machines running the Windows 2000 Professional, 2000 Server, XP Professional, or Server 2003 operating systems, you have to add the LSMS root directory to your path. To do this, execute the following command:

`SET PATH=%PATH%;<LSMSRootDir>`

where `<LSMSRootDir>` is the full pathname of your LSMS root directory. For example, if your LSMS root directory is *c:\users\isms\lmf*, you would execute this command:

SET PATH=%PATH%;c:\users\isms\lmf

If you do not set the Windows path, you will have to change directory manually before issuing the `lsmslogon` command.

Solaris

On a machine running the Solaris operating system, the root administrator has to make sure the files generated by the command line interface (see Chapter 3, “LSMS CLI Files”) are placed in a directory in which all command line interface users have read and execute privileges.

In addition, all command line interface users should be instructed to add this directory to their paths by executing the following command:

`PATH=$PATH:<directory_name>`

where `<directory_name>` is the name of the directory containing the generated files.

Important! When a user specifies a directory as an argument to the `lsmslogon` command, that directory is created relative to the current directory. Remember to supply the complete path to insure that you are accessing the directory you really want, and avoid the proliferation of similarly named directories throughout the directory tree.

Access the Command Line Interface

You can access the command line interface from the host running the LSMS and from a remote machine.

LSMS Host

If you are sitting at the host running the LSMS, you can access the command line interface by:

- Opening a Command Prompt window (Windows) or X-term window (Solaris), and
- Executing the `lsmslogon` command (see Chapter 2, "LSMS CLI Commands").

Remote

If you are logged into the machine remotely using Telnet or SSH, you can execute the `lsmslogon` command from the Telnet or SSH window.

A Telnet or SSH utility is provided with both Windows and Solaris. To run Telnet or SSH under Windows, select **Run** from the Windows Start menu. In the **Open:** field of the Run pop-up window enter `telnet` or `ssh`, followed by the path to the appropriate machine and directory.

Important! Windows 2000 provides an inbound Telnet or SSH capability, while Windows NT does not. To Telnet or SSH into a Windows machine running the LSMS software you must install a third-party Telnet or SSH server.

For security purposes, it is recommended that a secure remote access terminal program such as SSH, rather than telnet, be used to provide access for command line interface functions.

☐

How the Command Line Works

Overview

The LSMS command line interface allows Administrators to perform many of the tasks they ordinarily perform using the graphical user interface on the LSMS.

These tasks can be performed by executing individual commands from the command line interface, or by embedding these commands in scripts and executing the scripts.

The advantage of a script is that it allows you to execute multiple commands in a single action and can be used for automating complex functions in LSMS.

Commands

The command line interface consists of the `lsmslogon` command and several other LSMS commands.

To execute all of these commands *except* `lsmslogon`, you must precede the command (which can be one or two words) with the character string:

```
lsmscmd
```

as in:

```
lsmscmd gotogrp
```

or:

```
lsmscmd list brickruleset
```

Once an LSMS command is executed, the results of the command are reported and you are returned to the control of the operating system. This means you are free to issue another LSMS command, or another operating system command.

You can execute a script, or open and use a text editor, and then return to the command line interface and execute another LSMS command. You can continue in this mode until you log out and terminate the session.

All LSMS commands from the same session must be executed from the same window.

Types of Commands

The commands can be grouped into the following categories.

Access Commands

The access commands are `lsmslogon` and `lsmscmd logout`. To begin a command line interface session, enter `lsmslogon`, followed by the Admin ID, destination directory for generated files, and the password.

To end the session, execute the `lsmscmd logout` command. See the detailed description of the command under “logout” (p. 2-76) in Chapter 2, “LSMS CLI Commands”

Help Command

The `lsmscmd help` command displays a list of the other commands and provides a description of the syntax of each command.

Administrator Commands

These commands can be grouped according to the LSMS entity they effect: Brick, Brick ruleset, group, host group, service group, or dependency mask. The commands are briefly described in the following table:

Entity	Command	Description
Application Filter Commands	add applicationfilter	Adds a new application filter which then may be included as part of a service group.
	delete applicationfilter	Deletes the application filter specified in the command.
	list applicationfilter	Lists the contents of the application filter specified in the command.
	save applicationfilter	Saves the application filter specified in the command.

Entity	Command	Description
Brick Commands	add brick	Adds a new Brick to the LSMS database.
	apply brick	Applies any changed rulesets to their assigned ports on the specified Brick in your current group.
	boot brick	Allows an administrator to reboot a Brick from the command line
	bootgroup	Reboots all Bricks in the current group.
	delete brick	Deletes a Brick from the LSMS database.
	download brick	Updates Brick software after an LSMS software upgrade.
	failover brick	Initiates failover of a failover Brick pair.
	list brick	Writes the contents of a specified Brick to a file.
	rehome brick	In a redundant LSMS configuration, this command reassigns management of a specific Brick to the other LSMS of the pair.
	save brick	Saves the contents of the Brick file back into the LSMS database.

Entity	**Command**	**Description**
Brick Ruleset Commands	add brickruleset	Adds a Brick ruleset to the current group.
	apply brickruleset	Compiles and applies the security policy to any Brick ports that have older versions of the specified Brick ruleset in your current group.
	delete brickruleset	Deletes the specified Brick ruleset from the current group.
	list brickruleset	Retrieves the specified Brick ruleset and stores the information in a file with the name of the Brick ruleset on the machine running the LSMS.
	save brickruleset	Saves the specified Brick ruleset in the current group.
Dependency Mask Commands	add dependency masks	Adds a dependency mask with the specified name to the current group.
	delete dependency masks	Deletes the specified dependency mask from the current group.
	list dependency masks	Retrieves the specified dependency mask and stores the information in a file with the name of the dependency mask on the machine running the LSMS.
	save dependency masks	Saves the specified dependency mask in the current group.

Entity	Command	Description
Host Group Commands	add hostgroup	Adds a host group with the specified name to the current group.
	delete hostgroup	Deletes the specified host group from the current group.
	list hostgroup	Retrieves the specified host group and stores the information in a file with the name of the host group on the machine running the LSMS.
	save hostgroup	Saves the specified host group in the current group.
Service Group Commands	add servicegroup	Adds a service group with the specified name to the current group.
	delete service group	Deletes the specified service group from the current group.
	list servicegroup	Retrieves the specified service group and stores the information in a file with the name of the service group on the machine running the LSMS.
	save servicegroup	Saves the specified service group in the current group.
Group Commands	apply group	Applies all changed policies, devices, and rulesets for a given group.
	gotogrp	Changes the group the administrator is working in to the specified group.
	list current group	Echoes back the group that the administrator is currently working in.

Entity	Command	Description
VPN Commands	list brickteps	Lists all the tunnel endpoints of a given Brick to the console.
	list clienttunneldefaults	Retrieves the default Client tunnel configuration to a file.
	save clienttunneldefaults	Saves the default Client tunnel configuration back to the LSMS.
	list clienttunnel	Retrieves the specified Client tunnel configuration to a file.
	list unusedclientteps	Lists all the tunnel endpoints of a given Brick that are not used in any existing Client tunnel. The results are displayed on the console.
	save clienttunnel	Save the specified Client tunnel configuration back to the LSMS.
	add clienttunnel	Creates a new Client tunnel with the given name to the LSMS.
	delete clienttunnel	Deletes the specified Client tunnel from the LSMS.
	enable clienttunnel	Enables the specified Client tunnel in the LSMS.
	disable clienttunnel	Disables the specified Client tunnel in the LSMS.
	list lan2lantunneldefaults	Retrieves the default LAN-to-LAN tunnel configuration to a file.
	save lan2lantunneldefaults	Saves the default LAN-to-LAN tunnel configuration back to the LSMS.
	list lan2lantunnel	Retrieves the specified LAN-to-LAN tunnel configuration to a file.
	save lan2lantunnel	Saves the specified LAN-to-LAN tunnel configuration back to the LSMS.
	add lan2lantunnel	Creates a new LAN-to-LAN tunnel with the given name to the LSMS.
	delete lan2lantunnel	Deletes the specifiied LAN-to-LAN tunnel from the LSMS.
	enable lan2lantunnel	Enables the spedified LAN-to-LAN tunnel from the LSMS.

Entity	Command	Description
disable lan2lantunnel	Disables the specified LAN-to-LAN tunnel from the LSMS.	

Files

The `list` commands retrieve information from the LSMS and save it in one or more files on the machine running the LSMS. These files are put in the directory that you specify, either when you log in or when you execute the command.

The files that are produced are ASCII files that can be edited using any standard text editor, such as Notepad on Windows or vi on Solaris. To make changes to a file:

- Edit the appropriate file using a text editor.
- Save the file in the LSMS (using the`save` commands).
- Load or apply the file to the brick (using the `apply brick` or`apply brick ruleset` commands).

Examples

There are many kinds of tasks you can perform using the command line interface. The following are just a few examples:

- You can execute a `list hostgroup` command, and then edit the resulting host group file.
- You can execute a `list brick` command, and then edit the resulting brick file to add one or more new port assignments.
- You can write a script containing both LSMS and non-LSMS commands.
  This script could include:
  - An `lsmslogon` command to begin the session,
  - List commands to create the appropriate files,
  - Non-LSMS commands to edit the files,
  - Apply commands to apply the results to a Brick, and
  - A `logout` command to terminate the session.

□

2 LSMS CLI Commands

Overview

Purpose

This chapter describes the commands that can be executed from the LSMS command line interface.

The commands are listed in alphabetical order. For each command, the chapter provides an overview, a description and explanation of the format, and examples.

All the commands in this chapter — except the `lsmslogon` command — must be preceded by the keyword `lsmscmd` and a space.

Objectives

This chapter provides information to do the following:

1. Execute the `lsmslogon` and `logout` commands to begin and end a command line session.
2. Execute the `help` command to display a description of the syntax of each command.
3. Execute the `add hostgroup`, `delete hostgroup`, `list hostgroup`, and `save hostgroup` commands to add, delete, and save host groups, as well as list all available host groups.
4. Execute the `add servicegroup`, `delete servicegroup`, `list servicegroup`, and `save servicegroup` commands to add, delete, and save service groups, as well as list all available service groups.
5. Execute the `add dependencymasks`, `delete dependencymasks`, `list dependencymasks`, and `save dependencymasks` commands to add, delete, and save dependency masks, as well as list all available dependency masks.
6. Execute the `add brickruleset`, `delete brickruleset`, `list brickruleset`, and `save brickruleset` commands to add, delete, and save brick zone rulesets, as well as list all available brick zone rulesets.

7. Execute the `add brick`, `delete brick`, `list brick`, and save brick commands to add, delete, and save Lucent VPN Firewall*Brick*® devices. Execute the `apply brick` command to apply policy information to a specified Brick. Execute the `apply brickruleset` command to apply a ruleset to all ports with that ruleset assigned.
8. Execute the `goto group` command to administer a different group, and execute the `list group` command to see the current group that is being administered.
9. Add, delete, list, and save application filters.
10. List, save, delete, enable, and disable Client and LAN-to-LAN tunnels and list tunnel endpoint information.

Contents

add applicationfilter	2-5
add brick	2-6
add brickruleset	2-25
add clienttunnel	2-26
add dependencymasks	2-27
add hostgroup	2-28
add lan2lantunnel	2-29
add servicegroup	2-30
apply brick	2-31
apply brickruleset	2-32
apply group	2-33
boot brick	2-34
bootgroup	2-35
delete applicationfilter	2-36
delete brick	2-37
delete brickruleset	2-38
delete clienttunnel	2-39
delete dependencymasks	2-40
delete hostgroup	2-41
delete lan2lantunnel	2-42
delete servicegroup	2-43
disable clienttunnel	2-44

disable lan2lantunnel	2-45
download brick	2-46
downloadgroup	2-48
enable clienttunnel	2-49
enable lan2lantunnel	2-50
failover brick	2-51
gotogrp	2-52
help	2-53
list applicationfilter	2-57
list brick	2-58
list brickruleset	2-60
list brickteps	2-62
list clientlicenselimits	2-63
list clienttunnel	2-66
list clienttunneldefaults	2-67
list current group	2-68
list dependencymasks	2-69
list groups	2-70
list hostgroup	2-71
list lan2lantunnel	2-72
list lan2lantunneldefaults	2-73
list servicegroup	2-74
list unusedclientteps	2-75
logout	2-76
lsmslogon	2-77
refreshmac brick	2-79
rehome brick	2-80
save brick	2-81
save brickruleset	2-82
save clientlicenselimits	2-83
save clienttunnel	2-84
save clienttunneldefaults	2-85

save dependencymasks	2-86
save hostgroup	2-87
save lan2lantunnel	2-88
save lan2lantunneldefaults	2-89
save servicegroup	2-90

add applicationfilter

Overview

The `add applicationfilter` command adds a new application filter which then may be included as part of a service group.

Format

The format of the `add applicationfilter` command is:

```
lsmscmd add applicationfilter <application filter name> [folder name]
```

where:

- *<application filter name>* is the name of the new application filter. This arguement is required.
- `[folder name]` is the name of the folder into which you would like the brick ruleset added. This argument is optional.

Explanation

An application filter allows additional application layer validation, inspection and access control on the Brick.

Example

```
lsms add applicationfilter H323
```

This command creates an application filter called H323.

□

add brick

Overview

The `add brick` command creates a new Brick, based on user-specified configuration information provided in a file, in the LSMS database.

Format

The format of the `add brick` command is:

`lsmscmd add brick <filename> [folderName]`

where:

- *<filename>* is the name of the configuration file for the Brick being created.
- `[folderName]` is the name of the folder into which you would like the Brick added. This argument is optional.

Explanation

Use the `add brick` command to add a new Brick instance to the LSMS database.

Executing this command is equivalent to right-clicking on the Bricks folder, selecting **New Brick** from the pop-up menu, completing the GUI fields, and selecting **Save**.

All of the configuration information for a Brick is specified in a file. The name of the file is *<filename>* and is located on the LSMS in the directory *<cli dir>/<group>/Devices/Brick*, where *<cli dir>* is given as an argument to the lsmslogon command, and *<group>* is the current group that you are in when executing the `add brick` command. By default, the group is **system**.

Data in the Brick configuration file is organized in a `<name>=<value>` format. To specify table data, an index is added to the name to specify the row in the table to which it applies.

Only one name/value pair exists per line. The order of the lines do not affect the execution of the command.

The following table describes each name/value pair that is defined in the file for a new Brick:

Field	Explanation
name= <brick name>	The name of the Brick that appears in the **Brick Name** field on the Brick tab. You cannot modify the name of a Brick.

<table>
<tr><th>Field</th><th>Explanation</th></tr>
<tr><td>firewallIP= <IP address|”” ></td><td>The Brick IP address that in the Brick IP Address field on the Brick tab. It may be blank if the Addressing Method is not a static address. The IP address of a Brick cannot be modified.</td></tr>
<tr><td>mobile=true|false</td><td>This corresponds to the checkbox labeled Dynamically Learn Address that appears on the Brick tab. The value true is equal to a checked checkbox.</td></tr>
<tr><td>gateway= <IP Address|dhcp|pppoe1|pppoe2></td><td>The gateway IP address that appears in the LSMS GUI in the Gateway IP Address field on the Brick tab. This field may be blank. If the radio button for the Brick/Gateway IP Addressing Method is set for DHCP, then this field is dhcp. If the radio button is set for PPPOE #1, then this field is pppoe1. If the radio button is set for PPPOE #2, then this field is pppoe2.</td></tr>
<tr><td>description= <one line description></td><td>The description that appears in the Descriptionfield on the Brick tab. This field is optional.</td></tr>
<tr><td>showVLANView=true|false</td><td>This corresponds to the checkbox labeled Always Show VLAN Information on the Brick tab. The value true is equal to a checked checkbox. This field is used to indicate if VLAN-specific information should be displayed on the LSMS GUI or exposed in the CLI file. Once this field is set to true, it cannot be changed back to false. If set to false, the following fields are not displayed by the <code>list brick</code>command (inputting the fields manually has no effect):VLANDomain, VLANMembership, receiveFormat, transmitFormat, partitionCount, partitionName, partitionVLANID, zoneVLANID, sourcePartition, nextHopPartition, allowBridgedVLANs</td></tr>
<tr><td>timeOffsetFromSms= <number></td><td>This is the time offset that appears in the LSMS Relative Time Offset field on the Brick tab. The value range is from -24 to 24 and can accept decimal values (example: 3.5).</td></tr>
</table>

Field	Explanation
brickType= `<brick type>`	This is the Brick type that appears in the **Brick Type** field on the Brick tab. The `<brick type>` value is as follows for each Model Brick: *microbrick* for the Model 20 or 50 *brick* for the Model 80, 150, or 201 *brick300* for the Model 300 *brick350* for the Model 350 *brick500* for the Model 500 *brick1000A* for the Model 1000 (9/2/0) *brick1000B* for the Model 1000 (7/2/0) *brick1000C* for the Model 1000 (5/4/0) *brick1100A* for the Model 1100 (7/0/13) *brick1100B* for the Model 1100 (7/4/1) *brick1100C* for the Model 1100 (7/6/1)
stickiness= `<number in secs>`	This field corresponds to the **Rehome Delay field** on the Brick tab. If the checkbox labeled **Rehome If Higher Priority LSMS or LSCS Is Available** is unchecked, then this field should have the value **0**.
adminServerIP= `<IP address,...>` *adminServerName=* `<name,...>` *adminServerType=* `<type,...>` *adminServerAssocLSMS=* `<assocLSMS,...>`	This is a comma-separated list of the LSMSs (Primary, Secondary, Compute Server) and a corresponding list of their server names, types, and associated LSMSs. This information appears in the **Home LSMS/LSCS Priority** table on the Brick tab. The order of this field list is the order in which they appear in the GUI table. Only **adminServer** can be modified, so **adminServerNames, adminServerTypes**, and **adminServerAssocLSMSs values** are not necessary for the `add` or `save` commands. The IP addresses must be either in the range of the Gateway or the range of one of the physical port addresses.

Field	Explanation
brickInterfaceCount	In the following fields that refer to **brickInterfaceCount**, the value of this field is derived from brickType as follows: if **brickType** is **microbrick, brickInterfaceCount=4** if **brickType is brick, brickInterfaceCount=5** if **brickType** is **brick300, brickInterfaceCount**=**9** if **brickType** is **brick350, brickInterfaceCount**=*9* if **brickType** is **brick500, brickInterfaceCount**=**16** if **brickType** is **brick1000A, brickInterfaceCount**=**8** if **brickType** is **brick1000B, brickInterfaceCount**=*10* if **brickType** is **brick1000C, brickInterfaceCount**=*12* if **brickType** is **brick1000D, brickInterfaceCount**=*21* if **brickType** is **brick1100B, brickInterfaceCount**=*13* if **brickType** is **brick1100C, brickInterfaceCount**=*15*
interfaceName[i]= <port name>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the **Port** field on the Physical Ports tab of the Brick Editor.
aggregatePort[i]= <port name>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the **Aggegate With** field on the Physical Ports tab of the Brick Editor. This field is optional.
portDescription [i]= <one line port description>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the Description field on the Physical Ports tab of the Brick Editor.
VLANDomain[i]= <vlan domain>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the VLAND Domain field on the Physical Ports tab of the Brick Editor. **showVLANView** must be checked to see this field on the LSMS GUI.
defaultVLANID[i]= <vlan id or range>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the **Default VLAN ID** field on the Physical Ports tab of the Brick Editor. **showVLANView** must be checked to see this field on the LSMS GUI. In the CLI file, this field can be cross-referenced with the VLANID field to determine the IP Address/Mask (**VLANipAddress**) assigned to each Ethernet port (**interfaceName**).

Field	Explanation
VLANMembership[i]= <vlan id or range>	**i**is in the range of **0..brickInterfaceCount-1**. This field corresponds to the **VLAN Membership** field on the Physical Ports tab of the Brick Editor. **showVLANView**must be checked to see this field on the LSMS GUI.
receiveFormat [i]= <Untagged | 802.1Q | Any>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the **Receive Format** field on the Physical Ports tab of the Brick Editor. **showVLANView** must be checked to see this field on the LSMS GUI.
transmitFormat[i]= <Untagged | 802.1Q | Any>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the **Transmit Format** field on the Physical Ports tab of the Brick Editor. **showVLANView** must be checked to see this field on the LSMS GUI.
dhcpRequest[i]=true | false	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the checkbox labeled **Send/Receive DHCP request on this port**on the Physical Ports tab of the Brick Editor. **true** means that the checkbox is checked.
enableQOS[i]=true | false	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the checkbox labeled Enable Port Bandwidth Parameters on the Physical Ports tab of the Brick Editor. **true** means that the checkbox is checked.
transmitBitRate[i]= <number>B|K|M|G	**i** is in the range of **0..brickInterfaceCount-1**. **B**=Bits/sec **K**=Kilobits/sec **M**=Megabits/sec **G**=Gigabits/sec This field corresponds to the **Transmit Bandwidth** field on the Physical Ports tab of the Brick Editor.

Field	Explanation
receiveBitRAte[i]= <num ber>*B|K|M|G*	**i** is in the range of **0..brickInterfaceCount-1**. **B**=Bits/sec **K**=Kilobits/sec **M**=Megabits/sec **G**=Gigabits/sec This field corresponds to the **Receive Bandwidth** field on the Physical Ports tab of the Brick Editor.
interfaceMode[i]= <auto |100BASE-TXFD | 100BASE-TX | 10BASE-TFD | 10BASE-T | 1000BASE-FXFDFC | 1000BASE-FXFDnoF>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the Mode field on the Physical Ports tab of the Brick Editor. The values are as follows: **auto**=Auto **100BASE-TXFD**=100Mb Full Duplex **100BASE-TX**=100Mb Half Duplex **10BASE-TFD**=10Mb Full Duplex **10BASE-T**=10Mb Half Duplex **1000BASE-FXFDFC**=1000Mb Full Duplex - Flow Control On **1000BASE-FXFDnoFC**=1000Mb Full Duplex - Flow Control Off The last two choices are for Gigabit ports only.
enableJumboFrame[i]=true|false	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the field labeled **Enable Jumbo Frames**on the Physical Ports tab of the Brick Editor. true means Yes. This field is only visibile for Gigabit ports on the LSMS GUI.
mtu[i]= <512-1500>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the **MTU** field on the Physical Ports tab of the Brick Editor. This field may be blank.
ignoreHeartBeatFailures[i]= true|false	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the checkbox labeled **Ignore heartbeat failures on this link** on the Physical Ports tab of the Brick Editor. **true** means that the checkbox is checked.

Field	Explanation
brickVLANIPCount= <number of VLAN/IP assignments>	This count is the number of rows on the VLAN/IP Assignment tab. This tab is only displayed if **showVLANView** is checked.
VLANID[i]= <vlan id>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the the **VLAN ID** field on the VLAN/IP Assignment tab of the Brick Editor. In the CLI file, this field can be cross-referenced with the **defaultVLANID** field to determine the IP Address/Mask (**VLANipAddress**) assigned to each Ethernet port (**interfaceName**).
VLANipAddress[i]= <IP address mask> | dhcp | pppoe1 | pppoe2	**i** is in the range of **0..brickInterfaceCount-1**. If **showVLANView** is **true**, this field corresponds to the **IP Address/Mask** field on the VLAN/IP Assignment tab of the Brick Editor; otherwise, it corresponds to the **IP Address/Mask** field on the Physical Ports tab of the Brick Editor.
allowBridgedVLANs=true|false	This field corresponds to the checkbox labeled **Bridge VLANs** in same partition with same IP Address/Mask at the bottom of the VLAN/IP Assignment tab of the Brick Editor. **true**means that the checkbox is checked.
partitionCount= <number of partitions>	This count is the number of rows on the Partitions tab of the Brick Editor. This tab is only displayed if **showVLANView** is checked.
partitionNam [i]= <partition>	*i* is in the range of *0..brickInterfaceCount*-1. This field corresponds to the Partition field on the Partitions tab of the Brick Editor.
partitionVLANID[i]= [local,] <vlan id | comma separated list | range>	**i** is in the range of **0..brickInterfaceCount-1**. This field corresponds to the **VLAN IDs** field on the Partitions tab of the Brick Editor. Adding **local** to the list of vlan ids is equivalent to checking the **Local Partition for LSMS Communication** checkbox in the Brick VLAN Partition Editor.
localPartition= <partition>	This field corresponds to the partition name that is assigned as the **Local Partition for LSMS Communication**. If no partition in the list has this checkbox explicitly checked, then the **localPartition** is *Default. Only one partition can be the local partition.

Field	Explanation
zoneInterfaceCount= <number of policy assignments>	This count is the number of rows on the Policy Assignment tab of the Brick Editor. There can be more than one entry per port. The first row, the one with the firewall zone on the local port and an index of *0*, cannot be modified.
interfaceNumber[i]= <index of interfaceName>	**i** is in the range of **0..brickInterfaceCount-1**. The value of this field indirectly determines the *Port* field on the Policy Assignment tab (Basic tab). It is the index (row) into the Brick Interface table that is used here in the Zone interface. The Policy Assignment tab Port=**interfaceName [interfaceNumber [i]].**
policy[i]= <zone ruleset>	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the Zone Ruleset field on the Policy Assignment tab of the Brick Editor (Basic tab).
virtualBrickAddress[i]= <IP address | dhcp | pppoe1 | pppoe2	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the **Tunnel Endpoint/Virtual Brick Address** field on the Policy Assignment tab (Basic tab). Note: the TEP/VBA Addressing Method radio buttons are encoded as values in this field.
matchVBAPackets[i]=true|false	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the checkbox labeled **Match Packets to or from this VBA** on the Policy Assignment tab ->Basic tab. **true** means that the checkbox is checked.
*zoneIPHost[i]= <host group> zoneIPAddressOrRange [i]= <IP address | range| subnet mask | *>*	**i** is in the range of **0..zoneInterfaceCount-1**. These fields correspond to the **Hosts Behind Tunnel/Zone IP Adddresses** field on the Policy Assignment tab—> Basic tab. If one is filled in, the other is blank.
localPresenceHost[i]= <host group> localPresenceOrRange [i]= <IP adddress | range | subnet mask>	**i** is in the range of **0..zoneInterfaceCount-1**. These fields correspond to the **Local Map Adddresses** field on the Policy Assignment tab —>Basic tab. If one is filled in, the other is blank.

Field	Explanation
vpnCertificate[i]= <certificate>	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the **VPN Certificate**field on the Policy Assignment tab —> Basic tab. It may be blank.
defaultAuthService[i]= <authentication service>	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the **Authentication Service**field on the Policy Assignment tab —> Basic tab. It may be blank.
AuthTimeOut [i]= <number of minutes>	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the **Authentication Timeout** field on the Policy Assignment tab —> Basic tab.
*SourceIPs[i]= <IP address | comma separated list | range | *>*	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the **Allowed Source IP Addresses** *field on the Policy Assignment tab —> Basic tab.*
localAddressmapping[i]=direct	**i** is in the range of **0..zoneInterfaceCount-1**. This field is not shown in LSMS GUI. It is hardcoded to **direct**.
zonePriority[i]=<0..31>	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the **Zone Priority** field on the Policy Assignment tab—> Bandwidth tab.
maxQueueLatency[i]= <number in milliseconds>	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the **Maximum Queue Latency**field on the Policy Assignment tab —> Bandwith tab.
guarZoneRateIn[i]= <number> B|K|M|G	**i** is in the range of **0..zoneInterfaceCount-1**. **B**=Bits/sec **K**=Kilobits/sec **M**=Megabits/sec **G**=Gigabits/sec A bare number represents bits/sec. This field corresponds to the **Guarantees Into Zone** field on the Policy Assignment tab —> Bandwidth tab.

Field	Explanation
maxZoneRateIn[i]= <number> B|K|M|G	**i** is in the range of **0..zoneInterfaceCount-1**. **B**=Bits/sec **K**=Kilobits/sec **M**=Megabits/sec **G**=Gigabits/sec A bare number represents bits/sec. This field corresponds to the **Limits Into Zone** field on the Policy Assignment tab —> Bandwidth tab.
maxZoneRateOut[i]= <number> B|K|M|G	**i** is in the range of **0..brickInterfaceCount-1**. **B**=Bits/sec **K**=Kilobits/sec **M**=Megabits/sec **G**=Gigabits/sec A bare number represents bits/sec. This field corresponds to the **Limits Out of Zone** field on the Policy Assignment tab —> Bandwidth tab.
maxZoneConcSession[i]= <number>	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the **Limits Into Zone...Simultaneous Sessions** field on the Policy Assignment tab —> Bandwidth tab of the Brick Editor.
maxZoneConcSessTotal[i]= <number>	**i** is in the range of **0..zoneInterfaceCount-1**. This field corrsponds to the **Limits Entire Zone...Simultaneous Sessions** field on the Policy Assignment tab —> Bandwidth tab of the Brick Editor.

Field	Explanation
qosParamsActive[i]= <2 digit hex number>	**i** is in the range of **0..zoneInterfaceCount-1**. This field is an ASCII hex representation of a byte that encodes the enabled/disabled setting of the seven QOS fields combined. **guarZoneRateIn —>** bit 1 **guarZoneRateOut —>** bit2 **maxZoneRateIn —>**bit 3 **maxZoneRateOut —>** bit 4 **maxZoneConSessTotal —>**bit 5 **maxZoneConcSessIn —>**bit 6 Example: **qosParamsActive [2] = [17]** In the above example, **guarZoneRateIn, guarZoneRateOut, maxZoneRateIn, maxZoneConcSessTotal** are enabled, while **maxZoneRateOut, maxzoneConcSess** and **maxZoneConcSessOut** are disabled.
setTOSDiffServBits[i]=true|false	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the **Set TOS/DiffServ Bits** checkbox on the Policy Assignment tab —> Bandwidth tab. **true** means that the checkbox is checked.
separateBorrowSetting[i]= true|false	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to the **Bit Template** radio buttons on the Policy Assignment tab —> Bandwidth tab.
bitPatternNonBorrow[i]= <2 digit hex number>	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to any of the **Raw Bit, Hex**, and **Verbal view**fields on the Policy Assignment tab —> Bandwidth tab. This is the set of fields on the left hand side when **Separate Guarantee Settings** is checked, or shown alone if unchecked. If **Separate Guarantee Settings** is unchecked, then **bitPatternBorrow [i]**should take on the value of **bitPatternNonBorrow [i]**.

Field	Explanation
bitPatternBorrow[i]= <2 digit hex number>	**i** is in the range of **0..zoneInterfaceCount-1**. This field corresponds to any of the **Raw Bit, Hex**, and **Verbal view** fields on the Policy Assignment tab —> Bandwidth tab. This is the set of fields on the right hand side when **Separate Guarantee Settings** is checked, or shown alone if unchecked. If **Separate Guarantee Settings** is unchecked, then **bitPatternBorrow [i]** should take on the value of **bitPatternNonBorrow [i]**.
routeCount= <number of static routes>	This count is the number of rows on the Static Routes tab.
sourcePartition[i]= <partition>	**i** is in the range of **0..routeCount-1**. This field corresponds to the **Partition**field on the Static Routes tab. showVLANView must be checked to see this field on the LSMS GUI.
routeDisable[i]=true|false	**i** is in the range of **0..routeCount-1**. This field corresponds to the **Route Active**field on the Static Routes tab. **true** means that the route is disabled.
destinationNetwork [i]=true|false	**i** is in the range of **0..routeCount-1**. On the Static Routes tab, this field corresponds to the **Destination** field when **showVLANView** is checked, and **Destination IP Address/Mask** when it is not checked.
gatewayIP[i]= <IP address>	**i** is in the range of **0..routeCount-1**. On the Static Routes tab, this field corresponds to the **Next Hop** field when **showVLANView** is checked and the **Gateway IP**radio button is checked in the Brick Static Route Editor, or it corresponds to the **Gateway IP Address** field when **showVLANView** is not checked.
nextHopPartition [i]= <partition>	**i** is in the range of **0..routeCount-1**. On the Static Routes tab, this field corresponds to the **Next Hop** field when the Partition radio button is checked in the Brick Static Route Editor. **showVLANView** must be checked to see this field on the LSMS GUI.

Field	Explanation
routeDescription[i]=	**i** is in the range of **0..routeCount-1**. This field corresponds to the **Description** field on the Static Routes tab.
verifyRoute[i]=true|false	**i** is in the range of **0..routeCount-1**. Set this field to `true` to enable cost-based routing.
routeCost[i]=<0-32767>	**i** is in the range of **0..routeCount-1**. The Brick uses the lowest cost, available route.
routPintDestAddr[i]=<IP Address>	**i** is in the range of **0..routeCount-1**. The IP address of the router or other device to be pinged by the Brick to determine if this route is still available.
routePingSrcAddr[i]=<IP Address>	**i** is in the range of **0..routeCount-1**. The source IP address of the Brick interface from which the ping will originate (either a VBA for the Brick, interface/VLAN address, pppoe1, pppoe2, or dhcp).
routePingInterval[i]=<2-999>	**i** is in the range of **0..routeCount-1**. The time interval for sending a ping, in seconds. The default value is 10.
routePingTimeout[i]=<1-99>	**i** is in the range of **0..routeCount-1**. The maximum time to wait for a ping response, in seconds. The default value is 1.
routePingMaxFail[i]=<1-999>	**i** is in the range of **0..routeCount-1**. The number of consecutive responses to fail before the route is declared to be unavailable. The default value is 3.
proxycount= `<number of static routes>`	This count is the number of rows on the Proxies tab. To see all rows, uncheck the **Hide System Proxy Entities** checkbox.
zone[i]= `<brick ruleset>`	**i** is in the range of **0..proxyCount-1**. This field corresponds to the **Zone** field on the Proxies tab.
service[i]= `<proto/dest port/src port>`	**i** is in the range of **0..proxyCount-1**. This field corresponds to the *Service* field on the Proxies tab.
proxyDescription[i]= `<one line description>`	**i** is in the range of **0..proxyCount-1**. This field corresponds to the **Description** field on the Proxies tab.

Field	Explanation
proxyIP[i]= <IP address> |@ManageServer	**i** is in the range of **0..proxyCount-1**. This field corresponds to the **Proxy IP** field on the Proxies tab. The value **LSMS_IP** in the GUI corresponds to **@ManageServer** in the file.
proxyPort[i]= <port>	**i** is in the range of **0..proxyCount-1**. This field corresponds to the **Proxy Port** field on the Proxies tab.
encrypt[i]=true|false	**i** is in the range of **0..proxyCount-1**. This field corresponds to the **Encrypt** field on the Proxies tab.
thekey[i]= <16 hex chars>	**i** is in the range of **0..proxyCount-1**. This field corresponds to the **Key** field on the Proxies tab.
reflectionType[i]=single|dual	**i** is in the range of **0..proxyCount-1**. This field corresponds to the **Reflection** field on the Proxies tab.
passNoLPA[i]=true|false	**i** is in the range of **0..proxyCount-1**. This field corresponds to the If **LPA Unavailable Pass Traffic** checkbox on the Proxies tab. true means that the checkbox is checked.
enableICM=true|false	This field corresponds to the checkbox labeled **Enable Intelligent Cache Management** on the Cache Mgmt tab. **true** means that the checkbox is checked.
activationThreshold= <1-100>	This field corresponds to the **Global Activation Threshold** field on the Cache Mgmt tab.
targetFloorUtiliziation= <0-99>	This field corresponds to the **Target Floor Utilization** field on the Cache Mgmt tab.
icmCount= <number of cache mgmt entries>	This count is the number of rows on the Cache Mgmt tab.
icmName[i]= <name>	**i** is in the range of **0..icmCount-1**. This field corresponds to the **Name** field on the Cache Mgmt tab.
icmDescription[i]= <one line description>	**i** is in the range of **0..icmCount-1**. This field corresponds to the **Description** field on the Cache Mgmt tab.

<table>
<tr><th>Field</th><th>Explanation</th></tr>
<tr><td>icmService[i]=tcp|udp|icmp|*</td><td>i is in the range of 0..icmCount-1.
This field corresponds to the Service field on the Cache Mgmt tab.</td></tr>
<tr><td>icmThreshold[i]= <0-100></td><td>i is in the range of 0..icmCount-1.
This field corresponds to the Threshold field on the Cache Mgmt tab.</td></tr>
<tr><td>icmAudit[i]=yes|no|any</td><td>i is in the range of 0..icmCount-1.
This field corresponds to the Audit field on the Cache Mgmt tab.</td></tr>
<tr><td>icmDrop[i]=yes|no|any</td><td>i is in the range of 0..icmCount-1.
This field corresponds to the Drop field on the Cache Mgmt tab.</td></tr>
<tr><td>icmHalfOpen[i]=yes|no|any</td><td>i is in the range of 0..icmCount-1.
This field corresponds to the Half-Open field on the Cache Mgmt tab.</td></tr>
<tr><td>pppoeAsRedundantPair=
true|false</td><td>This field corresponds to the Treat the two PPPoE sessons as a redundant pair checkbox on the Dynamic Addresses tab. true means that the checkbox is checked.</td></tr>
<tr><td>dhcpServers= <IP address|IP address with subnet mask>|*
dhcpServerHostGroupName= <host group></td><td>These fields correspond to the Allow DHCP responses from these servers field on the Dynamic Addresses tab. One has a value and the other is blank. * is only allowed if the DHCP Request Method is Broadcast Discover.</td></tr>
<tr><td>dhcpAdddresses= <IP address|IP address with subnet mask>|*
dhcpAddressHostGroupName= <host group></td><td>These fields correspond to the Allow DHCP addresses in rangefield on the Dynamic Addresses tab. One has a value and the other is blank.</td></tr>
<tr><td>dhcpMethod=broadcast|unicast</td><td>This field corresponds to the DHCP Request Method radio buttons on the Dynamic Addresses tab.</td></tr>
<tr><td>pppoe1UserId= <user id></td><td>This field corresponds to the User ID field in the PPPoE #1 Options of the Dynamic Addresses tab.</td></tr>
<tr><td>pppoe1Password= <passwd></td><td>This field corresponds to the PAP Password field inthe PPPoE #1 Options of the Dynamic Addresses tab.</td></tr>
</table>

Field	Explanation
pppoe1MACAddr=	This field corresponds to the **MAC Address** field in the PPPoE #1 Options of the Dynamic Addresses tab.
pppoe1KeepAliveIntv1= <number in secs>	This field corresponds to the **Keep-Alive Interval** field in the PPPoE #1 Options of the Dynamic Addresses tab.
pppoe1KeepAliveRetryCnt= <count>	This field corresponds to the **Keep-Alive Retry Count**field in the PPPoE #1 Options of the Dynamic Addresses tab.
pppoe2UserId= <user id>	This field corresponds to the **User ID** field in the PPPoE #2 Options of the Dynamic Addresses tab.
pppoe2Password= <passwd>	This field corresponds to the **PAP Password field** inthe PPPoE #2 Options of the Dynamic Addresses tab.
pppoe2MACAddr=	This field corresponds to the **MAC Address** field in the PPPoE #2 Options of the Dynamic Addresses tab.
pppoe2KeepAliveIntv1= <number in secs>	This field corresponds to the **Keep-Alive Interval** field in the PPPoE #2 Options of the Dynamic Addresses tab.
pppoe2KeepAliveRetryCnt= <count>	This field corresponds to the **Keep-Alive Retry Count**field in the PPPoE #2 Options of the Dynamic Addresses tab.
pppoe2ChapKey= <hex chars>	This field corresponds to the **CHAP Key** field in thePPPoE #2 Options of the Dynamic Addresses tab. The LSMS GUI allows you to view it in several representations. The CLI must have this value represented in hex.
auditWait=true|false	This field corresponds to the **Halt All Traffic If Audit Fails** checkbox on the Options tab. *true* means that the checkbox is checked.
multicastToFirstZone=true|false	This field corresponds to the **Route Multicast Packets To First Matching Zone** checkbox on the Options tab. **true** means that the checkbox is checked.
autoRefreshMac=false	This field corresponds to the **Allow MAC Addresses To Move** checkbox on the Options tab. **true** means that the checkbox is checked.

Field	**Explanation**
useBrickAddr=true|false	This field corresponds to the Use **Brick Address Instead OfPort Address During Bootstrap** checkbox on the Options tab. **true** means thatthe checkbox is checked.
routeReturn=true|false	This field corresponds to the **Route Return Path Packets toCached Source MAC Address** checkbox on the Options tab. **true** means that the checkbox is checked.
enableBrickFailover=true|false	This field corresponds to the **Enable Brick Failover** checkbox on the Failover tab. **true** means that the checkbox is checked.
failoverActvTime= `<number of 10ths of secs>`	This field corresponds to the **Activation Time** field on the Failover tab.
failoverYldTime= `<number in 10ths of secs>`	This field corresponds to the **Yield Time** field on the Failover tab.
failoverPrfStshInt=ether <n> |auto	This field corresponds to the **Preferred State-Sharing Port** field on the Failover tab.
encryptPreferredLink=true|false	This field corresponds to the two radio buttons **Encrypt all links and Encrypt all links except Preferred State-Sharing link** on the Failover tab. Checking **Encrypt all links** means **true**. Checking **Encrypt all links except Preferred State-Sharing link** means **false**.
macAddressA=<2-6 hex octets, colon separated>	This field corresponds to the **Ether 0 MAC Address** field of Brick A on the Failover tab. If fewer than six octets are entered, the Brick will try to match this with the rightmost portion of the MAC address.
macAddressB=<2-6 hex octets, colon separated>	This field corresponds to the **Ether 0 MAC Address** field of Brick B on the Failover tab. If fewer than six octets are entered, the Brick will try to match this with the rightmost portion of the MAC address.
failoverLabelA=<up to 16 char label>	This field corresponds to the **Failover Label** field of Brick A on the Failover tab.
failoverLabelB=<up to 16 char label>	This field corresponds to the **Failover Label** field of Brick B on the Failover tab.
primaryBrick=<brickA, brickB, none>	This field corresponds to the Brick designated as Primary (if any) on the Failover tab.

Field	Explanation
failbackDelay=<number in seconds>	The length of time the Bricks must be in continuous contact before the failover to the Primary Brick is initiated. This field corresponds to the **Fail Back Delay (secs)** field on the Failover tab.
pingMinActive=<number in seconds>	The minimum time, in seconds, that a Brick must be active before initiating a failover due to loss of ping. This field corresponds to the **Min Time active before Failover (secs)** field on the Failover tab.
pingFailoverCount=<number of ping failover entries>	This count is in the number of rows in the Ping Failover Table.
pingFoDestinationIP[i]=<IP address>	*i* is in the range *0..pingFailoverCount-1*. The IP address the Brick will attempt to ping. This field corresponds to the **Ping Destination IP Address** field on the Failover tab.
pingFoSourceIP[i]=<IP address>	*i* is in the range *0..pingFailoverCount-1*. The IP address the Brick will use as the source address for the ping. This field corresponds to the **Ping Source IP Address** field on the Failover tab.
pingFoInterval[i]=<number in seconds>	*i* is in the range *0..pingFailoverCount-1*. The interval, in seconds, at which to send the ping. This field corresponds to the **Ping Interval** field on the Failover tab.
pingFoTimeout[i]	*i* is in the range *0..pingFailoverCount-1*. The maximum time to wait for a ping responses. This field corresponds to the **Ping Timeout** field on the Failover tab.
pingFoFailureThreshold[i]=<number of pings>	*i* is in the range *0..pingFailoverCount-1*. The number of consecutive responses to pings that are required to fail before initiating a failover. This field corresponds to the **Ping Failures** field on the Failover tab.
pingFoPartition[i]=partition name	*i* is in the range *0..pingFailoverCount-1*. The partition from which to initiate the ping. This field corresponds to the **Ping Partition** field on the Failover tab.
pingFoDescription[i]=<description>	*i* is in the range *0..pingFailoverCount-1*. A description for this ping failover entry. This field corresponds to the **Description** field on the Failover tab.

Field	Explanation
enableMsgsNoLogin=true|false	This field corresponds to the **Enable Messages to Serial Port Without Logging In** checkbox on the Options tab. **true** means that the checkbox is checked.
remoteLoginID= <password>	This field corresponds to the **Password** field on theOptions tab. If the **remoteLoginID** is not blank, the list command returns its value as the hardcoded string **lv**. When you save it with this value, the backend software does not modify its actual value. Saving it with any other value causes the **remoteLoginID** to change to that value.
version= <brick version string>	No correspondence to a Brick GUI field. Informational only. Its value is not saved back into the LSMS database.
certType= <certificate type>	No correspondence to a Brick GUI field. Informational only. Itsvalue is not saved back into the LSMS database.
skipRouteCheck=true|false	Setting this field to true allows you to ignore error N7049, which is a complaint about a missing return route between two partitions. The line with this name/value pair is optional. If missing for a **save brick, skipRouteCheck** defaults to **false**.

Example

```
lsmscmd add brick brick33
```

This command creates a new Brick named ″brick33″.

□

add brickruleset

Overview

The `add brickruleset` command adds a new Brick ruleset with the specified Brick ruleset name to the current group.

Format

The format of the `add brickruleset` command is:

```
lsmscmd add brickruleset <brick ruleset name> [folder name]
```

where:

- *<brick ruleset name>* is the name of the new brick ruleset.This arguement is required.
- [folder name] is the name of the folder into which you would like the Brick ruleset added. This argument is optional.

Explanation

A Brick ruleset is the set of rules that govern traffic through the bricks being administered.

Example

```
lsms add brickruleset sales
```

This command creates a brick ruleset called *sales*.

add clienttunnel

Overview

Creates a new Client tunnel with the given name to the LSMS.

Format

```
add clienttunnel <clientTunnelName>
```

Explanation

When this command is run, a file called `<clientTunnelName>`containing a new Client tunnel configuration must exist in the *<cli_dir>/<group>/VPN/Client_Tunnels*folder.

Example 1

```
add clienttunnel pppoe1Tunnel
```

The following is a typical example of output for this command:

```
ADD CLIENT TUNNEL: OK
```

Example 2

```
add clienttunnel 13.45.43.2
```

The following is a typical example of output for this command:

```
ADD CLIENT TUNNEL:N2013: Duplicate; already exists
```

This error usually means that the value given for the localTep is already used in some other client tunnel.

□

add dependencymasks

Overview

The `add dependencymasks` command adds a new dependency mask with the specified name to the current group.

Format

The format of the `add dependencymasks` command is:

`lsmscmd add dependencymasks` *`<dependencymask name>`*

where:

- *`<dependencymask name>`* is the name of the new dependency mask

Explanation

A dependency mask is a feature that allows an administrator to create a rule that will not permit a matching packet through the brick until it verifies that a particular session is found in the session cache.

Example

```
lsmscmd add dependencymasks client
```

This command adds a dependency mask named *client*.

□

add hostgroup

Overview

The `add hostgroup` command adds a new host group with the specified name.

Format

The format of the `add hostgroup` command is:

```
lsmscmd add hostgroup <hostgroup name> [foldername]
```

where:

- *<hostgroup name>* is the name of the new host group
- *[folder name]* is the folder into which you would like the new host group added. This argument is optional.

Explanation

A host group is a collection of IP addresses grouped together to enable you to enter more than one address in situations where a list of host IP addresses is required.

Example

```
lsmscmd add hostgroup marketing
```

This command adds a host group named *marketing*.

add lan2lantunnel

Overview

Creates a new LAN-to-LAN tunnel with the given name to the LSMS.

Format

`add lan2lantunnel [<lan2lanTunnelName>]`

where *<lan2lanTunnelName>* is the name of the tunnel.

Explanation

When this command is run, a file called *<lan2lanTunnelName>* containing a new LAN-to-LAN tunnel configuration must exist in the *<cli_dir>/<group>/VPN/Lan2Lan_Tunnels* folder.

Example 1

`add lan2lantunnel custTunnel`

The following is a typical example of output for this command:

```
ADD LAN2LAN TUNNEL: OK
```

Example 2

`add lan2lantunnel 13.45.43.2_23.45.62.198`

The following is a typical example of output for this command:

```
ADD LAN2LAN TUNNEL:N2013: Duplicate; already exists
(This error usually means that the values given for the localTep/
   remoteTep combination or its reverse is already used in some other
   LAN-to-LAN tunnel.)
```

☐

add servicegroup

Overview

The `add servicegroup` command adds a new service group with the specified service group name.

Format

The format of the `add servicegroup` command is:

`lsmscmd add servicegroup` *<servicegroup name>* *[folder name]*

where:

- *<servicegroup name>* is the name of the new service group.
- *[folder name]* is the folder into which you would like the dependency mask added. This argument is optional.

Explanation

A service group is a collection of services and protocols. It is used to identify the services in a rule and perform destination port mapping when adding network address translation to a rule.

Example

```
lsmscmd add servicegroup special
```

This command adds a service group named *special*.

□

apply brick

Overview

The `apply brick` command applies any changed rulesets to their assigned ports on the specified Brick in your current group.

Format

The format of the `apply brick` command is

```
lsmscmd apply brick<brick name> <cache option -clear keep>
```

where:

- *<brick name>* is the name of the Brick to which you want its policy applied.
- *<cache option -clear keep>* is either `clear` (clear session cache) or `keep` (do not clear session cache)

Explanation

The *<cache>* argument is required. The default is `keep`.This means, if you do not enter this argument, the session cache for each Brick in the current group will not be cleared.

If you enter a `clear` in the command string, the session cache for each Brick in the current group will be cleared when you execute the command.

Keep in mind that clearing the cache can disrupt existing client tunnel sessions, FTP sessions, or sessions that rely on dependency masks. Note that when the cache is cleared, tunnels remain intact, but the sessions that use them are lost.

Executing this comand is equivalent to saving and applying a Brick in the LSMS GUI.

Example 1

```
lsmscmd apply brick sales1 keep
```

This command applies any changed Brick rulesets to all assigned brick ports in the current group. Since the *<keep>* argument is included, the cache is not cleared (the default).

Example 2

```
lsmscmd apply brick sales2 clear
```

This command applies any changed Brick rulesets to all assigned bricks in the current group. The *<cache>* argument is set to `clear`, indicating the cache will be cleared.

□

apply brickruleset

Overview

The `apply brickruleset` command compiles and applies the security policy to any Brick ports in your current group that have older versions of the specified Brick ruleset.

Format

The format of the `apply brickruleset` command is

```
lsmscmd apply brickruleset <brick ruleset name> <cache -clear keep>
```

where:

- *<brick ruleset name>* is the name of the Brick ruleset you want to apply
- *<cache -clear keep>* is either `clear` (clear session cache) or `keep` (do not clear session cache)

Explanation

The *<cache option>* argument is required. The default is `keep`.This means, if you do not enter this argument, the session cache for each Brick in the current group will not be cleared.

If you enter a `clear` in the command string, the session cache for each Brick in the current group will be cleared when you execute the command.

Keep in mind that clearing the cache can disrupt existing sessions that use dependency masks.

Example 1

```
lsmscmd apply brickruleset proxyzone keep
```

This command applies the ruleset *proxyzone* to all Brick ports that have previously had this ruleset applied. Since the *<keep>* argument is included, the cache is not cleared (the default).

Example 2

```
lsmscmd apply brickruleset proxyzone clear
```

This command applies the ruleset *proxyzone* to all Brick ports that have previously had this ruleset applied. The *<cache>* argument is set to `clear`, indicating the cache will be cleared.

□

apply group

Overview

The `apply group` command compiles and applies any changed rulesets and devices to your current group.

Format

The format of the `apply group` command is:

```
lsmscmd apply group
```

Explanation

The `apply group` command looks at all devices and rulesets in the current group to see if any have changed since they were last applied. The command then compiles and applies any devices and rulesets that have changed.

Executing this command is equivalent to applying a group in the LSMS GUI.

Example

```
lsmscmd apply group test
```

This command compiles all device and ruleset information for the group named ″test″ and applies the information.

boot brick

Overview

The `boot brick` command allows an administrator to reboot a Brick from the command line.

Format

The format of the `boot brick` command is:

lsmscmd boot brick <*brickname*>

where <*brickname*> is the name of the brick you want to reboot.

Explanation

The `boot brick` command is used to reboot a single Brick from the command line.

Note that you should alert other administrators that secure client tunnels to this Brick will be terminated, and that users will need to reestablish connections after the Brick comes back online.

Executing this command is equivalent to selecting **Reboot** from the Brick Utilities menu in the Brick Editor, or right-clicking a Brick in the LSMS Navigator window and selecting **Reboot** from the pop-up menu.

Example

```
lsmscmd boot brick brick22
```

This command reboots a Brick named ″brick22.″

☐

bootgroup

Overview

The `bootgroup` command reboots all Bricks in the current group.

Format

The format of the `bootgroup` command is:

```
lsmscmd bootgroup
```

Explanation

Use the `bootgroup` command to reboot all Bricks in the current group.

The `bootgroup` command can be used following the `downloadgroup` command to reboot all bricks that have been upgraded.

Note that you should alert other administrators that secure client tunnels to bricks in this group will be terminated, and that users will need to reestablish connections after the Bricks come back online.

Example

```
lsmscmd bootgroup
```

This command reboots all Bricks in the current group.

□

delete applicationfilter

Overview

The `delete applicationfilter` command deletes an application filter with the specified name.

Format

The format of the delete `applicationfilter` command is:

`lsmscmd delete applicationfilter <application filter name>`

where:

- *`<application filter name>`* is the name of the application filter to be deleted.

Explanation

The `delete application filter` command is used to remove obsolete or otherwise non-functional rulesets.

Example

```
lsmscmd delete application filter H323
```

This command deletes an application filter called *H323*.

□

delete brick

Overview

The `delete brick` command deletes a Brick from the LSMS database.

Format

The format of the `delete brick` command is:

`lsmscmd delete brick` *<brick name>*

where:

- *<brick name>*is the name of the Brick to be deleted.

Explanation

Use the `delete brick` command to delete a Brick instance from the LSMS database.

Example

`lsmscmd delete brick radbrick`

This command deletes a Brick named ″radbrick″ from the LSMS database.

□

delete brickruleset

Overview

The `delete brickruleset` command deletes a Brick ruleset with the specified name.

Format

The format of the `delete brickruleset` command is:

`lsmscmd delete brickruleset` *`<brick ruleset name>`*

where:

- `<brick ruleset name>` is the name of the Brick ruleset to be deleted.

Explanation

The `delete brickruleset` command is used to remove obsolete or otherwise non-functional rulesets.

Example

```
lsmscmd delete brickruleset sales
```

This command deletes a Brick ruleset called *sales*.

☐

delete clienttunnel

Overview

Deletes a Client tunnel with the given name to the LSMS.

Format

```
delete clienttunnel <clientTunnelName>
```

Explanation

This command removes a tunnel from services permanently, deleting its configuration from the LSMS.

Example

```
delete clienttunnel pppoe1Tunnel
```

The following is a typical example of output for this command:

```
DELETE CLIENT TUNNEL: OK
```

□

delete dependencymasks

Overview

The `delete dependencymasks` command deletes a dependency mask with the specified name.

Format

The format of the `delete dependencymasks` command is:

`lsmscmd delete dependencymasks` *`<dependency mask name>`*

where:

- *`<dependency mask name>`* is the name of the dependency mask to be deleted.

Explanation

The `delete dependencymasks` command is used to remove obsolete or otherwise non-functional dependency masks.

Example

```
lsmscmd delete dependencymasks client
```

This command deletes a dependency mask called *client*.

□

delete hostgroup

Overview

The `delete hostgroup` command deletes the host group with the specified host group name.

Format

The format of the `delete hostgroup` command is:

`delete hostgroup` *`<hostgroup name>`*

where:

- *`<hostgroup name>`* is the name of the host group to be deleted.

Explanation

The `delete hostgroup` command is used to remove obsolete or otherwise non-functional hostgroups.

Example

```
lsmscmd delete hostgroup marketing
```

This command deletes a host group called *marketing*.

delete lan2lantunnel

Overview

Deletes a LAN-to-LAN tunnel with the given name to the LSMS.

Format

```
delete lan2lantunnel [<lan2lanTunnelName>]
```

where *<lan2lanTunnelName>* is the name of the tunnel.

Explanation

This command removes a tunnel from service permanently, deleting its configuration from the LSMS.

Example 1

```
delete lan2lantunnel custTunnel
```

The following is a typical example of output for this command:

```
DELETE LAN2LAN TUNNEL: OK
```

□

delete servicegroup

Overview

The `delete servicegroup` command deletes a service group with the specified name.

Format

The format of the `delete servicegroup` command is:

`lsmscmd delete servicegroup <service group name>`

where:

- *`<service group name>`* is the name of the service group to be deleted.

Explanation

The `delete servicegroup` command is used to remove obsolete or otherwise non-functional service groups.

Example

`lsmscmd delete servicegroup special`

This command deletes a service group called *special*.

□

disable clienttunnel

Overview

Disables a Client tunnel with the given name to the LSMS.

Format

```
disable clienttunnel <clientTunnelName>
```

Explanation

This command takes a Clkient tunnel out of service without deleting it outright.

Example

```
disable clienttunnel pppoe1Tunnel
```

The following is a typical example of output for this command:

```
DISABLE CLIENT TUNNEL: OK
```

□

disable lan2lantunnel

Overview

Disables a LAN-to-LAN tunnel with the given name to the LSMS.

Format

```
disable lan2lantunnel <lan2lanTunnelName>
```

Explanation

This command takes a LAN-to-LAN tunnel out of service without deleting it outright.

Example

```
disable lan2lantunnel custTunnel
```

The following is a typical example of output for this command:

```
DISABLE LAN2LAN TUNNEL: OK
```

download brick

Overview

The `download brick` command is used to update Brick software after an LSMS software upgrade.

Format

The format of the `download brick` command is:

`lsmscmd download brick` *`<brick name> [active|both]`*

where:

- `<brick name>` is the name of the Brick to which you want to download the new software version.
- [active|both] refers to the case where a failover pair of Bricks is upgraded. Use the **active** option to upgrade only the active brick of the failover pair. Use the **both** option to upgrade both Bricks of the failover pair. The default option is **active**.

Explanation

The `download brick` command updates the software on a Brick after the LSMS software has been updated. Periodically, Lucent will release patches, point releases, or major new versions of the LSMS software which will be made available on CD-ROM or from a website. Brick software must be updated after such an LSMS revision is installed.

Executing this command is equivalent to selecting **Software Download** from the Brick Utilities menu in the Brick Editor, or right-clicking a brick in the LSMS Navigator window and selecting **Software Download** from the pop-up menu.

In the case of a failover pair of Bricks, if you use the **both** option, when the download to the active Brick is complete, issue a failover yield command or reboot the Brick, causing the standby brick to become active. A download to the newly active Brick will begin. When this second download is complete, reboot that Brick, causing a second failover, so that the Brick that was initially in the active state is returned to the active state. Both Bricks of the pair will now have the updated software.

If you use the **active** option, which is the default, the active Brick will be upgraded and you will need to reboot it to make the updated software active. When you reboot, failover will occur. The newly active Brick will still need to be upgraded at a later date.

Example

```
lsmscmd download brick brick44
```

This command updates the Brick software on a Brick named ″brick44.″

□

downloadgroup

Overview

The `downloadgroup` command is used to update Brick software on all Bricks in the current group after an LSMS software upgrade.

Format

The format of the `downloadgroup` command is:

```
lsmscmd downloadgroup
```

Explanation

The `downloadgroup` command updates Brick software on all Bricks in the current group after a software update has been applied to the LSMS software.

Periodically, Lucent will release patches, point releases, or major new versions of the LSMS software which will be made available on CD-ROM or from a website. Brick software must be updated after such an LSMS revision is installed.

Example

```
lsmscmd downloadgroup
```

This command updates the Brick software on all Bricks in the current group.

□

enable clienttunnel

Overview

Enables a Client tunnel with the given name to the LSMS.

Format

`enable clienttunnel <clientTunnelName>`

Explanation

This command puts a Client tunnel back in service that had been previously disabled.

Example

```
enable clienttunnel pppoe1Tunnel
```

The following is a typical example of output for this command:

```
ENABLE CLIENT TUNNEL: OK
```

enable lan2lantunnel

Overview

Enables a LAN-to-LAN tunnel with the given name to the LSMS.

Format

```
enable lan2lantunnel <lan2lanTunnelName>
```

Explanation

This command puts a LAN-to-LAN tunnel back in service that had been previously disabled.

Example

```
enable lan2lantunnel custTunnel
```

The following is a typical example of output for this command:

```
ENABLE LAN2LAN TUNNEL: OK
```

□

failover brick

Overview

The `failover brick` command initiates failover of a failover Brick pair.

Format

The format of the `failover brick` command is:

```
lsmscmd failover brick <brickName> [force]
```

where:

- *<brickName>* is the name of the Brick failover pair.

Explanation

A failover Brick pair is a redundant Brick installation in both Bricks share a single Brick name and IP address. One Brick is the ″active″ brick, while the second is the ″standby.″ Should the active Brick fail for any reason, the standby immediately takes over the active role. This transition from standby to active is termed ″failover.″ The `failover brick` command issued from the LSMS CLI initiates failover.

Example

```
lsmscmd failover brick brick_pair_one
```

This command initiates failover of the Brick pair named brick_pair_one.

□

gotogrp

Overview

The `gotogrp` command enables an Administrator who has privileges in multiple groups to select the specified group as his or her current group.

Format

The format of the `gotogrp` command is:

```
lsmscmd gotogrp <group name>
```

where:

- *<group name>* is the name of the group that the Administrator wants to make his or her current group.

Explanation

Executing the `gotogrp` command is equivalent to choosing another group folder on the GUI.

Example

```
lsmscmd gotogrp sales
```

This command makes *sales* the cuurent group for the Administrator.

□

help

Overview

The `help` command displays a brief description of the syntax of each command. Note that you can issue the lsmscmd help command without loggin in to the LSMS first.

Format

The format of the `help` command is:

`lsmscmd help <command name>`

The <*command name*> argument is optional. Issuing the `lsmscmd help` command without the <*command name*> argument returns the entire set of LSMS commands.

Explanation

The `help` command provides a brief explanation of the syntax of each command, including the optional and required parameters.

You cannot use the `help` command to display the syntax of a specific command. It can only display help for all commands.

Example

```
lsmscmd help
```

The Help command issued without an argument returns the entire list of LSMS commands, as shown below:

```
The various commands and their syntax are as follows.

Commands for logging in and logging out

1> lsmslogon <admin_id> <destination-directory>
      [-p password_admin_key_file or -f <password> -a <admin_key>] [-
  t <logon_shell_timeout_in_secs>]
2> lsmscmd logout
System Administration Commands

3> lsmscmd
  help
4> lsmscmd gotogrp <groupName>
5> lsmscmd list current group
6> lsmscmd list groups
7> lsmscmd list brick <brickName>
8> lsmscmd save brick <brickName>
9> lsmscmd add brick <brickName> [folderName]
8> lsmscmd list brickruleset <brickRulesetName>
9> lsmscmd save brickruleset <brickRulesetName>
10> lsmscmd delete brick <brickName>
11> lsmscmd list brickruleset <brickRulesetName>
12> lsmscmd save brickruleset <brickRulesetName>
13> lsmscmd add brickruleset <brickRulesetName> [folderName]
14> lsmscmd delete brickruleset <brickRulesetName>
15>lsmscmd list brickteps <brickName>
16> lsmscmd list servicegroup <servicegroupname>
17> lsmscmd save servicegroup <servicegroupname>
18> lsmscmd add servicegroup <servicegroupname> [folderName]
19> lsmscmd delete servicegroup <servicegroupname>
20> lsmscmd list domainnamegroup <domainnamegroupname>
21> lsmscmd save domainnamegroup <domainnamegroupname>
22> lsmscmd add domainnamegroup <domainnamegroupname> [folderName]
23> lsmscmd delete domainnamegroup <domainnamegroupname>
24> lsmscmd list applicationfilter <applicationfiltername>
25> lsmscmd save applicationfilter <applicationfiltername>
26> lsmscmd add applicationfilter <applicationfiltername>
  [folderName]
27> lsmscmd delete applicationfilter <applicationfiltername>
28> lsmscmd list hostgroup <hostgroupname>
29> lsmscmd save hostgroup <hostgroupname>
30> lsmscmd add hostgroup <hostgroupname> [folderName]
31> lsmscmd delete hostgroup <hostgroupname>
29> lsmscmd list dependencymasks <dependencymaskname>
30> lsmscmd save dependencymasks <dependencymaskname>
31> lsmscmd add dependencymasks <dependencymaskname>
32> lsmscmd list dependencymasks <dependencymaskname>
```

```
33> lsmscmd save dependencymasks <dependencymaskname>
34> lsmscmd add dependencymasks <dependencymaskname>
35> lsmscmd delete dependencymasks <dependencymaskname>
36> lsmscmd list clienttunneldefaults
37> lsmscmd save clienttunneldefaults
38> lsmscmd list clienttunnel [<clientTunnelName>]
39> lsmscmd list unusedclientteps <brickName>
40> lsmscmd list clientlicenselimits
41> lsmscmd save clientlicenselimits
42>lsmscmd save clienttunnel <clientTunnelName>
43> lsmscmd add clienttunnel <clientTunnelName>
44> lsmscmd delete clienttunnel <clientTunnelName>
45> lsmscmd enable clienttunnel <clientTunnelName>
46> lsmscmd disable clienttunnel <clinetTunnelName>
47> lsmscmd list lan2lantunneldefaults
48> lsmscmd save lan2lantunneldefaults
49> lsmscmd list lan2lantunnel [<lan2lanTunnelName>]
50> lsmscmd save lan2lantunnel <lan2lanTunnelName>
51> lsmscmd add lan2lantunnel <lan2lanTunnelName>
52> lsmscmd delete lan2lantunnel <lan2lanTunnelName>
53> lsmscmd enable lan2lantunnel <lan2lanTunnelName>
54> lsmscmd disable lan2lantunnel <lan2lanTunnelName>
55> lsmscmd apply brickRuleset <brickRulesetName> <keep|clear>
56> lsmscmd apply group
57> lsmscmd apply brick <brickName> <keep|clear>
58> lsmscmd boot brick <brickName>
59> lsmscmd download brick <brickName> [active|both]
60> lsmscmd bootgroup
61> lsmscmd downloadgroup [active|both]
62> lsmscmd failover brick <brickName> [force]
63> lsmscmd refreshmac brick <brickName>
64> lsmscmd rehome brick <brickName> <LsmsIpAddress>NOTES
```

Important! Notes:

1. Parameters between '<' and '>' are required, and the parameters between '[' and ']' are optional.
2. In command 1, if no password is provided on the command line or in the password_admin_key file, the user will be prompted.
3. In command 1, if an admin_key is required (e.g., for RADIUS or SecurID), provide it on the command line or add it as the 2nd line in the password_admin_key file. If no admin_key is provided and one is required, the user will be prompted.
4. In command 1, if no timeout is given, the GUI timeout is the default timeout value used.

5. In commands 9, 13, 19, 18, 22, 26, 30 the default folderName is the destination_directory given in command 1.
6. In commands 59, 61, the default value is both.

□

list applicationfilter

Overview

The `list applicationfilter` command lists the contents of the desired application filter.

Format

The format of the `list applicationfilter` command is:

`lsmscmd list current group`

Explanation

Use the `list current group` command to determine what group you are currently administering.

Example

`lsmscmd list current group`

The following is an example of typical output from this command:

```
LIST CURRENT GROUP:OK:
Group ='system'
```

☐

list brick

Overview

The `list brick` command displays the configuration information for the specified Brick. If you supply the *`<brick name>`* argument, the output of the command is an ASCII file that is saved in the directory you specified at logon.

Format

The format of the `list brick` command is:

`lsmscmd list brick <brick name`

where:

- *`<brick name>`* is the name of the Brick for which configuration information is to be displayed.

Explanation

The complete list of Bricks available to the current group can be obtained by omitting the *`<brick name`* argument. However, this list will only be echoed to the console; no file will be written to the target directory if you omit the argument.

After you issue a `list brick` command, you can open the file with any text editor, edit the entries, and issue a `save brick` command. See "save brick" (p. 2-81) later in this chapter for details on this command.

All of the configuration information for a Brick is specified in a file. The name of the file is *`<brick name>`* and is located on the LSMS in the directory *<cli dir>/<group>/Devices/Brick*, where *`<cli dir>`* is given as an argument to the `lsmslogon` command, and *`<group>`* is the current group that you are in when executing the `add brick` command. By default, the group is **system.**

Data in the Brick configuration file is organized in a *`<name>=<value>`* format. Refer to the descripton of the "add brick" (p. 2-6) command for details about the contents of the configuration file.

The list command does not specify or guarantee a certain order for the lines in the file.

Example

`lsmscmd list brick sales1`

The following is an example of typical output from this command:

```
LIST BRICKS:OK
```

Important! The list commands create a directory structure, within the directory that you specified in the lsmslogon command, that corresponds to the LSMS directory structure.

Thus, issuing the list brick command will create the directory tree /System/Devices/Bricks in your target directory.

□

list brickruleset

Overview

The `list brickruleset` command gets the list of rules that belong to the Brick ruleset. If you supply the *brick ruleset name* `argument`, the output of the command is an ASCII file that is saved in the directory you specified at logon.

Format

The format of the `list brickruleset` command is:

`list brickruleset <brick ruleset name>`

where:

- *<brick ruleset name>* is the name of the Brick ruleset for which you want to see a list of rules.

Explanation

Executing this command is equivalent to right-clicking a particular Brick zone ruleset in the LSMS Navigator. You can get a complete list of Brick rulesets available in your current group by omitting the *<brick ruleset name>* argument. This is equivalent to clicking the Brick Zone Rulesets folder on the GUI. Note that this list will only be echoed to the console; no file will be written to the target directory if you omit the argument.

After you issue a `list brickruleset` command, you can open the file with any text editor, edit the entries, and issue a `save brickruleset` command. See "save brickruleset" (p. 2-82) later in this chapter for details on this command.

Example 1

```
lsmscmd list brickruleset sales
```

The following is an example of typical output from this command:

```
LIST BRICK RULESET:OK
```

Example 2

```
lsmscmd list brickruleset
```

The following is an example of typical output from this command:

```
LIST BRICK RULESET:OK
1 'vpnzone'
2 'proxyzone'
3 'nocgwzone'
4 'firewall'
5 'administrativezone'
```

□

list brickteps

Overview

The `list brickteps` command lists the tunnel endpoints (TEPs) of a given Brick to the console.

Format

The format of the `list brickteps` command is:

`lsmscmd list brickteps <brickName>`

where:

<brickName> is the name of a Brick in the current group. This arguement is required.

Explanation

This command helps the user know what TEPs are available for editing the localTep field of a Client tunnel configuration or the localTep or remoteTep fields of a LAN-to-LAN tunnel configuration. (For Client tunnels, the 'list unusedClientTeps' command is even more useful.)

Example

`lsmscmd list brickteps v1brick`

The following is an example of typical output from this command:

```
LIST BRICK TEPS:OK
1 '87.3.58.153, rock_zone'
2 '132.16.34.91, paper_zone'
3 '76.23.10.44, scissor_zone'
The first field in each list entry is the virtual Brick address and
  the second is the Brick ruleset.
```

□

list clientlicenselimits

Overview

The `list clientlicenselimits` command lists the number of client user licenses assigned to groups and TEPs.

Example

```
lsmscmd list clientlicenselimits
```

The following is an example of typical output from this command:

```
licenseFromKey=10100
groupName[0]=group1
groupLimit[0]=10
groupName[1]=group2
groupLimit[1]=0
groupName[2]=group3
groupLimit[2]=0
groupName[3]=group4
groupLimit[3]=0
groupName[4]=group5
groupLimit[4]=0
groupName[5]=system
groupLimit[5]=5000
groupCount=6
count=7
device[0]=brick12
tepIP[0]=12.12.12.103
activeSessionCount[0]=0
policy[0]=vpnzone12ca2
enabled[0]=true
name[0]=brick12ca3
licenseLimit[0]=10
device[1]=brick12
tepIP[1]=dhcp
activeSessionCount[1]=0
policy[1]=administrativezone
enabled[1]=true
name[1]=brick12dhcp
licenseLimit[1]=10
device[2]=brick12
tepIP[2]=12.12.12.253
activeSessionCount[2]=0
policy[2]=vpnzone12
enabled[2]=true
name[2]=brick12vpn12
licenseLimit[2]=10
device[3]=brick13
tepIP[3]=13.13.13.253
activeSessionCount[3]=0
policy[3]=vpnzone13
enabled[3]=true
name[3]=brick13vpn
licenseLimit[3]=0
device[4]=brick55
tepIP[4]=55.55.55.203
activeSessionCount[4]=0
policy[4]=vpnzone55
```

```
enabled[4]=true
name[4]=brick55vpn55
licenseLimit[4]=10
device[5]=model80
tepIP[5]=dhcp
activeSessionCount[5]=0
policy[5]=administrativezone
enabled[5]=true
name[5]=model80admin
licenseLimit[5]=100
device[6]=model80
tepIP[6]=10.10.10.251
activeSessionCount[6]=0
policy[6]=vpnzone
enabled[6]=true
name[6]=model80tep
licenseLimit[6]=1000
```

□

list clienttunnel

Objective

Retrieves the specified Client tunnel configuration to a file under the *<cli_dir>/<group>/VPN/Client_Tunnels* folder.

Format

```
list clienttunnel [<clientTunnelName>]
```

where `clientTunnelName` is the name of the tunnel. If you leave off this argument, the names of all the Client tunnels in this group will be displayed on the console.

Explanation

Use this command to list the Client tunnel configuration of a particular tunnel to a file of the same name, or have it list all the tunnel names.

Example 1

```
lsmscmd list clienttunnel
```

The following is a typical example of output for this command:

```
LIST CLIENT TUNNEL: OK
1 '132.16.34.91'
2 '13.45.43.2'
```

Example 2

```
lsmscmd list clienttunnel 13.45.43.2
```

The following is a typical example of output for this command:

```
LIST CLIENT TUNNEL: OK
```

□

list clienttunneldefaults

Objective

The `list clienttunneldefaults` command retrieves the default Client tunnel configuration to a file called ’defaults’ under the *<cli_dir>/<groups>/VPN/Client_Tunnels* folder.

Format

```
list clienttunneldefaults
```

Explanation

Use this command to view, perhaps with the intent to edit and save, the Client tunnel defaults configuration.

Example

```
lsmscmd list clienttunneldefaults
```

The following is an example of typical output from this command:

```
LIST CLIENT TUNNEL DEFAULTS:OK
```

□

list current group

Overview

The `list current group` command echoes back the group that the Administrator is currently working in.

Format

The format of the `list current group` command is:

`lsmscmd list current group`

Explanation

Use the `list current group` command to determine what group you are currently administering.

Example

`lsmscmd list current group`

The following is an example of typical output from this command:

```
LIST CURRENT GROUP:OK:
Group ='system'
```

□

list dependencymasks

Overview

The `list dependencymasks` command gets the list of parameters that belong to the specified dependency mask. If you supply the *`dependency mask name`* argument, the output of the command is an ASCII file that is saved in the directory you specified at logon.

Format

The format of the `list dependencymasks` command is:

`list dependencymasks <dependency mask name>`

where:

- *`<dependency mask name>`* is the name of the dependency mask for which you want to see a list of parameters.

Explanation

Executing this command is equivalent to right-clicking a particular dependency mask in the GUI. You can get a complete list of dependency masks available in your current group by omitting the *`<dependency mask name>`* argument. This is equivalent to clicking the Dependency Masks folder on the GUI. Note that this list will only be echoed to the console; no file will be written to the target directory if you omit the argument.

After you issue a `list dependencymasks` command, you can open the file with any text editor, edit the entries, and issue a `save dependencymasks` command. See "save dependencymasks" (p. 2-86) later in this chapter for details on this command.

Example

```
lsmscmd list dependencymasks test
```

The following is a typical example of output from this command:

```
LIST DEPENDENCYMASKS:OK
```

Important! The list commands create a directory structure, within the directory that you specified in the lsmslogon command, that corresponds to the LSMS directory structure.

Thus, issuing the `list dependencymasks` command will create the directory tree */System/Policy_Components/Dependency_Masks* in your target directory.

list groups

Overview

The `list groups` command lists the names of all groups that currently exist in the LSMS database.

Format

The format of the `list groups` command is:

```
lsmscmd list groups
```

Explanation

Use the `list groups` command to list the names of all groups that currently exist in the LSMS database.

Example

```
lsmscmd list groups
```

☐

list hostgroup

Overview

The `list hostgroup` command gets the list of IP addresses that belong to the specified host group. If you supply the *`hostgroup name`* argument, the output of the command is an ASCII file that is saved in the directory you specified at logon.

Format

The format of the `list hostgroup` command is:

list hostgroup *`hostgroup name`*

where:

- *`hostgroup name`* is the name of the host group for which you want to see a list of IP addresses.

Explanation

Executing this command is equivalent to clicking a particular host group in the GUI. You can get a complete list of host groups available in your current group by omitting the *`hostgroup name`* argument. This is equivalent to clicking the Host Groups folder on the GUI. Note that this list will only be echoed to the console; no file will be written to the target directory if you omit the argument.

After you issue a `list hostgroup` command, you can open the file with any text editor, edit the entries, and issue a `save hostgroups` command. See "save hostgroup" (p. 2-87) later in this chapter for details on this command.

Example

```
lsmscmd list hostgroup bricks
```

The following is a typical example of output for this command:

```
LIST HOSTS:OK
```

Important! The list commands create a directory structure, within the directory that you specified in the lsmslogon command, that corresponds to the LSMS directory structure.

Thus, issuing the `list hostgroup` command will create the directory tree */System/Policy_Components/Host_Groups* in your target directory.

□

list lan2lantunnel

Overview

Retrives the specified LAN-to-LAN tunnel configuration to a file under the*<cli_dir>/<group>/VPN/Lan2Lan_Tunnels* folder.

Format

`list lan2lantunnel [lan2lanTunnelName]`

where *lan2lanTunnelName* is the name of the tunnel. If you leave off this argument, the names of all the LAN-to-LAN tunnels in this group will be displayed on the console.

Explanation

Use this command to list the LAN-to-LAN tunnel configuration of a particular tunnel to a file of the same name, or have it list all the tunnel names.

Example 1

`list lan2lantunnel`

The following is a typical example of output for this command:

```
LIST LAN2LAN TUNNEL: OK
1 '13.45.43.2_23.45.62.198'
2 'dhcp-paper@test02_87.3.58.153'
3 'pppoe1-v1zone_87.3.58.153['
```

Example 2

`list lan2lantunnel 13.45.43.2_23.45.62.198`

The following is a typical example of output for this command:

```
LIST LAN2LAN TUNNEL: OK
```

□

list lan2lantunneldefaults

Overview

The `list lan2lantunneldefaults` command retrieves the defualt LAN-to-LAN tunnel configuration to a file called ’defaults’ under the *<cli_dir>/<group>/VPN/Lan2Lan_Tunnels* folder.

Format

```
list lan2lantunneldefaults
```

Explanation

Use this command to view, perhaps with the intent to edit and save, the LAN-to-LAN tunnel defaults configuration.

Example

```
list lan2lantunneldefaults
```

The following is a typical example of output for this command:

```
LIST LAN2LAN TUNNEL DEFAULTS: OK
```

☐

list servicegroup

Overview

The `list servicegroup` command gets the protocol/srcport/destport tuple that belongs to the specified service group. If you supply the *`service group name`* argument, the output of the command is an ASCII file that is saved in the directory you specified at logon.

Format

The format of the `list servicegroup` command is:

`lsmscmd list servicegroup` *`service group name`*

where *`service group name`* is the name of the service group for which you want to see the protocol/srcport/destport information.

Explanation

Executing this command is equivalent to clicking a particular service group in the GUI. You can get a complete list of service groups available in your current group by omitting the *`service group name`* argument. This is equivalent to clicking the Service Groups folder on the GUI. Note that this list will only be echoed to the console; no file will be written to the target directory if you omit the argument.

After you issue a `list servicegroup` command, you can open the file with any text editor, edit the entries, and issue a `save hostgroups` command. See "save servicegroup" (p. 2-90) later in this chapter for details on this command.

Examples

```
lsmscmd list servicegroup bootpc
```

The following is a typical example of output for this command:

```
LIST SERVICES:OK
```

> **Important!** The list commands create a directory structure, within the directory that you specified in the lsmslogon command, that corresponds to the LSMS directory structure.
>
> Thus, issuing the `list servicegroup` command will create the directory tree */System/Policy_Components/Service_Groups* in your target directory.

□

list unusedclientteps

Overview

The `list unusedclientteps` command lists the tunnel endpoints (TEPs) of a given Brick that are not currently being used by any Client tunnel. The listing is displayed on the console.

Format

`lsmscmd list unusedclientteps` *`<brickName>`*

where `brickName` is the name of the Brick in the current group. This argument is required.

Explanation

This command helps the user know what TEPs are available for editing the localTep field of an existing Client tunnel or of a new Client tunnel.

Example

`lsmscmd list unusedclientteps v1brick`

The following is a typical example of output for this command:

```
LIST UNUSED CLIENT TEPS: OK
1 'pppoe1 s1zone'
```

The first field in each list entry is the Virtual Brick Address (VBA) and the second is the Brick ruleset.

□

logout

Overview

The `logout` command terminates an Administrator session of the command line interface.

Format

The format of the `logout` command is:

`lsmscmd logout [admin id]`

Explanation

The `logout` command is executed by an Administrator to terminate a command line interface session.

The `[admin id]` option allows you to log out another administrator. You must be an LSMS Administrator to be able to log out another administrator.

□

lsmslogon

Overview

The `lsmslogon` command enables an Administrator to log into the LSMS. The LSMS authenticates the login and determines the type of administrator and privileges, based on the administrator's Admin ID. Note that if you are logged in to the LSMS Navigator or another CLI with the same ID, the previous session will be terminated.

Format

The format of the `lsmslogon` command on *Windows* is:

`lsmslogon <admin_ID> <destination_directory> [-p <password_file> or -f <password>] [-t <logon_shell_timeout>]`

and on *Solaris* is:

`. lsmslogon <admin_ID> <destination_directory> [-p <password_file> or -f <password>] [-t <logon_shell_timeout>]`

where:

- *"." on Solaris, means the command refers to the current directory*
- *<admin_ID>* is the administrator's Admin ID.
- *<destination_directory>* is the directory in which the LSMS will store any zone assignment or policy files. This directory is created in the directory in which you installed the LSMS software. To specify a different directory, supply the complete path.
- `-p` *<password_file>* is the pathname of a file containing the administrator's password.
- `-f` *<password>* is the administrator's password.
- `-t` *<logon_shell_timeout>* is the timeout period for the command line interface (in seconds).

Explanation

The *<admin_ID>* is the ID the administrator uses to log into the LSMS graphical user interface (GUI). It is a required argument, and must be the first argument of the `lsmslogon` command.

The *<destination_directory> is also a required argument, and* must be the second argument of the `lsmslogon` command. It is the directory in which any zone assignment or policy files will be placed. You can enter a dot (.) to indicate the current directory, or the path to another directory. Note that if you enter only a directory name, without the complete path, the directory will be created relative to the current directory.

> **Important!** The administrator must have write permission for the destination directory, or the login will fail. If the destination directory does not exist, the software will create it and give the user read/write permission.

The administrator's password is also required. There are three ways to enter it:

- Enter the password directly with the `-f` parameter.
- Create a file containing the password and enter the pathname with the `-p` parameter.
- Do neither of the above, and the system will prompt you to enter the password from the keyboard.

By default, the command line interface has a timeout equal to the LSMS GUI timeout. If you wish to change the timeout period, you can enter a value in seconds with the `-t` parameter.

Example 1

`lsmslogon abc . -f abc123` (Windows)

`.lsmslogon abc . -f abc123` (Solaris)

In this command, the administrator's Admin ID is `abc`, the files will be stored in the current directory, and the administrator's password is `abc123`.

Example 2

`lsmslogon lsmsadm` *c:\users\files* (Windows)

`. lsmslogon lsmsadm` *c:\users\files* (Solaris)

In this command, the administrator's Admin ID is `lsmsadm` and the files will be stored in *c:\users\files*. Since no password was entered, the system will prompt the administrator to enter the password.

Example 3

`lsmslogon grpadm` *c:\users\files* `-p pwdfile -t 3000` (Windows)

`. lsmslogon grpadm` *c:\users\files* `-p pwdfile -t 3000` (Solaris)

In this command, the administrator's Admin ID is grpadm, the retrieved files will be stored in c:\users\files, and the administrator's password is found in the file pwdfile. In addition, the command line timeout has been set to 3000 seconds (50 minutes).

□

refreshmac brick

Overview

The `refreshmac brick` command is used to refresh the MAC table of a Brick.

Format

The format of the `refreshmac brick` command is:

```
refreshmac brick <brickname>
```

where:

- *<brickname>* is the name of the Brick for which you want to refresh the MAC table.

Explanation

The `refreshmac brick` command is only issued for Bricks which do not have the **Allow MAC Address To Move** checkbox on the Brick Editor Options tab checked.

Example

```
lsmscmd refreshmac brick_one
```

rehome brick

Overview

The rehome brick command is used in a redundant LSMS configuration to reassign management of a specific Brick to the other LSMS of the pair.

Format

The format of the rehome brick command is:

```
rehome brick brickname> <LsmsIpaddress>
```

where:

- *<brickname>* is the name of the Brick you want to rehome.
- *<LsmsIpaddress>* is the IP address of the LSMS to which you want to rehome this Brick.

Explanation

In a redundant LSMS configuration, each Brick is ″homed″ to one of the two available LSMSs. The LSMS that a Brick is homed to keeps the log records for that Brick.

Example

```
lsmscmd rehome brick brick101 10.10.10.10
```

```
CONTROL BRICKS: OK
```

This command rehomes brick101 to the LSMS with the IP address 10.10.10.10.

```
lsmscmd
rehome
brick
brick101 10.10.10.55
```

```
CONTROL BRICKS:Valid LSMS IP Addresses for brick 'brick101' are:
10.10.10.10
10.10.10.5
```

This example illustrates the response when an administrator attempts to rehome a Brick, in this case perhaps because of a typing error, to an invalid IP address. The LSMS responds with a list of valid LSMS IP addresses.

□

save brick

Overview

The `save brick` command saves a new brick configuration, or modifications to an existing brick configuration, in the LSMS database.

The `save brick` command is used in conjunction with the `list brick` command to make modifications to the settings of a Brick and save those modifications.

Format

The format of the `save brick` command is:

`lsmscmd save brick <filename>`

where:

- *`<filename>`* is the name of the Brick configuration file for the Brick being saved. Refer to the "add brick" (p. 2-6) command for a description of the Brick configuration file.

Explanation

Use the `save brick` command to save a new Brick or to save modifications to the settings of an existing Brick.

Executing this command is equivalent to selecting **Save** from the File menu of the Brick Editor.

Example

`lsmscmd save brick brick33`

This command saves a Brick named ″brick33″.

□

save brickruleset

Overview

The `save brickruleset` command saves the contents of the current Brick ruleset with the specified ruleset name.

The `save brickruleset` command is used in conjunction with the `list brickruleset` command to make modifications to a brick ruleset and save those modifications.

Format

The format of the `save brickruleset` command is:

`save brickruleset` *`<brick ruleset name>`*

where:

- *`<brick ruleset name>`* is the name of the ruleset you want to save.

Explanation

Use the `save brickruleset` command to save any changes to a Brick ruleset.

Example

```
lsmscmd save brickruleset sales
```

This command saves a Brick ruleset named *sales*.

□

save clientlicenselimits

Overview

Saves the current configuration configuration of client license limits on groups and TEPs.

Format

```
save clientlicenselimits
```

Example

save clientlicenselimits

□

save clienttunnel

Overview

Save the specified Client tunnel configuration back to the LSMS.

Format

`save clienttunnel <clientTunnelName>`

Explanation

When this command is run, a file called `<clientTunnelName>` containing a Client tunnel configuration must exist in the

<cli_dir>/<group>/VPN/Clinet_Tunnels folder.

Example

save clienttunnel 13.45.43.2

The following is a typical example of output for this command:

```
SAVE CLIENT TUNNEL: OK
```

save clienttunneldefaults

Objective

After editing the file called 'default' under the *<cli_dir>/<group>/VPN/Client_Tunnels* folder, this command saves the changes back to the LSMS.

Format

```
save clienttunneldefaults
```

Explanation

Use this command to save the Client tunnel defaults configuration.

Example

```
lsmscmd save clienttunneldefaults
```

The following is a typical example of output for this command:

```
SAVE LIENT TUNNEL DEFAULTS: OK
```

□

save dependencymasks

Overview

The `save dependencymasks` command saves the contents of the current dependency mask with the specified dependency mask name.

The `save dependencymasks` command is used in conjunction with the `list dependencymasks` command to make modifications to dependency masks and save those modifications.

Format

The format of the `save dependencymasks` command is:

save `dependencymasks` *<dependency mask name>*

where:

- *<dependency mask name>* is the name of the dependency mask you want to save.

Explanation

Use the `save dependencymasks` command to save any changes to a dependency mask.

Example

lsmscmd `save dependencymasks` client

This command saves a dependency mask named client.

□

save hostgroup

Overview

The `save hostgroup` command saves the contents of the current hostgroup with the specified host group name.

The `save hostgroup` command is used in conjunction with the `list hostgroup` command to make modifications to host groups and save those modifications.

Format

The format of the `save hostgroup` command is:

save hostgroup <*hostgroup name*>

where:

- <*hostgroup name*> is the name of the host group you want to save.

Explanation

Use the `save hostgroup` command to save any changes to a host group.

Example

```
lsmscmd save hostgroup marketing
```

This command saves a host group named marketing.

□

save lan2lantunnel

Overview

Save the specified LAN-to-LAN tunnel configuration back to the LSMS.

Format

```
save lan2lantunnel [<lan2lanTunnelName>]
```

where lan2lanTunnelName is the name of the tunnel.

Explanation

When this command is run, a file called *<lan2lanTunnelName>* containing a LAN-to-LAN tunnel configuration must exist in the *<cli_dir>/<group>/VPN/Lan2Lan_Tunnels* folder.

Example 1

```
save lan2lantunnel 13.45.43.2_23.45.62.198
```

The following is a typical example of output for this command:

```
SAVE LAN2LAN TUNNEL: OK
```

□

save lan2lantunneldefaults

Overview

After editing the file called *'default'* under the *<cli_dir>/<group>/VPN/Lan2Lan_Tunnels* folder, this command saves the changes back to the LSMS.

Format

```
save lan2lantunneldefaults
```

Explanation

Use this command to save the LAN-to-LAN tunnel defaults configuration.

Example

```
save lan2lantunneldefaults
```

The following is a typical example of output for this command:

```
SAVE LAN2LAN TUNNEL DEFAULTS: OK
```

save servicegroup

Overview

The `save servicegroup` command saves the contents of the current service group with the specified service group name.

The `save servicegroup` command is used in conjunction with the `list servicegroup` command to make modifications to service groups and save those modifications.

Format

The format of the `save servicegroup` command is:

`save servicegroup <servicegroup name>`

where:

- *<servicegroup name>* is the name of the service group you want to save.

Explanation

Use the `save servicegroup` command to save any changes to a service group.

Example

```
lsmscmd save servicegroup special
```

This command saves a service group named *special*.

□

3 LSMS CLI Files

Overview

Purpose

This chapter provides the information required to understand and edit, if necessary, the policy information in Lucent VPN Firewall *Brick®* device, Brick ruleset, host group, service group, dependency mask, client tunnel, and LAN-LAN tunnel files.

File commands

This chapter describes the files that are created when an Administrator executes the following commands:

- `list applicationfilter`
- `list brick`
- `list brickruleset`
- `list clienttunnel`
- `list clienttunneldefault`
- `list dependencymasks`
- `list hostgroup`
- `list lan2lantunnel`
- `list lan2lantunneldefaults`
- `list servicegroup`

These files are stored in the group directory that you specified when you logged in. The list commands create a directory structure, within that directory that corresponds to the LSMS directory structure.

Thus, issuing the `list brick` command will create the directory tree:

/System/Devices/Bricks

in your target directory.

You can edit these files using a standard text editor to add or update Brick ports, Brick rulesets, host groups, service groups, dependency masks, client tunnels, and LAN-LAN tunnels.

Contents

brick File	3-3
brickruleset File	3-12
client license limits File	3-20
client tunnel defaults File	3-24
client tunnel File	3-34
hostgroups File	3-49
lan2lan tunnel defaults File	3-51
lan2lan tunnel File	3-57
servicegroups File	3-72
dependency masks File	3-74

brick File

Overview

The brick file contains configuration information for a specific Brick.

The name of the file is <`brick name`> and is located on the LSMS in the directory *<cli dir>/<group>/Devices/Brick*, where <`cli dir`> is given as an argument to the `lsmslogon` command, and <`group`> is the current group that you are in when executing the `add brick` command. By default, the group is **system**.

Data in the Brick configuration file is organized in a <`name`>=<`value`> format. To specify table data, an index is added to the name to specify the row in the table to which it applies.

Only one name/value pair exists per line. The order of each name/value pair in the file can vary.

Explanation

Refer to the configuration file description in the "add brick" (p. 2-6) command section in Chapter 2, "LSMS CLI Commands" for a description of each field.

Example

The following is an example of a portion of a typical Brick file:

```
pppoe1ChapKey=
adminServerIP=192.168.10.27,20.20.20.20
loginBannerText=
pppoe1KeepAliveRetryCnt=5
pppoe2ChapKey=
failoverActvTime=4
firewallIP=50.50.50.50
pingMinActive=120
failoverYldTime=15
showVLANView=false
auditWait=false
activationThreshold=80
enableICM=true
multicastToFirstZone=false
failbackDelay=
dhcpAddresses=*
name=brick50
description=
gateway=50.50.50.1
pppoe1MACAddr=
pppoeAsRedundantPair=false
routeReturn=false
pppoe2MACAddr=
pppoe2Password=
remoteLoginID=
pppoe2KeepAliveIntvl=30
dhcpMethod=broadcast
timeOffsetFromSms=0.0
pppoe2KeepAliveRetryCnt=5
enableMsgsNoLogin=false
promonLogsIntervalSecs=30
pppoe1UserId=
brickType=brick
enableBrickFailover=false
macAddressB=
pppoe2UserId=
useBrickAddr=false
macAddressA=
mobile=false
encryptPreferredLink=true
dhcpServers=*
pppoe1Service=
pppoe2Service=
autoRefreshMac=false
pppoe1KeepAliveIntvl=30
primaryBrick=none
failoverPrfStshInt=auto
```

```
localPartition=*Default
version=9.0.199
failoverLabelB=
failoverLabelA=
dynNatDelaySecs=60
stickiness=300
pppoe1Password=
targetFloorUtilization=65
dhcpServerHostGroupName=
dhcpAddressHostGroupName=
VLANID[0]=1
VLANipAddress[0]=50.50.50.50/24
brickVLANIPCount=1
interfaceName[0]=local
aggregatePort[0]=
defaultVLANID[0]=1
receiveBitRate[0]=100M
transmitBitRate[0]=100M
enableQOS[0]=false
dhcpRequest[0]=false
interfaceMode[0]=auto
portDescription[0]=
enableJumboFrame[0]=false
mtu[0]=
ignoreHeartBeatFailures[0]=false
interfaceName[1]=ether0
aggregatePort[1]=
defaultVLANID[1]=1
receiveBitRate[1]=100M
transmitBitRate[1]=100M
enableQOS[1]=false
dhcpRequest[1]=false
interfaceMode[1]=auto
portDescription[1]=
enableJumboFrame[1]=false
mtu[1]=
ignoreHeartBeatFailures[1]=false
interfaceName[2]=ether1
aggregatePort[2]=
defaultVLANID[2]=1
receiveBitRate[2]=100M
transmitBitRate[2]=100M
enableQOS[2]=false
dhcpRequest[2]=false
interfaceMode[2]=auto
portDescription[2]=
enableJumboFrame[2]=false
mtu[2]=
ignoreHeartBeatFailures[2]=false
```

```
interfaceName[3]=ether2
aggregatePort[3]=
defaultVLANID[3]=1
receiveBitRate[3]=100M
transmitBitRate[3]=100M
enableQOS[3]=false
dhcpRequest[3]=false
interfaceMode[3]=auto
portDescription[3]=
enableJumboFrame[3]=false
mtu[3]=
ignoreHeartBeatFailures[3]=false
interfaceName[4]=ether3
aggregatePort[4]=
defaultVLANID[4]=1
receiveBitRate[4]=100M
transmitBitRate[4]=100M
enableQOS[4]=false
dhcpRequest[4]=false
interfaceMode[4]=auto
portDescription[4]=
enableJumboFrame[4]=false
mtu[4]=
ignoreHeartBeatFailures[4]=false
interfaceNumber[0]=0
policy[0]=firewall
virtualBrickAddress[0]=
dhcpTEPVBA[0]=false
matchVBAPackets[0]=false
zoneIPAddressOrRange[0]=*
zoneIPHost[0]=
allowedOutSourceIPAddressOrRange[0]=*
allowedOutSourceIPHost[0]=
localAddressmapping[0]=direct
localPresenceAddressOrRange[0]=
localPresenceHost[0]=
vpnCertificate[0]=
defaultAuthService[0]=
AuthTimeOut[0]=480
SourceIPs[0]=*
zonePriority[0]=16
maxQueueLatency[0]=500
qosParamsActive[0]=00
guarZoneRateIn[0]=
guarZoneRateOut[0]=
maxZoneRateIn[0]=
maxZoneRateOut[0]=
maxZoneConcSessTotal[0]=
maxZoneConcSessIn[0]=
```

```
maxZoneConcSessOut[0]=
setTOSDiffServBits[0]=false
separateBorrowSetting[0]=false
bitTemplate[0]=TOS
bitPatternBorrow[0]=00
bitPatternNonBorrow[0]=00
interfaceNumber[1]=1
policy[1]=
virtualBrickAddress[1]=
dhcpTEPVBA[1]=false
matchVBAPackets[1]=false
zoneIPAddressOrRange[1]=*
zoneIPHost[1]=
allowedOutSourceIPAddressOrRange[1]=*
allowedOutSourceIPHost[1]=
localAddressmapping[1]=direct
localPresenceAddressOrRange[1]=
localPresenceHost[1]=
vpnCertificate[1]=
defaultAuthService[1]=
AuthTimeOut[1]=480
SourceIPs[1]=*
zonePriority[1]=16
maxQueueLatency[1]=500
qosParamsActive[1]=00
guarZoneRateIn[1]=
guarZoneRateOut[1]=
maxZoneRateIn[1]=
maxZoneRateOut[1]=
maxZoneConcSessTotal[1]=
maxZoneConcSessIn[1]=
maxZoneConcSessOut[1]=
setTOSDiffServBits[1]=false
separateBorrowSetting[1]=false
bitTemplate[1]=TOS
bitPatternBorrow[1]=00
bitPatternNonBorrow[1]=00
interfaceNumber[2]=2
policy[2]=
virtualBrickAddress[2]=
dhcpTEPVBA[2]=false
matchVBAPackets[2]=false
zoneIPAddressOrRange[2]=*
zoneIPHost[2]=
allowedOutSourceIPAddressOrRange[2]=*
allowedOutSourceIPHost[2]=
localAddressmapping[2]=direct
localPresenceAddressOrRange[2]=
localPresenceHost[2]=
```

```
vpnCertificate[2]=
defaultAuthService[2]=
AuthTimeOut[2]=480
SourceIPs[2]=*
zonePriority[2]=16
maxQueueLatency[2]=500
qosParamsActive[2]=00
guarZoneRateIn[2]=
guarZoneRateOut[2]=
maxZoneRateIn[2]=
maxZoneRateOut[2]=
maxZoneConcSessTotal[2]=
maxZoneConcSessIn[2]=
maxZoneConcSessOut[2]=
setTOSDiffServBits[2]=false
separateBorrowSetting[2]=false
bitTemplate[2]=TOS
bitPatternBorrow[2]=00
bitPatternNonBorrow[2]=00
interfaceNumber[3]=3
policy[3]=
virtualBrickAddress[3]=
dhcpTEPVBA[3]=false
matchVBAPackets[3]=false
zoneIPAddressOrRange[3]=*
zoneIPHost[3]=
allowedOutSourceIPAddressOrRange[3]=*
allowedOutSourceIPHost[3]=
localAddressmapping[3]=direct
localPresenceAddressOrRange[3]=
localPresenceHost[3]=
vpnCertificate[3]=
defaultAuthService[3]=
AuthTimeOut[3]=480
SourceIPs[3]=*
zonePriority[3]=16
maxQueueLatency[3]=500
qosParamsActive[3]=00
guarZoneRateIn[3]=
guarZoneRateOut[3]=
maxZoneRateIn[3]=
maxZoneRateOut[3]=
maxZoneConcSessTotal[3]=
maxZoneConcSessIn[3]=
maxZoneConcSessOut[3]=
setTOSDiffServBits[3]=false
separateBorrowSetting[3]=false
bitTemplate[3]=TOS
bitPatternBorrow[3]=00
```

```
bitPatternNonBorrow[3]=00
interfaceNumber[4]=4
policy[4]=vpnzone50
virtualBrickAddress[4]=50.50.50.253
dhcpTEPVBA[4]=false
matchVBAPackets[4]=false
zoneIPAddressOrRange[4]=*
zoneIPHost[4]=
allowedOutSourceIPAddressOrRange[4]=*
allowedOutSourceIPHost[4]=
localAddressmapping[4]=direct
localPresenceAddressOrRange[4]=
localPresenceHost[4]=
vpnCertificate[4]=
defaultAuthService[4]=
AuthTimeOut[4]=480
SourceIPs[4]=*
zonePriority[4]=16
maxQueueLatency[4]=500
qosParamsActive[4]=0
guarZoneRateIn[4]=
guarZoneRateOut[4]=
maxZoneRateIn[4]=
maxZoneRateOut[4]=
maxZoneConcSessTotal[4]=
maxZoneConcSessIn[4]=
maxZoneConcSessOut[4]=
setTOSDiffServBits[4]=false
separateBorrowSetting[4]=false
bitTemplate[4]=TOS
bitPatternBorrow[4]=00
bitPatternNonBorrow[4]=00
zoneInterfaceCount=5
zone[0]=vpnzone50
service[0]=6/443/*
proxyDescription[0]=Automatic entry for user authentication
proxyIP[0]=@ManageServer
proxyPort[0]=9011
encrypt[0]=false
thekey[0]=
reflectionType[0]=single
passNoLPA[0]=false
proxyCount=1
destinationNetwork[0]=30.30.30.30/24
gatewayIP[0]=50.50.50.10
routeDisabled[0]=false
routeDescription[0]=NJ gateway
verifyRoute[0]=true
routeCost[0]=0
```

```
routePingDestAddr[0]=30.30.30.101
routePingSrcAddr[0]=50.50.50.253
routePingInterval[0]=10
routePingTimeout[0]=1
routePingMaxFail[0]=3
destinationNetwork[1]=30.30.30.30/24
gatewayIP[1]=50.50.50.75
routeDisabled[1]=false
routeDescription[1]=PA gateway
verifyRoute[1]=true
routeCost[1]=10
routePingDestAddr[1]=30.30.30.101
routePingSrcAddr[1]=50.50.50.253
routePingInterval[1]=10
routePingTimeout[1]=1
routePingMaxFail[1]=3
routeCount=2
icmDescription[0]=Drop and No Audit
icmName[0]=Drop_Unaud
icmService[0]=*
icmThreshold[0]=0
icmAudit[0]=no
icmDrop[0]=yes
icmHalfOpen[0]=any
icmDescription[1]=Drop and Audit
icmName[1]=Drop_Audit
icmService[1]=*
icmThreshold[1]=0
icmAudit[1]=yes
icmDrop[1]=yes
icmHalfOpen[1]=any
icmDescription[2]=ICMP
icmName[2]=ICMP
icmService[2]=icmp
icmThreshold[2]=15
icmAudit[2]=any
icmDrop[2]=no
icmHalfOpen[2]=any
icmDescription[3]=UDP
icmName[3]=UDP
icmService[3]=udp
icmThreshold[3]=25
icmAudit[3]=any
icmDrop[3]=no
icmHalfOpen[3]=any
icmDescription[4]=TCP Half-Open
icmName[4]=TCP_SYN
icmService[4]=tcp
icmThreshold[4]=45
```

```
icmAudit[4]=any
icmDrop[4]=no
icmHalfOpen[4]=yes
icmDescription[5]=TCP Full-Open
icmName[5]=TCP_Full
icmService[5]=tcp
icmThreshold[5]=100
icmAudit[5]=any
icmDrop[5]=no
icmHalfOpen[5]=no
icmDescription[6]=All other
icmName[6]=Any
icmService[6]=*
icmThreshold[6]=25
icmAudit[6]=any
icmDrop[6]=no
icmHalfOpen[6]=any
icmCount=7
pingFailoverCount=0
adminServerName=lsms-primary,lsms-secondary
adminServerType=source,target
adminServerAssocLSMS=,

certType=dss
skipRouteCheck=false
```

☐

brickruleset File

Overview

The rules file contains all the rules in a given Brick ruleset security policy.

The rules are presented in numerical order, beginning with the first rule. Each rule begins with a line that reads

```
***RULE #***
```

where # is the number of the rule. Each field in the rule occupies a separate line in the file under the rule number.

Format

The following shows the format of each rule in the Rules file:

```
*** RULE # ***
ruleNumber=
ruleDescription=
disabled=
sourceIP=
destinationIP=
service=
direction=
act=
dropAction=
sessionTimeout=
depMask=
auditSession=
alarmCode=
natSourceIP=
natSourceType=
natDestinationIP=
natDestinationType=
destinationPortMapping=
maxUseTotal=
maxUseConcurrent=
authorizeReturnChannel=
vpn=
synFloodType=
synFloodTimeout=
synFloodCount=
vlanID=
allowIcmpReplies=
```

Explanation

The following table describes each field in a rule:

Field	Explanation
`ruleNumber`	The number of the rule. It can be a number from 1-65529. The rule number is actually determined by the position of the rule in the file. If the position changes, and a "save dependencymasks" (p. 2-86) command is executed, the rule's number will change accordingly. Rule 0 should never be edited.
`ruleDescrtion`	A descrtion of the rule. This field corresponds to the **Descrtion** field in the Brick Zone Rule Editor, Basic tab.

Field	Explanation
`disabled`	A value of **True** disables the rule; a value of **False** enables the rule. The default is **True**. This field corresponds to the **Rule Active** field on the Brick Zone Rule Editor, Basic tab.
`sourceIP`	The IP address of the source host. It can be an asterisk (wildcard), a single IP address, a host group, or a user group. If you enter a host group, the host group must also be in the HostGroups file to be valid. If you enter a user group, you must prefix the name with a ~(e.g., ~group1). This field corresponds to the **Source** field on the Brick Zone Rule Editor, Basic tab.
`destinationIP`	The IP address of the destination host. It can be an asterisk (wildcard), a single IP address, a host group, a user group, or a Virtual Brick Address (VBA). If you enter a host group, the host group must also be in the HostGroups file to be valid. If you enter a user group, you must prefix the name with a ~(e.g., ~group1). This field corresponds to the **Destination** field on the Brick Zone Rule Editor, Basic tab.
`service`	The protocol, destination port and source port. It can be an asterisk (wildcard), a choice from the drop-down menu, or a service group name. This field corresponds to the **Service or Group** field on the Brick Zone Rule Editor, Basic tab
`direction`	The direction of the packet flow relative to the Brick zone ruleset. It can be In To Zone, Out Of Zone, or Both. This field corresponds to the **Direction** field on the Brick Zone Rule Editor, Basic tab.
`act`	The action to be taken if the source host, destination host and service in the rule match that of the packet. It can be Drop, Pass, Proxy, VPN, or VPN Proxy. Default is Drop. This field corresponds to the **Action** field on the Brick Zone Rule Editor, Basic tab.

Field	Explanation
`dropAction`	Refers to rules that have Drop as their action. It determines whether the Brick notifies the originator of a session that the rule has caused it to drop the session. It can be Yes or No. Default is No. This field corresponds to the **Drop Action** box on the Brick Zone Rule Editor, Basic tab.
`sessionTimeout`	The number of seconds of inactivity before an entry is removed from the session cache. It can be a number from 1-99,999. Default is 300. This field corresponds to the **Session Timeout** field on the Brick Zone Rule Editor, Advanced tab.
`depmask`	The name of a dependency mask. This field corresponds to the **Dependency Mask** field on the Brick Zone Rule Editor, Advanced tab.
`auditSession`	This field corresponds to the **Audit Session** field on the Brick Zone Rule Editor, Basic tab.
`alarmCode`	A code associated with an alarm. The alarm is triggered when a packet matching the rule arrives at the Brick. It can be a number from 1-65535, or a blank. This field corresponds to the **Alarm Code** field on the Brick Zone Rule Editor, Advanced tab.
`natSourceIP`	Used for network address translation. It is the address that the source host will be mapped to. It can be a single IP address, host group, VBA, or virtual_fw_addr. This field corresponds to the **Source Address Mapping** box on the Brick Zone Rule Editor, Address Translation tab.
`natSourceType`	Determines the type of source address mapping to be performed. It can be Direct, Pool or Local: • Direct causes the Brick to map source addresses on a one-to-one basis. It is the default. • Pool causes the Brick to map source addresses in a round robin fashion. • Local activates local presence feature which creates a pool of local addresses to use for client-LAN VPNs. (If the type is Local, the natSource field should be blank). This field corresponds to the Source Address Mapping **Type** field on the Brick Zone Rule Editor, Address Translation tab.

Field	Explanation
`natDestinationIP`	Used for network address translation. It is the address that the destination host will be mapped to. It can be a Virtual Brick address or a host group. This field corresponds to the **Destination Address Mapping** box on the Brick Zone Rule Editor, Address Translation tab.
`natDestinationType`	Determines the type of destination address mapping to be performed. It can be Direct, Pool or Local: • Directly causes the Brick to map destination addresses on a one-to-one basis. It is the default. • Pool causes the brick to map destination addresses in a round robin fashion. • Local activates the local presence feature which creates a pool of local addresses to use for client-LAN VPNs. (If the type is Local, the natDestination field should be blank). This field corresponds to the Destination Address Mapping **Type** field on the Brick Zone Rule Editor, Address Translation tab.
`destinationPortMapping`	The port(s) that will be used for destination port mapping. It can be a single port, a ranges of ports, or blank. This field corresponds to the **Destination Port Mapping** field on the Brick Zone Rule Editor, Address Translation tab.
`maxUseTotal`	Specifies the maximum number of times a rule can be invoked. The rule is disabled after the limit is reached. This field corresponds to the **Max Use Total** field on the Brick Zone Rule Editor, Advanced tab.
`maxUseConcurrent`	Specifies the maximum number of sessions authorized by the rule that can be active at one time. The rule becomes disabled when the limit is reached, and remains disabled until the count falls below the limit. This field corresponds to the **Max Use Concurrent** field on the Brick Zone Rule Editor, Advanced tab.
`authorizeReturnChannel`	Determines whether the initial packet of a session will create forward and reverse channels in cache with the same action, so that a separate rule is not needed to create a return channel. It can be True or False. Default is True. This field corresponds to the **Authorize Return Channel** checkbox on the Brick Zone Rule Editor, Advanced tab.

Field	Explanation
`vpn`	Determines whether packets authorized by this rule will be encrypted inside the Brick zone ruleset, outside the Brick zone ruleset, or both. It can be External, Internal, or Both. This field corresponds to the **Virtual Private Network** field on the Brick Zone Rule Editor, Advanced tab.
`synFloodType`	The type of SYN Flood protection for this rule. This field corresponds to the **SYN Flood Protection Type** field on the Brick Zone Rule Editor, Advanced tab.
`synFloodTimeout`	The timeout period after which the Brick sends a reset to the destination host. SYN Flood protection must be enabled on the rule for this timeout to be in effect. This field corresponds to the **SYN Flood Reset Timeout** field on the Brick Zone Rule Editor, Advanced tab.
`vlanID`	The VLAN identifier. The VLAN ID in the packets must match this VLAN ID for the rule to match a session. This field corresponds to the **VLAN ID** field on Brick Zone Rule Editor, Basic tab.
`allowIcmpReplies`	If enabled, allows ICMP messages containing the same 5-tuple — protocol, source address, destination address, source port (if any) and destination port (if any) — to be passed back to the other side of the Brick. This field corresponds to the **Allow ICMP Replies** checkbox on Brick Zone Rule Editor, Advanced tab.

Example

The following is an example of a fragment of a typical Brick ruleset file:

```
*** Brick Rule 1 ***
ruleNumber=1000
ruleDescription=
disabled=true
sourceIP=*
destinationIP=10.10.10.10
service=*
direction=out
act=pass
dropAction=
sessionTimeout=300
depMask=
auditSession=true
alarmCode=
natSourceIP=
natSourceType=pool
natDestinationIP=
natDestinationType=pool
destinationPortMapping=
maxUseTotal=
maxUseConcurrent=
authorizeReturnChannel=true
vpn=
synFloodType=none
synFloodTimeout=
synFloodCount=
vlanID=*
allowIcmpReplies=false
```

```
** Brick Rule 2 ***
ruleNumber=1001
ruleDescription=
disabled=true
sourceIP=10.10.10.10
destinationIP=*
service=brick_from_SMS_Services
direction=in
act=pass
dropAction=
sessionTimeout=300
depMask=
auditSession=true
alarmCode=
natSourceIP=
natSourceType=pool
natDestinationIP=
natDestinationType=pool
destinationPortMapping=
maxUseTotal=
maxUseConcurrent=
authorizeReturnChannel=true
vpn=
synFloodType=none
synFloodTimeout=
synFloodCount=
vlanID=*
allowIcmpReplies=false
```

□

client license limits File

Overview

Retrieves the client license limits configuration to a file.

Format

The following shows the format of the client license limits file.

```
licenseFromKey=
groupCount=
groupName[i]=
groupLimit[i]=
count=
device[i]=
tepIP[i]=
activeSessionCount[i]=
policy[i]=
enabled[i]=
name[i]=
licenseLimit[i]=
```

Explanation

The following table describes each field in the client license limits file:

Field	Explanation
`licenseFromKey`	This field shows the total number of client licenses on the LSMS based on the installed Feature Option keys.
`groupCount`	This field shows the number of Groups configured on the LSMS.
`groupName[i]`	i is in the range of 0...groupCount-1. This field contains the name of the Group.
`groupLimit[i]`	i is in the range of 0...groupCount-1. This field contains the number of client licenses that are allocated to this group. The sum of the licenses allocated to all the groups must be less than or equal to licenseFromKey.

Field	Explanation
`count`	This field shows the number of client tunnel endpoints in the current group.
`device[i]`	i is in the range of 0...count-1. This field contains the Brick name of the TEP.
`tepIP[i]`	i is in the range of 0...count-1. This field contains the IP address of the TEP.
`activeSessionCount[i]`	i is in the range of 0...count-1. This field contains the number of currently active client users connected to the TEP.
`policy[i]`	i is in the range of 0...count-1. This field contains the name of the zone assigned to the TEP.
`enabled[i]`	i is in the range of 0...count-1. This field is true if the TEP is enabled or false if it is disabled.
`name[i]`	i is in the range of 0...count-1. This field contains the name of the TEP.
`licenseLimit[i]`	i is in the range of 0...count-1. This field contains the number of client licenses allocated to this TEP. The sum of the licenses assigned to all the TEPs in a group must be less than or equal to groupLimit for that group.

Example

The following is an example of a typical client license limits file.

```
licenseFromKey=10100
groupName[0]=group1
groupLimit[0]=10
groupName[1]=group2
groupLimit[1]=0
groupName[2]=group3
groupLimit[2]=0
groupName[3]=group4
groupLimit[3]=0
groupName[4]=group5
groupLimit[4]=0
groupName[5]=system
groupLimit[5]=5000
groupCount=6
count=7
device[0]=brick12
tepIP[0]=12.12.12.103
activeSessionCount[0]=0
policy[0]=vpnzone12ca2
enabled[0]=true
name[0]=brick12ca3
licenseLimit[0]=10
device[1]=brick12
tepIP[1]=dhcp
activeSessionCount[1]=0
policy[1]=administrativezone
enabled[1]=true
name[1]=brick12dhcp
licenseLimit[1]=10
device[2]=brick12
tepIP[2]=12.12.12.253
activeSessionCount[2]=0
policy[2]=vpnzone12
enabled[2]=true
name[2]=brick12vpn12
licenseLimit[2]=10
device[3]=brick13
tepIP[3]=13.13.13.253
activeSessionCount[3]=0
policy[3]=vpnzone13
enabled[3]=true
name[3]=brick13vpn
licenseLimit[3]=0
device[4]=brick55
tepIP[4]=55.55.55.203
activeSessionCount[4]=0
policy[4]=vpnzone55
```

```
enabled[4]=true
name[4]=brick55vpn55
licenseLimit[4]=10
device[5]=model80
tepIP[5]=dhcp
activeSessionCount[5]=0
policy[5]=administrativezone
enabled[5]=true
name[5]=model80admin
licenseLimit[5]=100
device[6]=model80
tepIP[6]=10.10.10.251
activeSessionCount[6]=0
policy[6]=vpnzone
enabled[6]=true
name[6]=model80tep
licenseLimit[6]=1000
```

☐

client tunnel defaults File

Overview

Retrieves the default client tunnel configuration to a file.

Format

The following shows the format of the client tunnel defaults file:

```
heartBeatInterval=
idleTimeout=
userGroupID=
userGroupKeyword=
groupKey=
receiveAnyProposals=
primaryDNS=
primaryWINS=
secondaryDNS=
secondaryWINS=
clientFirewall=
useEnhancedFirewall=
disableFirewall=
allowPasswordSave=
dhGroup=
isakmpEncryptionType=
isakmpAuthType=
ipsecProtocol=
ipsecEncryptionType=
ipsecAuthType=
ipsecSaLifetimeKBytes=
ipsecSaLifetimeSec=
ikeV2ipsecSaLifetimeKBytes
ikeV2ipsecSaLifetimeSec
enablePerfectFwdSecrecy=
enableCompression=
phase1GroupID=
transportMethod=
udpEncapPorts=
encapType=
ikeV1AuthMethod=
ikeV2AuthMethod=
ikeV2PresharedKey=
ikeV2IDType=
ikeV2ID=
ikeV2CertIDType=
allowVpnCerts=
remoteUserIDField=
```

Explanation

The following table describes each field in the client tunnel defaults file:

Field	Explanation
`heartBeatInterval`	The Lucent IPSec Client is programmed to send out a keepalive/heartbeat message at regularly scheduled intervals to verify that it still has network connectivity to the other end of the tunnel. The Brick that receives the message generates a message back to the client. The default is 300 seconds. You can change this to any 1-second interval between 10 and 172800 seconds. Enter a value of zero (0) to disable keepalive/heartbeat messages. If the client does not receive after three heartbeat intervals (2 missed, plus one more), it disables the tunnel.
`idleTimeout`	A client tunnel will time out after a specified period of time if there is no activity in the tunnel in either direction. The default is 30 minutes. The valid range is 1-2880 minutes.
`userGroupID`	Field is not currently used.
`userGroupKeyword`	Field is not currently used.
`groupKey`	A default group key is generated randomly by the system. If you are not using digital certificates to authenticate client users, the key will be used by all IKEv1 clients users in the group when they set up a tunnel from their PCs to the tunnel endpoint. You can change the key, which must contain between 8 and 20 characters. Valid characters include a - z, A - Z, 0 - 9, and the special characters : ; | + ? ^{2} ( ) < > ^ % $ # &
`receiveAnyProposals`	If this field is set to false, the client parameter and policy settings must match the TEP settings exactly for the tunnel to come up.If this field is set to true, the parameter and policy settings configured for the TEP are the preferred settings, but the Brick accepts any combination of settings that the client proposes, provided the Brick supports those options.

Field	Explanation
primaryDNS primaryWINS secondaryDNS secondaryWINS	If your environment includes DNS and/or WINS servers, you can enter the the IP addresses of these servers in the appropriate fields. The default in each field is 0.0.0.0.
clientFirewall	This field determines whether packets to and from the client that are *not* going through the tunnel will be passed or dropped. There are three allowed values: • *pass* (allows the packets through the Brick) • *drop* (allows *no* packets through the Brick) • *client* (sessions started on the Client will be allowed out, but no *new* sessions allowed in; supported by v3.1 of the Client and above) If you leave the default asterisk in the **Hosts Behind Tunnel** field when setting up a client tunnel, all traffic will automatically go through the tunnel. In this case, it does not matter what you enter in this field. The value of this field automatically overrides the firewall setting in the Lucent IPSec Client, if the two settings are different.
useEnhancedFirewall	Field is not currently used. Set to false.
disableFirewall	Entering true in this field will disable the built-in firewall. This feature only works if using IPSec Client V6.0.1 or higher.
allowPasswordSave	This field is a convenience for the Lucent IPSec Client user. By default, this field is true. This means client users have the option of saving their password the first time they enable a tunnel, so that they do not have to enter it again each time they enable the tunnel. If the value is set to false, client users will not have this option and will have to enter their password each time they enable a tunnel.
dhGroup	This field sets the Diffie-Hellman Group. The choices are: Group 1, Group 2, Group 5, and Group 14. The default is Group 2.

Field	Explanation
`isakmpEncryptionType`	This field sets the encryption type for the IKE SA proposal. The options are TRIPLE DES CBC (default), DES CBC, AES CBC 128, AES CBC 192, and AES CBC 256.
`isakmpAuthType`	This field sets the authentication type for the IKE SA proposal. The default is HMAC SHA1, but you can change this to HMAC MD5.
`ipsecProtocol`	This field sets the protocol for the IPSec SA Proposal. The default is ESP-50, but this can be changed to AH-51. ESP-50 provides both encryption and authentication for every packet, while AH-51 only provides authentication.
`ipsecEncryptionType`	This field sets the encryption type for the IPSec SA Proposal. The options are TRIPLE DES CBC (default), DES CBC, AES CBC 128, AES CBC 192, and AES CBC 256.
`ipsecAuthType`	This field sets the authentication type for the IPSec SA Proposal. The default is HMAC SHA1, but you can change this to HMAC MD5.

`ipsecSaLifetimeKBytes`
`ipsecSaLifetimeSec`
`ikeV2ipsecSaLifetimeKBytes`
`ikeV2ipsecSaLifetimeSec`
	The Security Association has a lifetime specified in kilobytes and seconds (set for IKEv1 and IKEv2 client TEPs, respectively). The defaults are 14400 seconds (4 hours) and 5,000,000 kilobytes (approximately T1 speed for 8 hours). You can change the lifetime in seconds to any value between 120 -157,680,000 seconds (between 2 minutes and 5 years). You can change the lifetime in kilobytes to any value between 1000 - 10,000,000 kilobytes.
	You can set the value of this field to 0 to disable SA expiration.
	The Security Association will expire after the *first* of the above two lifetimes is reached. The session will then have to re-key.

`enablePerfectFwdSecrecy`	By default, this value is false, If it is true, a second Diffie-Hellman key exchange will take place during processing. This can improve security, but it also can impact re-keying performance.

Field	Explanation
`enableCompression`	The compression feature only applies when the tunnel endpoint is a Brick or another device that supports LZS compression. If the device is a Brick, it must be equipped with an encryption accelerator card (either a Model 201, 300, 350, 500, 1000 or 1100). For Bricks with an encryption card, this feature should be enabled. Allowed values for this field are true and false. This feature is supported in v3.1 and above of the Lucent IPSec Client.
`phase1GroupID`	The phase1GroupID is only used by non-Lucent IP. Sec Client programs. It is used in conjunction with Group Key (see below) for the first phase of the IKE negotiation process. If you are using the Lucent IPSec Client, you can ignore this field. The default is *gatewaygroupID*. If you are using a non-Lucent client, you can keep the default or change it. Whatever value you assign as the Group ID, you have to use it when configuring the non-Lucent client.

Field	Explanation
transportMethod udpEncapPorts encapType	These 3 fields define the allowed IPSec transport methods. Values for transportMethod are: pureipsec, udpencap, or both. Set to pureipsec for Pure IPSec (IP type 50/51), set to udpencap for IKEv1 UDP Encapsulation or IKEv2 NAT Traversal, and set to both if both Pure IPSec and UDP Encapsulation (IKEv1 and/or IKEv2) are allowed. Values for encapType are: lucent, natt, or both. Set to lucent for Pure IPSec or IKEv1 UDP Encapsulation, set to natt for IKEv2 NAT Traversal, and set to both if both IKEv1 UDP Encapsulation and IKEv2 NAT Traversal are allowed. The udpEncapPorts field contains a single port, comma separated list (no spaces), or a range of port numbers to be used for IKEv1 UDP Encapsulation. NAT Traversal automatically uses the standard port 4500 so it does not need to be included in the udpEncapPorts field.
ikeV1AuthMethod	This field specifies the IKEv1 Gateway authentication method. Set the value to key if the TEP should use the preshared key defined in the groupKey field for IKEv1 client authentications. Set the value to radius if the TEP should obtain the preshared key from RADIUS using the ″Per User Preshared Key (UMA)″ feature. If the value is set to radius, you must specify the IKEv1 Preshared Key RADIUS Attribute Code in the RADIUS response attributes for the RADIUS Authentication Service assigned to the TEP.
ikeV2AuthMethod	This field specified the IKEv2 Gateway authentication method. Set the value to key if the TEP should use the preshared key defined in the ikeV2PresharedKey field for IKEv2 client authentications. Set the value to cert if clients will use X.509 certificate authentication.

<table>
<tr><th>Field</th><th>Explanation</th></tr>
<tr><td><code>ikeV2PresharedKey</code></td><td>A default preshared key is generated randomly by the system. If you are not using digital certificates to authenticate client users, the key will be used by all IKEv2 clients users in the group when they set up a tunnel from their PCs to the tunnel endpoint. You can change the key, which must contain between 8 and 20 characters. Valid characters include a - z, A - Z, 0 - 9, and the special characters : ; | + ? ^{2} ( ) < > ^ % $ # &</td></tr>
<tr><td><code>ikeV2IDType</code></td><td>This field specifies the ID Type to be used by the Gateway for IKEv2 client negotiations when the ikeV2AuthMethod is key. The allowed values are IP Address, Email Address, and Domain Name.</td></tr>
<tr><td><code>ikeV2ID</code></td><td>This field contains the Gateway IKEv2 ID value of the type specified in the ikeV2IDType field. This field only needs to be populated if the ikeV2AuthMethod is key. For certificate authentications, the ID value is automatically taken from the certificate. The special value ″Virtual Brick Address″ may be used when the ikeV2IDType is IP Address.</td></tr>
<tr><td><code>ikeV2CertIDType</code></td><td>This field specifies the ID Type to be used by the Gateway for IKEv2 client authentications when the ikeV2AuthMethod is cert. The allowed values are IP Address, Email Address, Domain Name, and Distinguished Name. For certificate authentication to work, the specified ID Type field must have a value in the certificate assigned to the TEP.</td></tr>
<tr><td><code>allowVpnCerts</code></td><td>Set this field to true to allow IKEv2 clients to authenticate using X.509 Certificates. A VPN Certificate must be assigned to the TEP if you set this field to true. You must also assign a VPN Authentication Service to the TEP to specify the attribute checking to be done on client certificates.</td></tr>
</table>

Field	Explanation
`remoteUserIDField`	This field defines the attributes from the client's VPN certificate that should be used as the userID during authentication.Set to email to use the Email Address attributes, or dn to use the Domain Name attribute.

Example

The following is an example of a typical client tunnel defaults file:

```
heartBeatInterval=300
idleTimeout=30
userGroupID=
userGroupKeyword=
groupKey=5qbEcthsZSrgXhbVXGny
receiveAnyProposals=true
primaryDNS=0.0.0.0
primaryWINS=0.0.0.0
secondaryDNS=0.0.0.0
secondaryWINS=0.0.0.0
clientFirewall=client
useEnhancedFirewall=false
disableFirewall=false
allowPasswordSave=true
dhGroup=Group 2
isakmpEncryptionType=TRIPLE DES CBC
isakmpAuthType=HMAC SHA1
ipsecProtocol=ESP-50
ipsecEncryptionType=TRIPLE DES CBC
ipsecAuthType=HMAC SHA1
ipsecSaLifetimeKBytes=5000000
ipsecSaLifetimeSec=14400
ikeV2ipsecSaLifetimeKBytes=5000000
ikeV2ipsecSaLifetimeSec=14400
enablePerfectFwdSecrecy=false
enableCompression=false
phase1GroupID=gatewaygroupid
transportMethod=pureipsec
udpEncapPorts=501
encapType=lucent
ikeV1AuthMethod=key
ikeV2AuthMethod=key
ikeV2PresharedKey=5qbEcthsZSrgXhbVXGny
ikeV2IDType=IP Address
ikeV2ID=Virtual Brick Address
ikeV2CertIDType=IP Address
allowVpnCerts=false
remoteUserIDField=email
```

□

client tunnel File

Overview

Retrieves the specified client tunnel configuration to a file.

Format

The following shows the format of the client tunnel file:

```
localTep
brick
authService
authTimeout
hostGroup
name
hostIPs
licenseLimitenabled
useDefaultParameters
heartBeatInterval
idleTimeout
groupKey
receiveAnyProposals
primaryDNS
primaryWINS
secondaryDNS
secondaryWINS
clientFirewall
disableFirewall
allowPasswordSave
useDefaultPolicy
dhGroup
isakmpEncryptionType
isakmpAuthType
ipsecProtocol
ipsecEncryptionType
ipsecAuthType
ipsecSaLifetimeKBytes
ipsecSaLifetimeSec
ikeV2ipsecSaLifetimeKBytes
ikeV2ipsecSaLifetimeSec
enablePerfectFwdSecrecy
enableCompression
phase1GroupID
transportMethod
udpEncapPorts
debugLevel
debugSourceIP
ikeV1Allowed
ikeV2Allowed
encapType
useDefaultIKEv1
useDefaultIKEv2
useDefaultRemoteClientID
ikeV1AuthMethod
ikeV2AuthMethod
ikeV2PresharedKey
```

```
ikeV2IDType
ikeV2ID
ikeV2CertIDType
allowVpnCerts
remoteUserIDField
enablePDGAccounting
defaultDMSAPN
pdgMCCMNC
pdgIntervalMins
pdgIntervalBytes
waitForAccountingStart
skipOverlapCheck
```

Explanation

The following table describes each field in the client tunnel file:

Field	Explanation
`localTep`	This field contains two pieces of information separated by a comma and a space: the Virtual Brick Address of the TEP and the name of the zone assigned to the TEP.
`brick`	Enter the name of the Brick containing the TEP.
`authService`	Enter the name of the Authentication Service used to authenticate client users that connect to the TEP.
`authTimeout`	Enter the amount of time (in minutes) that client users will be authenticated. When the authentication times out, users will be disconnected. Allowed values are 1 to 2628000 minutes (5 years).

Field	**Explanation**
`hostGroup`	This field specifies the hosts behind the tunnel endpoint. If you want all outbound traffic from the client to go through the tunnel, leave this field blank and set the hostIPs field to astsrisk (*). Otherwise, enter the name of a host group containing specific IP addresses. If you enter a host group, make sure the **clientFirewall** field is set properly. The **clientFirewall** field determines whether traffic to other hosts (traffic that does not go through the tunnel)will be passed or dropped by the Brick.
`name`	Enter a name for this TEP in this field.
`hostIPs`	If this field is set to *, all outbound traffic from the client will automatically go through the tunnel. Otherwise, a host group must be specified in the hostGroup field with specific IP addresses that should be routed through the tunnel.
`licenseLimit`	This field sets the maximum number of active client sessions that are allowed for this TEP.
`enabled`	Set this field to true to enable the TEP. If this field is set to false, clients will not be allowed to connect to this tunnel endpoint.
`useDefaultParameters`	Set this field to true if the fields on the Parameters tab for this TEP should be updated with the values from the group Client Defaults when the Client Defaults are updated.

Field	Explanation
`heartBeatInterval`	The Lucent IPSec Client is programmed to send out a heartbeat message at regularly scheduled intervals to verify that it still has network connectivity to the other end of the tunnel. The Brick receiving the message generates a message back to the client. If the client does not receive the message within the specified interval, it disable the tunnel. The default is 300 seconds. You can change this to any 1-second interval between 10 and 172800 seconds. Enter a value of zero (0) to disable keepalive/heartbeat messages. If the client does not receive after three heartbeat intervals (2 missed, plus one more), it disables the tunnel.
`idleTimeout`	A client tunnel will time out after a specified period of time if there is no activity in the tunnel in either direction. The default is 30 minutes. You can change this to any 1-minute period between 1 and 2880 minutes.
`groupKey`	A default group key is generated randomly by the system. If you are not using digital certificates to authenticate client users, the key will be used by all IKEv1 client users in the group when they set up a tunnel from their PCs to the tunnel endpoint. You can change the key, which must contain between 8 and 20 characters. Valid characters include a - z, A - Z, 0 - 9, and the special characters : ; | + ? ″ ( ) < > ^ % $ # &

`primaryDNS`
`primaryWINS`
`secondaryDNS`
`secondaryWINS`
	If your environment includes DNS and/or WINS servers, you can enter the the IP addresses of these servers in the appropriate fields. The default in each field is 0.0.0.0.

Field	Explanation
`clientFirewall`	This field determines whether packets to and from the client that are *not* going through the tunnel will be passed or dropped. There are three options: • *Pass All* (allows the packets through the Brick) • *Drop All* (allows **no** packets through the Brick) • *Pass if Client Initiated* (sessions started on the Client will be allowed out, but no **new** sessions allowed in; supported by v3.1 of the Client and above) The valid values for this field are: `pass`, `drop`, or `client`. If you leave the default asterisk in the **Hosts Behind Tunnel** field when setting up a client tunnel, all traffic will automatically go through the tunnel. In this case, it does not matter what you enter in this field. *Note:* The value you enter in this field automatically overrides the firewall setting in the Lucent IPSec Client, if the two differ.
`disableFirewall`	Entering true in this field will disable the built-in firewall. This feature only works if using IPSec Client V6.0.1 or higher. *Note:* Disabling the firewall also removes the ″Allow Multi-session Protocol″ feature from the client’s GUI.
`allowPasswordSave`	This field is a convenience for the Lucent IPSec Client user. By default, this field is true. This means client users have the option of saving their password the first time they enable a tunnel, so that they do not have to enter it again each time they enable the tunnel. If the value is set to false, client users will not have this option and will have to enter their password each time they enable a tunnel.

Field	Explanation
useDefaultPolicy	Set this field to true if the fields on the Policy tabfor this TEP should be updated with the values from the group Client Defaults when the Client Defaults are updated.
dhGroups	This field sets the Diffie-Hellman Group. The choices are: Group 1, Group 2, Group 5, and Group 14. The default is Group 2.
isakmpEncryptionType	This field sets the encryption type for the IKE SA proposal. The options are TRIPLE DES CBC (default), DES CBC, AES CBC 128, AES CBC 192, and AES CBC 256.
isakmpAuthType	This field sets the authentication type for the IKE SA proposal. The default is HMAC SHA1, but you can change this to HMAC MD5.
ipsecProtocol	This field sets the protocol for the IPSec SA Proposal. The default is ESP-50, but this can be changed to AH-51. ESP-50 provides both encryption and authentication for every packet, while AH-51 only provides authentication.
ipsecEncryptionType	This field sets the encryption type for the IPSec SA Proposal. The valid values are TRIPLE DES CBC (default), DES CBC, AES CBC 128, AES CBC 192, and AES CBC 256.
ipsecAuthType	This field sets the authentication type for the IPSec SA Proposal. The default is HMAC SHA1, but you can change this to HMAC MD5.

Field	Explanation

`ipsecSaLifetimeKBytes`
`ipsecSaLifetimeSec`
`ikeV2ipsecSaLifetimeKBytes`
`ikeV2ipsecSaLifetimeSec`
	The Security Association has a lifetime specified in kilobytes and seconds (for IKEv1 and IKEv2 client TEPs, respectively). The defaults are 14400 seconds (4 hours) and 5,000,000 kilobytes (approximately T1 speed for 8 hours). You can change the lifetime in seconds to any value between 120 -157,680,000 seconds (between 2 minutes and 5 years).
	You can change the lifetime in kilobytes to any value between 1000 - 10,000,000 kilobytes.
	You can set the value of this field to 0 to disable SA expiration.
	The Security Association will expire after the *first* of the above two lifetimes is reached. The session will then have to re-key.

`enablePerfectFwdSecrecy`	By default, this value is false, If it is true, a second Diffie-Hellman key exchange will take place during processing. This can improve security, but it also can impact re-keying performance.
`enableCompression`	The compression feature only applies when the tunnel endpoint is a Brick or another device that supports LZS compression. If the device is a Brick, it must be equipped with an encryption accelerator card (either a Model 201, 300, 350, 500, 1000 or 1100). For Bricks with an encryption card, this feature should be enabled. Allowed values for this field are true and false. This feature is supported in v3.1 and above of the Lucent IPSec Client.

Field	Explanation
phase1GroupID	The phase1GroupID is only used by non-Lucent IPSec Client programs. It is used in conjunction with Group Key (see above) for the first phase of the IKE negotiation process. If you are using the Lucent IPSec Client, you can ignore this field. The default is gatewaygroupID. If you are using a non-Lucent client, you can keep the default or change it. Whatever value you assign as the Group ID, you have to use it when configuring the non-Lucent client.
transportMethod udpEncapPorts encapType	These 3 fields define the allowed IPSec transport methods. Values for transportMethodare: pureipsec, udpencap, or both. Set to pureipsec for Pure IPSec (IP type 50/51), set to udpencap for IKEv1 UDP Encapsulation or IKEv2 NAT Traversal, and set to both if both Pure IPSec and UDP Encapsulation (IKEv1 and/or IKEv2) are allowed. Values for encapType are: lucent, natt, or both. Set to lucent for Pure IPSec or IKEv1 UDP Encapsulation, set to natt for IKEv2 NAT Traversal, and set to both if both IKEv1 UDP Encapsulation and IKEv2 NAT Traversal are allowed. The udpEncapPorts field contains a single port, comma separated list (no spaces), or a range of port numbers to be used for IKEv1 UDP Encapsulation. NAT Traversal automatically uses the standard port 4500 so it does not need to be included in the udpEncapPorts field.

Field	Explanation
debugLevel debugSourceIP	The debugLevel field controls the level of debugging for the TEP. When tunnel debugging is enabled, the Brick writes debug messages in the VPN log, identified by zone, tunnel endpoint name and remote TEP (client IP). The available choices are 0 (debugging off), 1, 2, and 3 (most verbose). Enter the IP address of the remote client machine in the debugSourceIP field.
ikeV1Allowed	Set to true to allow IKEv1 clients to authenticate to this TEP.
ikeV2Allowed	Set to true to allow IKEv2 clients to authenticate to this TEP.
useDefaultIKEv1	Set this field to true if the fields on the IKEv1 Gateway tab for this TEP should be updated with the values from the group Client Defaults when the Client Defaults are updated.
useDefaultIKEv2	Set this field to true if the fields on the IKEv2 Gateway tab for this TEP should be updated with the values from the group Client Defaults when the Client Defaults are updated.
useDefaultRemoteClientID	Set this field to true if the fields on the Remote Client ID tab for this TEP should be updated with the values from the group Client Defaults when the Client Defaults are updated.

Field	**Explanation**
`ikeV1AuthMethod`	This field specifies the IKEv1 Gateway authentication method. Set the value to key if the TEP should use the preshared key defined in the groupKey field for IKEv1 client authentications. Set the value to radius if the TEP should obtain the preshared key from RADIUS using the ″Per User Preshared Key (UMA)″ feature. If the value is set to radius, you must specify the IKEv1Preshared Key RADIUS Attribute Code in the RADIUS response attributes for the RADIUS Authentication Service assigned to the TEP.
`ikeV2AuthMethod`	This field specified the IKEv2 Gateway authentication method. Set the value to key if the TEP should use the preshared key defined in the ikeV2PresharedKey field for IKEv2 client authentications. Set the value to cert if clients will use X.509 certificate authentication.
`ikeV2PresharedKey`	A default preshared key is generated randomly by the system. If you are not using digital certificates to authenticate client users, the key will be used by all IKEv2 clients users in the group when they set up a tunnel from their PCs to the tunnel endpoint. You can change the key, which must contain between 8 and 20 characters. Valid characters include a - z, A - Z, 0 - 9, and the special characters : ; | + ? ^{2} ( ) < > ^ % $ # &
`ikeV2IDType`	This field specifies the ID Type to be used by the Gateway for IKEv2 client negotiations when the ikeV2AuthMethod is key, The allowed values are IP Address, Email Address, and Domain Name.

Field	Explanation
`ikeV2ID`	This field contains the Gateway’s IKEv2 ID value of the type specified in the ikeV2IDType field. This field only needs to be populated if the ikeV2AuthMethod is key. For certificate authentications, the ID value is automatically taken from the certificate. The special value ″Virtual Brick Address″ may be used when the ikeV2IDType is IP Address.
`ikeV2CertIDType`	This field specifies the ID Type to be used by the Gateway for IKEv2 client authentications when the ikeV2AuthMethod is cert. The allowed values are IP Address, Email Address, Domain Name, and Distinguished Name. For certificate authentication to work, the specified ID Type field must have a value in the certificate assigned to the TEP.
`allowVpnCerts`	Set this field to true to allow IKEv2 clients to authenticate using X.509 Certificates. A VPN Certificate must be assigned to the TEP if you set this field to true. You must also assign a VPN Authentication Service to the TEP to specify the attribute checking to be done on client certificates.
`remoteUserIDField`	This field defines the attributes from the client’s VPN certificate that should be used as the userID during authentication. Set to email to use the Email Address attributes, or dn to use the Domain Name attribute.
`enablePDGAccounting`	Set this field to true to enable the PDG Accounting feature. The default value is false.
`defaultDMSAPN`	This field contains the string that should be used as the Dual Mode Service Access Point Name if it is not included in the IKE authentication request sent by the Dual Mode Handset. The DMS-APN is used in the Called-Station-Id field in RADIUS authentication and accounting messages.

Field	Explanation
`pdgMCCMNC`	This field contains the Mobile Country Code and Mobile Network Code of the network the Brick belongs to. It is used in the 3GPP-GGSN-MCC-MNC field in RADIUS authentication and accounting messages.
`pdgIntervalMins`	The Brick device will send accounting updates to RADIUS at the interval configured in this field. The default value is 30.
`pdgIntervalBytes`	The Brick device will send accounting updates to RADIUS whenever the tunnel traffic volume (input+output bytes) reaches the configured value or a multiple of the value. The default is 0, which means the Brick device will not send accounting updates based on traffic volume.
`waitForAccountingStart`	This field is currently not used. The value should be true.
`skipOverlapCheck`	Set this field to true if you received error N7028, which indicates that the Hosts Behind Tunnel for this TEP overlap with hosts behind another tunnel, but you want to save this TEP anyway.

Example

The following is an example of a typical client tunnel file:

```
localTep=55.55.55.203, vpnzone55
brick=brick55
authService=NJRadius
authTimeout=480
hostGroup=
name=brick55vpn55
hostIPs=*
licenseLimit=10
enabled=true
useDefaultParameters=true
heartBeatInterval=300
idleTimeout=30
groupKey=key4ikev1
receiveAnyProposals=true
primaryDNS=0.0.0.0
primaryWINS=0.0.0.0
secondaryDNS=0.0.0.0
secondaryWINS=0.0.0.0
clientFirewall=client
disableFirewall=false
allowPasswordSave=true
useDefaultPolicy=true
dhGroup=Group 2
isakmpEncryptionType=TRIPLE DES CBC
isakmpAuthType=HMAC SHA1
ipsecProtocol=ESP-50
ipsecEncryptionType=TRIPLE DES CBC
ipsecAuthType=HMAC SHA1
ipsecSaLifetimeKBytes=5000000
ipsecSaLifetimeSec=14400
ikeV2ipsecSaLifetimeKBytes=5000000
ikeV2ipsecSaLifetimeSec=14400
enablePerfectFwdSecrecy=false
enableCompression=false
phase1GroupID=gatewaygroupid
transportMethod=both
udpEncapPorts=501
debugLevel=0
debugSourceIP=
ikeV1Allowed=true
ikeV2Allowed=true
encapType=natt
useDefaultIKEv1=false
useDefaultIKEv2=false
useDefaultRemoteClientID=true
ikeV1AuthMethod=key
ikeV2AuthMethod=key
```

```
ikeV2PresharedKey=12345678
ikeV2IDType=IP Address
ikeV2ID=Virtual Brick Address
ikeV2CertIDType=IP Address
allowVpnCerts=false
remoteUserIDField=email
enablePDGAccounting=true
defaultDMSAPN=njpdg.lucent.com
pdgMCCMNC=310150
pdgIntervalMins=30
pdgIntervalBytes=0
waitForAccountingStart=true
skipOverlapCheck=false
```

□

hostgroups File

Overview

The HostGroups file contains the fields for a host group in a given group's security policy.

Each file begins with a line that reads

```
***HOST GROUP***
```

Format

The following shows the format of the HostGroups file:

```
*** HOST GROUP***
ipAddressOrRange=
```

descrtion=

Explanation

The following table describes each field in a host group:

Field	Explanation
`Address`	The IP addresses in the host group. It can be a single IP address, a list of IP addresses, or a range of IP addresses. This field corresponds to the **Host Addresses** field on the Host Group Editor.
`description`	Any remarks the administrator chooses to enter. They can contain from 1-80 characters. This field corresponds to the **Descrtion** field on the Host Group Editor.

Example

The following is an example of a typical HostGroups file:

```
*** HOST GROUP ***
ipAddressOrRange=12.23.34.45
description=for marketing group
```

□

lan2lan tunnel defaults File

Overview

Retrieves the default LAN-to-LAN tunnel configuration to a file.

Format

The following shows the format of the lan2lan tunnel defaults file:

```
preSharedkey=
startDate=
endDate=
sendAllProposals=
receiveAnyProposals=
customerName=
email=
phone=
comment=
ipsecProtocol=
ipsecEncryptionType=
ipsecAuthType=
ipsecSaLifetimeSec=
ipsecSaLifetimeKBytes=
enablePerfectFwdSecrecy=
enableCompression=
dhGroup=
isakmpEncryptionType=
isakmpAuthType=
isakmpSaLifetimeSec=
transportMethod=
udpEncapPorts=
heartBeatInterval=
probeInterval=
probesPerReport=
roundTripThreshold=
pktLostWaitTime=
```

Explanation

The following table describes each field in the lan2lan tunnel defaults file:

Field	Explanation
`preSharedkey`	A key is automatically generated by the LSMS. You can change this key if you wish, but it must contain 8 to 20 characters. This key is used by both tunnel endpoints, if an IKE negotiation is needed. This can happen if two Bricks are managed by different LSMS or if a Brick is setting up a tunnel with another vendor's device. If you select this option, you have to obtain and install a certificate on the workstation running the application. See the chapter on *Digital Certificates* in the *LSMS Policy Guide*.

`startDate`
and
`enDate`
	These dates determine the time period during which this tunnel is operable.
	The default is a 99-year time period, beginning with the current date. You can change this.

`receiveAnyProposals`
and
`sendAllProposals`
	These two values are true by default. The purpose is to enhance interoperability.
	When these values are true, the ISAKMP and IPSec parameters will be negotiated at a possibly lower security level than you specified to allow devices configured differently to still serve as one tunnel endpoint.
	However, if you want to ensure that only devices sharing your ISAKMP and IPSec parameters can serve as a tunnel endpoint, set either or both values to false.

`customerName`
`email`
`phone`
`comment`
	If one of the endpoints is administered by another individual, you can endter information about that person in these fields.
	This data is purely informational and not used in tunnel negotiations.

Field	Explanation
`ipsecProtocol`	This field sets the protocol for the IPSec SA Proposal. The default is ESP-50, but this can be changed to AH-51. ESP-50 provides both encryption and authentication for every packet, while AH-51 only provides authentication.
`ipsecEncryptionType`	This field sets the encryption type for the IPSec SA Proposal. The options are TRIPLE DES CBC (default), DES CBC, AES CBC 128, AES CBC 192, and AES CBC 256.
`ipsecAuthType`	This field sets the authentication type for the IPSec SA Proposal. The default is HMAC SHA1, but you can change this to HMAC MD5.
`ipsecSaLifetimeSec`	This Security Association has a lifetime specified in seconds. The default is 14,400 seconds (6 hours). You can change the lifetime in seconds to any value between 120 and 172,860 seconds (between 2 minutes and 48 hours).
`secSaLifetimeKBytes`	This Security Association has a lifetime specified in kilobytes. The default is 10,000,000 kilobytes. To change this, enter a number between 61000 and 10,000,000 kilobytes. *NOTE:* The Security Association will expire after the *first* of two lifetimes is reached. However, no session will be permitted to timeout before one minute, even if one of the above two lifetimes is reached first.
`enablePerfectFwdSecrecy`	By default, this value is false. If it is true, a new Diffie-Hellman key exchange will take place at every rekey interval, thereby increasing rekey time and the load on the LSMS. This can improve security, but it can also affect performance.

Field	Explanation
`enableCompression`	This field applies only to LAN-LAN tunnels between a Brick with an encryption accelerator card and any device that supports LZS compression. By default, it is false. If it is true, traffic through the tunnel will be compressed. The advantage of compression is that it means less data has to be sent over the wires, which may help conserve bandwidth and speed up transmission. The disadvantage is that it requires extra processing (to compress and decompress) on either end of the tunnel — which has performance implications at the tunnel endpoints.
`dhGroup`	The default is Diffie-Hellman Group 1, but you can change this to Group 2. Group 2 provides more security than the Group 1, but rekey time may be better using Group 1.
`isakmpEncryptionType`	This field sets the encryption type for the IKE SA proposal. The options are TRIPLE DES CBC (default), DES CBC, AES CBC 128, AES CBC 192, and AES CBC 256.
`isakmpAuthType`	This field sets the authentication type for the IKE SA proposal. The default is HMAC SHA1, but you can change this to HMAC MD5.
`isakmpSaLifetimeSec`	This Security Association has a lifetime specified in seconds. The default is 86400 seconds (24 hours). You can change the lifetime in seconds to any value between 120 and 172, 860 seconds (between 2 minutes and 48 hours).
`transportMethod`	This field determines the transport method. The default is pure IPSec (IP type 50/51), but you can change this to UDP Encapsulated (IP type 17).
`udpEncapPorts`	If the transport method is set to UDP Encapsulated, the default port is set to port 501. You can change this port.

Field	Explanation
`heartBeatInterval`	The Brick will send heartbeat messages through the tunnel at regular intervals. This field allows you to set the interval (in seconds). The default is 30 seconds. Enter a value of zero (0) to disable keepalive/heartbeat messages. The heart beat is used to determine the status of the tunnel that is displayed in the LAN-LAN Tunnel Viewer and the Status Overview portion of the Status Monitor.
`probeInterval`	The default is a 30 second probe interval.
`probesPerReport`	The default is 10 probes per report.
`roundTrThreshold`	The default round tr delay threshold is 1000 miliseconds.
`pktLostWaitTime`	The default packet lost timeout period is 3000 miliseconds.

Example

The following is an example of a typical lan2lan tunnel defaults file:

```
preSharedkey=FyaffaTfxgbpyWqzTycx
startDate=2002-06-26 00:00:00.0
endDate=2101-06-26 00:00:00.0
sendAllProposals=false
receiveAnyProposals=false
customerName=
email=
phone=
comment=
ipsecProtocol=ESP-50
ipsecEncryptionType=TRIPLE DES CBC
ipsecAuthType=HMAC SHA1
ipsecSaLifetimeSec=14400
ipsecSaLifetimeKBytes=10000000
enablePerfectFwdSecrecy=false
enableCompression=false
dhGroup=Group 1
isakmpEncryptionType=TRIPLE DES CBC
isakmpAuthType=HAMC SHA1
isakmpSaLifetimeSec=86400
transportMethod=pureipsec
udpEncapPorts=501
heartBeatInterval=39
probeInterval=30
probesPerReport=19
roundTripThreshold=1000
pktLostWaitTime=3000
```

☐

lan2lan tunnel File

Overview

Retrieves the specified LAN-to-LAN tunnel configuration to a file.

Format

The following shows the format of the lan2lan tunnel file:

```
localTep=
remoteTep=
e1Brick=
e1Group=
e2Brick=
e2Group=
localHostGroup=
remoteHostGroup=
endpointType=
useDefaultSLA=
heartBeatInterval=
description=
e12e2ah_AuthKey=
ipsecAuthType=
ipsecProtocol=
e22e1esp_AuthKey=
unmanagedIP=
probeInterval=
email=
e12e2ah_spi=
isakmpAuthType=
useDefaultPolicy=
e12e2manualProtocol=
enableCompression=
preSharedkey=
comment=
remotePhase1ID=
vpnType=
localPhase1ID=
remoteVpnCA=
e12e2esp_AuthType=
e22e1esp_EncryptionKey=
e22e1ah_AuthKey=
receiveAnyProposals=
probesPerReport=
initiator=
customerName=
ipsecSaLifetimeKBytes=
unmanagedDeviceName=
e2PresharedKey=
e2IDType=
e22e1esp_spi=
remoteHostIPs=
e1AuthMethod=
e12e2esp_spi=
useDefaultParameters=
roundTripThreshold=
```

```
name=
e12e2ah_AuthType=
e12e2esp_EncryptionType=
endDate=
e1IDType=
isakmpEncryptionType=
isakmpSaLifetimeSec=
dstSLA=
enableSLA=
enableTunnel=
enablePerfectFwdSecrecy=
e22e1esp_EncryptionType=
e12e2esp_AuthKey=
encapType=
phone=
udpEncapPorts=
remoteDistinguishedName=
debugLevel=
srcSLA=
e22e1esp_AuthType=
ipsecSaLifetimeSec=
e22e1ah_spi=
e22e1ah_AuthType=
e2AuthMethod=
localHostIPs=
e22e1manualProtocol=
transportMethod=
pktLostWaitTime=
sendAllProposals=
e12e2esp_EncryptionKey=
ipsecEncryptionType=
dhGroup=
authByDistinguishedName=
startDate=
skipOverlapCheck=
```

Explanation

The following table describes each field in the lan2lan tunnel file:

Field	Explanation
`enableTunnel`	Set this field to true to enable the tunnel.
`name`	Enter a name for the tunnel in this field.
`description`	Enter a description for the tunnel in this field.

Field	Explanation
`vpnType`	Set this field to auto for IKEv1 tunnels, autoV2 for IKEv2 tunnels, and manual for Manual Key tunnels.
`initiator`	Set this field to true when Endpoint 2 is the initiator. If Endpoint 1 is the initiator, set the field to false.
`debugLevel`	The debugLevel field controls the level of debugging for the tunnel. When tunnel debugging is enabled, the Brick writes debug messages in the VPN log, identified by zone, tunnel endpoint name and remote TEP. The available choices are 0 (debugging off), 1, 2, and 3 (most verbose).
`localTep`	This field contains two pieces of information for Endpoint 1, separated by a comma and a space: the Virtual Brick Address of the TEP and the name of the zone assigned to the TEP.
`remoteTep`	This field contains two pieces of information for Endpoint 2, separated by a comma and a space: the Virtual Brick Address of the TEP and the name of the zone assigned to the TEP.
`e1Brick`	This field contains the name of the Endpoint 1 Brick.
`e1Group`	This field contains the name of the Group of the Endpoint 1 Brick.
`e2Brick`	This field contains the name of the Endpoint 2 Brick. If Endpoint 2 is not a Brick, this field is blank.
`e2Group`	This field contains the name of the Group of the Endpoint 2 Brick. If Endpoint 2 is not a Brick, this field is blank.
`localHostIPs` `localHostGroup`	These fields are used to specify the host IP addresses behind Endpoint 1. Use localHostIPs to specify asterisk (*) for all hosts, a single IP address, a range or an address with subnet mask. For a list of IP Addresses, ranges, or addresses with subnet masks, create a host group and specify the name of the host group in the localHostGroup field. Only fill one of these fields and leave the other blank.

Field	Explanation
`remoteHostIPs` `remoteHostGroup`	These fields are used to specify the host IP addresses behind Endpoint 2. Use remoteHostIPs to specify asterisk (*) for all hosts, a single IP address, a range or an address with subnet mask. For a list of IP addresses, ranges, or addresses with subnet masks, create a host group and specify the name of the host group in the remoteHostGroup field. Only fill one of these fields and leave the other blank.
`endpointType`	This field specifies the Endpoint 2 type. Allowed values are: device (endpoint is a Brick), ip (endpoint is not a Brick or is a Brick that is managed by another LSMS), or other (endpoint is mobile and will be identified By Name rather than By IP Address).
`unmanagedIP`	If endpointType is ip, this field contains the IP Address of Endpoint 2, otherwise this field is blank.
`unmanagedDeviceName`	If endpointType is other, or if endpointType is device and it is a mobile TEP (dhcp, pppoe1, or pppoe2), this field contains the IKE ID of Endpoint 2; otherwise, this field is blank.
`localPhase1ID`	If preshared key authentication is used, this field contains the IKE ID for Endpoint 1; otherwise, it contains the word `certificate`.
`remotePhase1ID`	If preshared key authentication is used, this field contains the IKE ID for Endpoint 2. If X.509 certificate authentication is used, and Endpoint 2 is a Brick, this field contains the word `certificate`; otherwise, it contains the IKE ID of Endpoint 2.
`e1AuthMethod` `e2AuthMethod`	These fields are set to key for preshared key authentication or cert for X.509 certificate authentication. Both fields must be set to the same value.
`e1IDType` `e2IDType`	These fields set the IKE ID Type for Endpoint 1 and Endpoint 2. For preshared key authentication the allowed values are: IP Address, Domain Name, or Email Address (IKEv2 only). For IKEv1 mobile endpoints or if endpointType is other, the ID Type must be Domain Name. For X.509 certificate authentication the allowed values are: Distinguished Name, IP Address, Domain Name (IKEv2 only), or Email Address (IKEv2 only).

Field	Explanation
`preSharedkey`	This field contains the key for IKEv1 preshared key authentication. The key must contain between 8 and 20 characters. Valid characters include a - z, A - Z, 0 - 9, and the special characters : ; | + ? ^{2} ( ) < > ^ % $ # &
`e2PresharedKey`	This field contains the key for IKEv2 preshared key authentication. The key must contain between 8 and 20 characters. Valid characters include a - z, A - Z, 0 - 9, and the special characters : ; | + ? ^{2} ( ) < > ^ % $ # &
`remoteDistinguishedName`	This field contains the Endpoint 2 Distinguished Name for X.509 certificate authentication. It is a semi-colon delimited field using the following format: CN=commonName;O=orgName;OU=orgUnit;L=locality;ST=state;C=country
`authByDistinguishedName`	This field is set to true if X.509 certificate authentication is being used and the ID Type is Distinguished Name.
`remoteVpnCA`	This field contains the Distinguished Name of the CA Certificate used by Endpoint 2 for X.509 certificate authentication.
`useDefaultParameters`	Set this field to true if the fields on the Parameters tab for this tunnel should be updated with the values from the group LAN-LAN Defaults when the LAN-LAN Defaults are updated.

`customerName`
`email`
`phone`
`comment`
	If one of the endpoints is administered by another individual, you can enter information about that person in these fields. This data is purely informational and not used in tunnel negotiations.

`startDate` `endDate`	These dates determine the time period during which this tunnel is operable. The default is a 99-year time period, beginning with the current date. You can change these dates.

Field	Explanation
`heartBeatInterval`	The Brick sends heartbeat/keepalive messages through the tunnel at regular intervals. This field allows you to set the interval (in seconds). The default is 30 seconds. Enter a value of zero (0) to disable keepalive/heartbeat messages. The heartbeat is used to determine the status of the tunnel that is displayed in the LAN-LAN Tunnel Viewer and the Status Overview portion of the Status Monitor.
`receiveAnyProposals` `sendAllProposals`	The purpose of these fields is to enhance interoperability. When they are set to true, the IKE SA parameters and IPSec parameters will be negotiated at a possibly lower security level than you specified, to allow devices configured differently to still serve as one tunnel endpoint. However, if you want to ensure that only devices sharing your IKE SA parameters and IPSec parameters can serve as a tunnel endpoint, set either or both of these fields to false.

`transportMethod`
`udpEncapPorts`
`encapType`
	These 3 fields define the tunnel IPSec transport method. Values for transportMethod are: pureipsec or udpencap. Set to pureipsec for Pure IPSec (IP type 50/51), set to udpencap for IKEv1 UDP Encapsulation or IKEv2 NAT Traversal. Values for encapType are: lucent or natt. Set to lucent for Pure IPSec or IKEv1 UDP Encapsulation, set to natt for IKEv2 NAT Traversal. The udpEncapPorts field contains the port number to be used for IKEv1 UDP Encapsulation. NAT Traversal automatically uses the standard port 4500 so it does not need to be included in the udpEncapPorts field.

`useDefaultPolicy`	Set this field to true if the fields on the Policy tab for this tunnel should be updated with the values from the group LAN-LAN Defaults when the LAN-LAN Defaults are updated.
`dhGroup`	This field sets the Diffie-Hellman Group. The choices are: Group 1, Group 2, Group 5, and Group 14. The default is Group 5.
`isakmpEncryptionType`	This field sets the encryption type for the IKE SA proposal. The options are TRIPLE DES CBC (default), DES CBC, AES CBC 128, AES CBC 192, and AES CBC 256.

Field	Explanation
isakmpAuthType	This field sets the authentication type for the IKE SA proposal. The default is HMAC SHA1, but you can change this to HMAC MD5.
isakmpSaLifetimeSec	This Security Association has a lifetime specified in seconds. The default is 86400 seconds (24 hours). You can change the lifetime in seconds to any value between 60-172,860 seconds (between 1 minute and 48 hours).
ipsecProtocol	This field sets the protocol for the IPSec SA Proposal. The default is ESP-50, but this can be changed to AH-51. ESP-50 provides both encryption and authentication for every packet, while AH-51 only provides authentication.
ipsecEncryptionType	This field sets the encryption type for the IPSec SA Proposal. The options are TRIPLE DES CBC (default), DES CBC, AES CBC 128, AES CBC 192, and AES CBC 256.
ipsecAuthType	This field sets the authentication type for the IPSec SA Proposal. The default is HMAC SHA1, but you can change this to HMAC MD5.
ipsecSaLifetimeKBytes	The Security Association has a lifetime specified in seconds and kilobytes. The defaults are 14400 seconds (4 hours) and 10,000,000 kilobytes. You can change the lifetime in seconds to any value between 60 -86400 seconds (between 1 minute and 24 hours). You can change the lifetime in kilobytes to any value between 1000 - 10,000,000 kilobytes. The Security Association will expire after the first of the above two lifetimes is reached. The session will then have to re-key.
enablePerfectFwdSecrecy	By default, this value is false, If it is true, a second Diffie-Hellman key exchange will take place during processing. This can improve security, but it also can impact re-keying performance.

Field	Explanation
`enableCompression`	The compression feature only applies to LAN-LAN tunnels between a Brick with an encryption accelerator cards and any device that supports LZS compression. By default this field is false. If set to true, traffic through the tunnel will be compressed. The advantage of compression is that it means less data has to be sent over the wires, which may help conserve bandwidth and speed up transmission. The disadvantage is that it requires extra processing (to compress and decompress) on either end of the tunnel, which has performance implications at the tunnel endpoints.
`useDefaultSLA`	Set this field to true if the fields on the SLA tab for this tunnel should be updated with the values from the group LAN-LAN Defaults when the LAN-LAN Defaults are updated.
`enableSLA`	Set this field to true to enable the SLA Probe feature for this tunnel.

`srcSLA`
 	
`dstSLA`
	Set the probe source and destination IP Addresses. For Brick endpoints, the keyword TEP may be used to indicate that the Virtual Brick Address of the dnpoint should be used.

`probeInterval`	This field sets the probe interval in seconds. Allowed values are 1-21,600.
`probesPerReport`	This field sets the number of probes per report. Allowed values are 1-86,400.
`roundTripThreshold`	This field sets the probe maximum round trip delay in milliseconds. Allowed values are 10-30,000.
`pktLostWaitTime`	This field sets the probe packet lost timeout value in milliseconds. Allowed values are 10-30,000.
`e12e2manualProtocol`	This field sets the protocol for Endpoint 1 to Endpoint 2 traffic for manual key tunnels. Allowed values are ESP or AH.
`e12e2esp_spi`	This field sets the ESP SPI for Endpoint 1 to Endpoint 2 traffic for manual key tunnels. When creating new tunnels this field can be set to Auto and the Brick will assign a value. The SPI should be a hex value in the range: 100-FFFF.

Field	Explanation
e12e2esp_ EncryptionType	This field sets the ESP encryption type for Endpoint 1 to Endpoint 2 traffic for manual key tunnels. Allowed values are NULL, DES CBC, TRIPLE DES CBC, AES CBC 128, AES CBC 192, and AES CBC 256.
e12e2esp_EncryptionKey	This field sets the ESP encryption key for Endpoint 1 to Endpoint 2 traffic for manual key tunnels.[1].
e12e2esp_AuthType	This field sets the ESP authentication type for Endpoint 1 to Endpoint 2 traffic for manual key tunnels. Allowed values are NULL, HMAC MD5, and HMAC SHA1.
e12e2esp_AuthKey	This field sets the ESP authentication key for Endpoint 1 to Endpoint 2 traffic for manual key tunnels.[2].
e12e2ah_spi	This field sets the AH SPI for Endpoint 1 to Endpoint 2 traffic for manual key tunnels. When creating new tunnels this field can be set to Auto and the Brick will assign a value. The SPI should be a hex value in the range: 100-FFFF.
e12e2ah_AuthType	This field sets the AH authentication type for Endpoint 1 to Endpoint 2 traffic for manual key tunnels. Allowed values are HMAC MD5 and HMAC SHA1.
e12e2ah_AuthKey	This field sets the AH authentication key for Endpoint 1 to Endpoint 2traffic for manual key tunnels.[3].
e22e1manualProtocol	This field sets the protocol for Endpoint 2 to Endpoint 1 traffic for manual key tunnels. Allowed values are ESP or AH.
e22e1esp_spi	This field sets the ESP SPI for Endpoint 2 to Endpoint 1 traffic for manual key tunnels. When creating new tunnels this field can be set to Auto and the Brick assigns a value. The SPI should be a hex value in the range: 100-FFFF.
e22e1esp_ EncryptionType	This field sets the ESP encryption type for Endpoint 2 to Endpoint 1 traffic for manual key tunnels. Allowed values are NULL, DES CBC, TRIPLE DES CBC, AES CBC 128, AES CBC 192, and AES CBC 256.
e22e1esp_EncryptionKey	This field sets the ESP encryption key for Endpoint 2 to Endpoint 1 traffic for manual key tunnels.[4].
e22e1esp_AuthType	This field sets the ESP authentication key for Endpoint 2 to Endpoint 1 traffic for manual key tunnels.[5]

Field	Explanation
e22e1ah_spi	This field sets the AH SPI for Endpoint 2 to Endpoint 1 traffic for manual key tunnels. When creating new tunnels this field can be set to Auto and the Brick assigns a value. The SPI should be a hex value in the range: 100-FFFF.
e22e1ah_AuthType	This field sets the AH authentication type for Endpoint 2 to Endpoint 1 traffic for manual key tunnels. Allowed values are HMAC MD5 and HMAC SHA1.
e22e1ah_AuthKey	This field sets the AH authentication key for Endpoint 2 to Endpoint 1 traffic for manual key tunnels.[6]
skipOverlapCheck	Set this field to true if you got error `N7028`, which indicates that the Hosts Behind Tunnel for this tunnel overlap with hosts behind another tunnel, but you want to save this tunnel anyway.

Notes:

1. For the `e12e2esp EncryptionKey` field, the key contains hex characters and should be the proper length based on the encryption type as shown in Table 3-1, “e12e2esp EncryptionKey Field Values” (p. 3-67).
2. For the `e12e2esp AuthKey` field, the key contains hex characters and should be the proper length based on the authentication type as shown in Table 3-2, “e12e2esp AuthKey Field Values” (p. 3-68).
3. For the `e12e2ah AuthKey` field, the key contains hex characters and should be the proper length based on the authentication type as shown in Table 3-3, “e12e2ah AuthKey Field Values” (p. 3-68).
4. For the `e22e1esp EncryptionKey` field, the key contains hex characters and should be the proper length based on the encryption type as shown in Table 3-4, “e22e1esp EncryptionKey Field Values” (p. 3-68).
5. For the `e22e1esp AuthKey` field, the key contains hex characters and should be the proper length based on the authentication type as shown in Table 3-5, “e22e1esp AuthKey Field Values” (p. 3-68).
6. For the `e22e1ah AuthKey` field, the key contains hex characters and should be the proper length based on the authentication type as shown in Table 3-6, “e22e1ah AuthKey Field Values” (p. 3-69).

Table 3-1 `e12e2esp EncryptionKey Field Values`

Encryption Type	Field Length
DES CBC	16

Table 3-1 `e12e2esp EncryptionKey Field Values` **(continued)**

Encryption Type	Field Length
TRIPLE DES CBC	48
AES CBC 128	32
AES CBC 192	48
AES CBC 256	64

Table 3-2 e12e2esp AuthKey Field Values

Authentication Type	Field Length
HMAC MD5	32
HMAC SHA1	40

Table 3-3 e12e2ah AuthKey Field Values

Authentication Type	Field Length
HMAC MD5	32
HMAC SHA1	40

Table 3-4 `e22e1esp EncryptionKey Field Values`

Encryption Type	Field Length
DES CBC	16
TRIPLE DES CBC	48
AES CBC 128	32
AES CBC 192	48
AES CBC 256	64

Table 3-5 `e22e1esp AuthKey Field Values`

Authentication Type	Field Length
HMAC MD5	32
HMAC SHA1	40

Table 3-6 e22e1ah AuthKey Field Values

Authentication Type	Field Length
HMAC MD5	32
HMAC SHA1	40

Example

The following is an example of a typical lan2lan tunnel file:

```
localTep=10.10.10.251, vpnzone
remoteTep=
e1Brick=salesbrick
e1Group=system
e2Brick=
e2Group=
localHostGroup=saleshosts
remoteHostGroup=
endpointType=ip
useDefaultSLA=true
heartBeatInterval=30
description=
e12e2ah_AuthKey=
ipsecAuthType=HMAC SHA1
ipsecProtocol=ESP-50
e22e1esp_AuthKey=
unmanagedIP=56.56.56.56
probeInterval=30
email=
e12e2ah_spi=
isakmpAuthType=HMAC SHA1
useDefaultPolicy=true
e12e2manualProtocol=
enableCompression=false
preSharedkey=model8056key
comment=
remotePhase1ID=56.56.56.56
vpnType=auto
localPhase1ID=Virtual Brick Address
remoteVpnCA=
e12e2esp_AuthType=
e22e1esp_EncryptionKey=
e22e1ah_AuthKey=
receiveAnyProposals=true
probesPerReport=10
initiator=false
customerName=
ipsecSaLifetimeKBytes=10000000
unmanagedDeviceName=
e2PresharedKey=model8056key
e2IDType=IP Address
e22e1esp_spi=
remoteHostIPs=56.56.56.10
e1AuthMethod=key
e12e2esp_spi=
useDefaultParameters=true
roundTripThreshold=1000
```

```
name=model80-5656
e12e2ah_AuthType=
e12e2esp_EncryptionType=
endDate=2104-01-08 00:00:00.0
e1IDType=IP Address
isakmpEncryptionType=TRIPLE DES CBC
isakmpSaLifetimeSec=86400
dstSLA=
enableSLA=false
enableTunnel=true
enablePerfectFwdSecrecy=false
e22e1esp_EncryptionType=
e12e2esp_AuthKey=
encapType=lucent
phone=
udpEncapPorts=505
remoteDistinguishedName=;;;;;C=US
debugLevel=0
srcSLA=
e22e1esp_AuthType=
ipsecSaLifetimeSec=14400
e22e1ah_spi=
e22e1ah_AuthType=
e2AuthMethod=key
localHostIPs=
e22e1manualProtocol=
transportMethod=pureipsec
pktLostWaitTime=3000
sendAllProposals=true
e12e2esp_EncryptionKey=
ipsecEncryptionType=TRIPLE DES CBC
dhGroup=Group 1
authByDistinguishedName=true
startDate=2005-01-08 00:00:00.0
skipOverlapCheck=false
```

□

servicegroups File

Overview

The ServiceGroups file contains the fields for a service group in a given group's Each service group begins with a line that reads

```
* * * SERVICE GROUP * * *
```

Each service group field occupies a seperate line in the file.

Format

The following shows the format of each service group in the ServiceGroups file:

```
***Service Group***
protocol=
descrtion=
```

Explanation

The table below describes each field in a service group:

Field	Explanation
protocol	A tuple consisting of: the protocol name (TCP, UDP, or ICMP), or an asterisk (wildcard), the source port, and the destination port. This field corresponds to the Protocol field, the Source Port/Range field, and the **Destination Port or Range** field on the Service Editor.
descrtion	Any remarks the administrator chooses to enter. They can contain from 1-80 characters. This field corresponds to the **Descrtion** field on the Service Editor.

Example

The following is an example of a typical ServiceGroup file:

```
*** SERVICE GROUP ***
protocol=tcp/21/*
descrtion=
```

□

dependency masks File

Overview

The depmasks file contains all the fields for a dependency mask in a given group's security policy.

Each dependency mask file begins with a line that reads

```
***DEPENDENCY MASKS***
```

Each field in the dependency mask occupies a separate line in the file.

Format

The following shows the format of each dependency mask in the dependency masks file:

```
*** DEPENDENCY MASKS ***
description=
destinationIP=
service=
direction=
action=
alarmCode=
hitCount=
```

Explanation

The following table describes each field in a dependency mask:

Field	Explanation
`description`	Any remarks the administrator chooses to enter. They can contain from 1-80 characters. This field corresponds to the **Descrtion** field on the Dependency Masks Editor.

Field	Explanation
`sourceIP`	The IP address of the source host. It can be a single IP address, a host group name, the keywords SOURCE or DESTINATION, or an asterisk (wildcard). This field corresponds to the **Source IP Addr. or Group** field on the Dependency Masks Editor.
`destinationIP`	The IP address of the destination host. It can be a single IP address, a host group name, the keywords SOURCE or DESTINATION, or an asterisk (wildcard). This field corresponds to the **Destination IP Addr. or Group** field on the Dependency Masks Editor.
`service`	The protocol, destination port and source port. It can be an asterisk (wildcard), a choice from the drop-down menu, or a Service Group name. This field corresponds to the **Service or Group** field on the Dependency Masks Editor.
`direction`	The direction of packet flow relative to the Brick zone ruleset.
`act`	Determines whether the dependency mask applies to passed sessions, dropped sessions, or any sessions. This field corresponds to the **Action** box on the Dependency Masks Editor.
`alarmCode`	A code associated with an alarm. The dependency mask will only apply to sessions authorized by rules with this alarm code. It can be a number or a blank. This field corresponds to the**Alarm Code** field on the Dependency Masks Editor.
`hitCount`	Specifies the minimum number of occurrences that must be found in the session cache for a match to take place. It can be a number from 1-65535. This field corresponds to the **Hit Count** field on the Dependency Masks Editor.

Example

The following is an example of a typical dependency masks file:

```
*** DEPENDENCY MASKS ***
description=dep mask for sales
sourceIP=SOURCE
destinationIP=DESTINATION
service=tcp
direction=
act=drop
alarmCode=
hitCount=1
```

☐

4 LSMS CLI Error Codes

Overview

Purpose

This chapter explains the error codes that are returned when an LSMS command is unsuccessfully executed.

Successful Execution

If a command is executed successfully, an exit code of zero is returned, and the following message is displayed (i.e., sent to `stdout`).

`<command_name>:OK`

where <*command_name*> is the name of the command that was executed successfully.

Unsuccessful Execution

If an error occurs in the execution of a command, an exit code of one (1) is returned, and the following message is displayed (in other words, sent to `stdout`):

`<command_name>:<error_code>:<error_message>`

where

The command syntax is as follows:

- <*command_name*> is the name of the command that was executed unsuccessfully.
- <*error_code*> is the code that identifies the error.
- <*error_message*> is a brief explanation of the error.

Examples

In the following example, an administrator has logged onto the system, and issued a `gotogroup` command for group ″abc,″ which does not exist. A `gotogroup` command is then issued for group ″group1,″ which does exist, and is successful.

```
$ lsmscmd gotogroup abc
GOTOGROUP:[B6000]Group 'abc' does not exist
$ lsmscmd gotogroup group1
GOTOGROUP:OK
$lsmscmd logout
LOGOUT:OK
```

Contents

Codes	4-3

Codes

Error code table

The table below shows each error code, the message associated with it, and a brief explanation. The acronym **CLI** stands for ″command line interface.″

Error Code	Error Message	Explanation
M1002	Another login with the given adminID is in progress. Please trylogging in later.	Another administrator with the same Admin ID is in the process of logging in.
M2000	Unable to retrieve current domain.	Indicates that the CLI client is not able to connect to the CLI server.
M2001	Parameters received from servlet could not be parsed properly.	Parameters received from servlet could not be parsed properly. Occurs at the time of logging in.
M2002	Error while reading from the client.	Protocol error in data from the CLI client.
M3000	Socket connection with the client broke while receiving data.	Socket connection between the CLI server and the client broke while the CLI server was receiving data.
M3001	Connection to SMS server failed; contact the SMS system administrator to check the configuration file or the LSMS services.	Indicates that the CLI server may not be listening on the right port for connections from the CLI client.
M4000	The supplied destination directory does not have a directory named ’%1’ and unable to create the same.	%1 indicates the name of a directory.
M4001	The current user does not have write permissions for the supplied destination directory,%1.	The LSMS writes to files in the destination directory. If the user does not have write permissions to the destination directory that has been supplied, this error message results.

Error Code	Error Message	Explanation
M4002	File%1 does not exist in the specified path.	For certain commands, (e.g., save policy), the LSMS reads from certain files in the destination directory as detailed in this document. If the file(s) does not exist, this error results.
M4003	The supplied destination directory%1 does not exist and unable to create the same.	Create the given destination directory, and try to issue the LSMS command again. This error normally occurs during lsmslogon, and may occur if a destination-directory name is provided with one of the LSMS commands.
N1000	Administrator does not have permission for operation.	The administrator does not have the privilege to issue the command.
N2200	No leaf called %1 in table %2.	No leaf with the filename %1 could be found. %2 is the path to file %1.
M7000	The following data/format problems exist.\n %1	%1 gives the details of the data/format errors in the CLI file.
M7001	A session with supplied uid, %1, is currently active.	This error message is given at the time of logon. Unlike the GUI, if a session with a given userID is active through the command-line, then a second logon attempt through the command-line is blocked.
M7002	Connection with the LSMS services could not be established. Please Check if the LSMS services are running and retry after starting them.	Please check if the LSMS services are running and retry after starting them.
M7004	Unidentified protocol message sent by the client.	Protocol message used to communicate with the command-line server is in improper format. (The user should not be seeing this error if the commands lsmslogon or lsmslogon are being used)

Error Code	Error Message	Explanation
M7005	A session with the supplied session ID is not active.	Each active session is identified by a sessionID. The administrator should first log in, and then try to issue the LSMS commands.
M7006	Incorrect usage — refer to documentation.	Usage of lsmscmd or lsmslogon is imcorrect. Refer to Chapter 2, "LSMS CLI Commands"
	Invalid command — refer to documentation.	The issued command was not recognized as part of the LSMS com mand set. Check for misspellings or other incorrect entries. Refer to Chapter 2, "LSMS CLI Commands"
	Incomplete command — refer to documentation.	This message is returned when the user types ″lsmscmd″ with no command following. Refer to Chapter 2, "LSMS CLI Commands"

5 Archive Recovery

Overview

Purpose

If you have chosen to operate the LSMS with the ″Detailed Policy Auditing″ function activated, you can recover earlier versions of individual Lucent VPN Firewall *Brick*®devices, Brick zone rulesets, host groups, service groups, application filters and dependency masks.

This can be a helpful function if you only need to recover one or two entities to the LSMS database. While you can always perform an LSMS ″restore″, the drawback is that you must restore the entire database.

To verify whether you have ″Detailed Policy Auditing″ enabled, check the setting in the LSMS Configuration Assistant. For assistance with the Configuration Assistant, please refer to Chapter 8 in the *LSMS Administration Guide*.

Contents

Creation of Archived Entities	5-2
Recovery of Archived Entities	5-3

Creation of Archived Entities

Overview

If you have the ″Detailed Policy Auditing″ option enabled (it is activated by default) in the LSMS Configuration Assistant, a directory called ″archive″ is automatically created in the root directory of your LSMS installation (default choices are \users\isms\lmf for Windows and /opt/isms for Solaris).

Whenever a change to a Brick, Brick zone ruleset, host group, service group, application filter or dependency mask is saved to the LSMS database, a copy of the previous configuration is written in the archive directory. A separate sub-folder is created for ″brick″, ″host group″, and so forth.

When the archive copy is saved, it is given a unique file name in the following format:

<entity name>_<admin name>_<date time stamp>.txt

The fields in the file name are:

- Entity Name - the name of the Brick, host group, or other entity as it is saved in the database
- Admin Name - the login ID of the administrator who created or saved the previous version of the entity
- Date Time Stamp - the time that the previous version was created or saved in the *MM_DD_HH_MM_SS* format.

As an example, suppose that admin ″jsmith″ created a Brick called ″brick10″ on June 4 at 9:30AM. Then, on June 12, admin ″lmiller″ makes a change to ″brick10″ and saves it to the database. A copy of the previous Brick configuration is saved in the LSMS installation directory in the archive/brick folder. The file name will be:

brick10_jsmith_Jun_04_09_30_07.txt

If you ever wish to restore the June 4 version of the Brick, you can simply use the appropriate CLI command, as outlined in the steps below.

□

Recovery of Archived Entities

Task

In order to recover an archived item, the admin should:

1 *Locate the desired file* and verify its contents. Go to the ″archive″ directory, and proceed to the proper sub-folder and identify the file that you wish to restore. You may edit the file as needed.

2 *Run the ″validateHash″ tool* to ensure that the file has not been altered since it was saved.

When the archive file is created, it is saved with a hash value. Execute this utility if you need to verify that the file contents are unchanged since the file was saved.

In order to run the tool, you must be in the LSMS Installation directory. The command format is:

```
local/bin/validateHash <archive file name> <hash value>
```

To obtain the hash value, first check the date time stamp on the archive file name. In the LSMS Event Log, under the ″History″ tab, bring up the time that the object was modified and highlight the hash value. The value may then be pasted on the command line with the rest of the validateHash command.

3 *Copy the item to the admin's login directory* for LSMS Command Line Interface (CLI) functions and rename it. When an admin logs into the LSMS CLI, you'll recall from Chapter 2 that the format of the command is:

```
lsmslogon <admin ID> <destination_directory> [-p <password file> or -f
   password]
```

Copy the archive file to the ″destination directory″ used for the CLI login. Rename the archive file to the ″entity name″ (i.e. strip off the admin name and timestamp from the file name.

4 *Save the archived entity* to the LSMS database with the appropriate LSMS CLI command. Remember, if you have restored a Brick or a Brick zone ruleset, you may also need to ″apply″ the entity in order to activate the change on the Brick.

END OF STEPS

6 Database Utilities

Overview

Purpose

There are a number of simple database utilities that can be found under the LSMS installation directory (e.g. \users\isms\lmf on Windows or /opt/isms/lmf on Solaris) in the local/bin folder. Some of the tools are present for support personnel, while others may be run by administrators.

All of the commands must be executed from the command line or a terminal window while in the LSMS installation directory. For instance, if a Windows admin wanted to run `dbsetup`, he would open a DOS window, ″cd″ to \users\isms\lmf and enter:

local\bin\dbsetup

In this chapter, only administrator commands will be described. `Do not attempt` to run any of the other utilities.

Contents

allowSecondarySetup	6-2
backup	6-3
changeInactiveBrickVersion	6-4
changeIP	6-5
changeName	6-6
convertToPrimary	6-7
dbsetup	6-8
restore	6-10
validateHash	6-11

allowSecondarySetup

Description

In a redundant LSMS configuration, the Primary LSMS must always be installed first. When the secondary LSMS is initially installed, a default certificate is used to encrypt communication between the databases. At the conclusion of the secondary installation, a random certificate is generated for subsequent communication between the databases.

In the event that the database must be restored on the secondary LSMS (i.e. copied from the primary LSMS), the certificate on the secondary is temporarily reset to the default. In order for the secondary to copy the database from the primary, the `allowSecondarySetup` command must be run on the primary. At the conclusion of the secondary restore, a new random certificate is created.

□

backup

Description

This utility is used for database backup. For more information, please review the *LSMS Administration Guide.*

□

changeInactiveBrickVersion

Description

This utility is used to change the Lucent VPN Firewall*Brick*® device software version internally in the LSMS database for staged Bricks that have not yet been activated. This utility must be run for each inactive Brick after a software upgrade to R8.0 in order to be able to manually enable the IKE on the Brick feature.

local/bin/changeInactiveBrickVersion brickName

□

changeIP

Description

This utility is used to change the IP address of the LSMS in the database after installation. For more information, please review *Changing the IP Address of the LSMS in the LSMS Administration Guide.*

□

changeName

Description

This tool allows an adminsitrator to change the name of the LSMS in the database after installation.

- Stop LSMS Services
- From the LSMS installation directory, run :
  local/bin/changeName <new LSMS nane>
- Start LSMS Services

☐

convertToPrimary

Description

This utility is not applicable to LSMS R8.0 and is kept in the documentation as a reference if some customers are still running older versions (up to LSMS R7.2). Starting from LSMS R8.0, even a single LSMS is installed as a Primary LSMS and there is no concept of a standalone LSMS.

With this utility, an administrator can convert an existing standalone LSMS to serve as a primary LSMS. However, this utility cannot be used to convert a secondary LSMS into a primary LSMS.

When this utility is execute, the database is updated and prepared to interact with a secondary LSMS.

- Stop LSMS Services
- From the LSMS installation directory, run:
  local/bin/convertToPrimary
  The user must answer answer a series of questions similar to the database portion of a new installation (see the *LSMS Installation Guide* for more details).
- Start LSMS Services
  After the services have been restarted, a secondary LSMS may be installed at any time.

☐

dbsetup

Description

When run on a Primary LSMS server, the *dbsetup* command can be used to reinstall a clean (empty) database or to set up the database after a restore of a designated backup database is completed.

When run on a Secondary LSMS server, the *dbsetup* command manually synchronizes its database by making an exact copy of the Primary LSMS database. This must be done after performing a database restore on the Primary LSMS, or if the Primary and Secondary LSMS(s) have not been communicating for more than a week.

The *dbsetup* command must be executed in the following database installation/restore scenarios:

- On a Primary LSMS server that is running R8.0.275 or earlier, the *dbsetup* command is run on the Primary LSMS server after a restore of the backup database is completed
- A clean (empty) version of the database is installed on an LSMS server without re-installing the LSMS
- A manual resynchronization of the Secondary LSMS database after performing a database restore on the Primary LSMS server or if the Primary and Secondary LSMS servers have not been communicating for more than a week

In cases where the Primary LSMS database has been restored from backup, or a clean (empty) version of the database is installed, *dbsetup* must be run on the Secondary LSMS(s) to resynchronize their database(s) with the Primary LSMS database.

To set up the database on a Primary LSMS after a restoring the database from a backup:

1. Stop the LSMS services.
2. Use the *restore* utility to revert to a designated backup database (see the *restore* command description below) or recreate the objects in the LSMS database (Bricks, rulesets, and so forth) manually.
   If the Primary LSMS server is running R8.0.275 or earlier, go to Step 3. If the Primary LSMS server is running a release later than R8.0.275, *dbsetup* is run automatically when the restore utility is invoked. In this case, skip Step 3.
3. From the LSMS installation directory, run: `/local/bin/dbsetup`
4. Start the LSMS services.

To install a “clean” (empty) version of the database on an LSMS server, do the following:

1. Stop the LSMS Services.
2. Remove the <*installdir*>/db/LSMS folder.

3. From the LSMS installation directory, run: `local/bin/dbsetup`
4. Start the LSMS Services.

To manually resynchronize the database on the Secondary LSMS after performing a database restore on the Primary LSMS or if the Primary and Secondary LSMS(s) have not been communicating for more than a week, do the following:

1. On the Primary LSMS, run the *allowSecondarySetup* utility.
2. On the Secondary LSMS, stop the LSMS Services.
3. On the Secondary LSMS, from the LSMS installation directory, run: `local/bin/dbsetup`
4. On the Secondary LSMS, start the LSMS Services.

□

restore

Description

This utility is used for database restore. For more information, please review *Chapter 11. Backing Up and Restoring Data in the LSMS Administration Guide.*

□

validateHash

Description

With the ″Detailed Policy Auditing″ checkbox enabled in the Configuration Assistant, a record is kept of all changes to Bricks, Brick zone rulesets, host groups, service groups, application filters and dependency masks. These files are preserved in the ″archive″ folder under the LSMS installation directory. Each file is saved with a randomly generated hash value. If necessary, the files can be restored with the appropriate LSMS (command line interface) commands.

This tool may be run prior to restoring a file with an LSMS ″CLI″ command to verify that its contents have not been altered.

local/bin/validateHash `<archive file name> <hash value>`

To obtain the hash value, first check the date time stamp on the archive file name. In the Event Log, under the ″History″ tab, bring up the time that the object was modified and highlight the hash value. The value may then be pasted on the command line with the rest of the validateHash command.

☐

7 LSMS Service Status

Overview

Purpose

Another tool that is available for LSMS Administrators is ″LSMS Service Status″. This utility presents a graphical representation of the resource usage of all eight individual LSMS processes as well as a summary total of all the processes.

This tool can be used to monitor CPU and memory usage. The ″Maximum Heap″ column corresponds to the value set in the LSMS Configuration Assistant under the option ″Tunable Parameters″. Depending on the ″Used Heap″ value as observed over the long term, you may elect to increase or decrease the Max Heap value in the Configuration Assistant.

While the utility does not have the ability to present a ″historical″ view of past resource usage, you may elect to keep the LSMS Service Status window up for an extended period and use the tool to observe patterns of resource usage.

For more information on the LSMS Configuration Assistant, please refer to *Chapter10. Using the Configuration Assistant* in the *LSMS Administration Guide.*

Contents

Displaying LSMS Service Status	7-2
Summary View	7-3
Individual Service View	7-5

Displaying LSMS Service Status

Overview

There are two different means to display the LSMS Service Status.

1. *From the local LSMS* — On a Windows platform, proceed to **Start** → **Programs** → **Lucent Security Management Server** → **Utilities** and select **LSMS Status.**
   If you are on a local Solaris LSMS, proceed to the LSMS installation directory (default location is */opt/isms/lmf*) and enter:
   ./StartLSMS Status
   Or, from the Solaris menu, navigate to **Lucent Security Mangement Server** → **Utilities** and choose **LSMS Status**.
2. *From the LSMS Navigator or the LSMS Remote Navigator* — Proceed to the Utilities menu, select **System Utilities,** and then select **LSMS Service Status**.

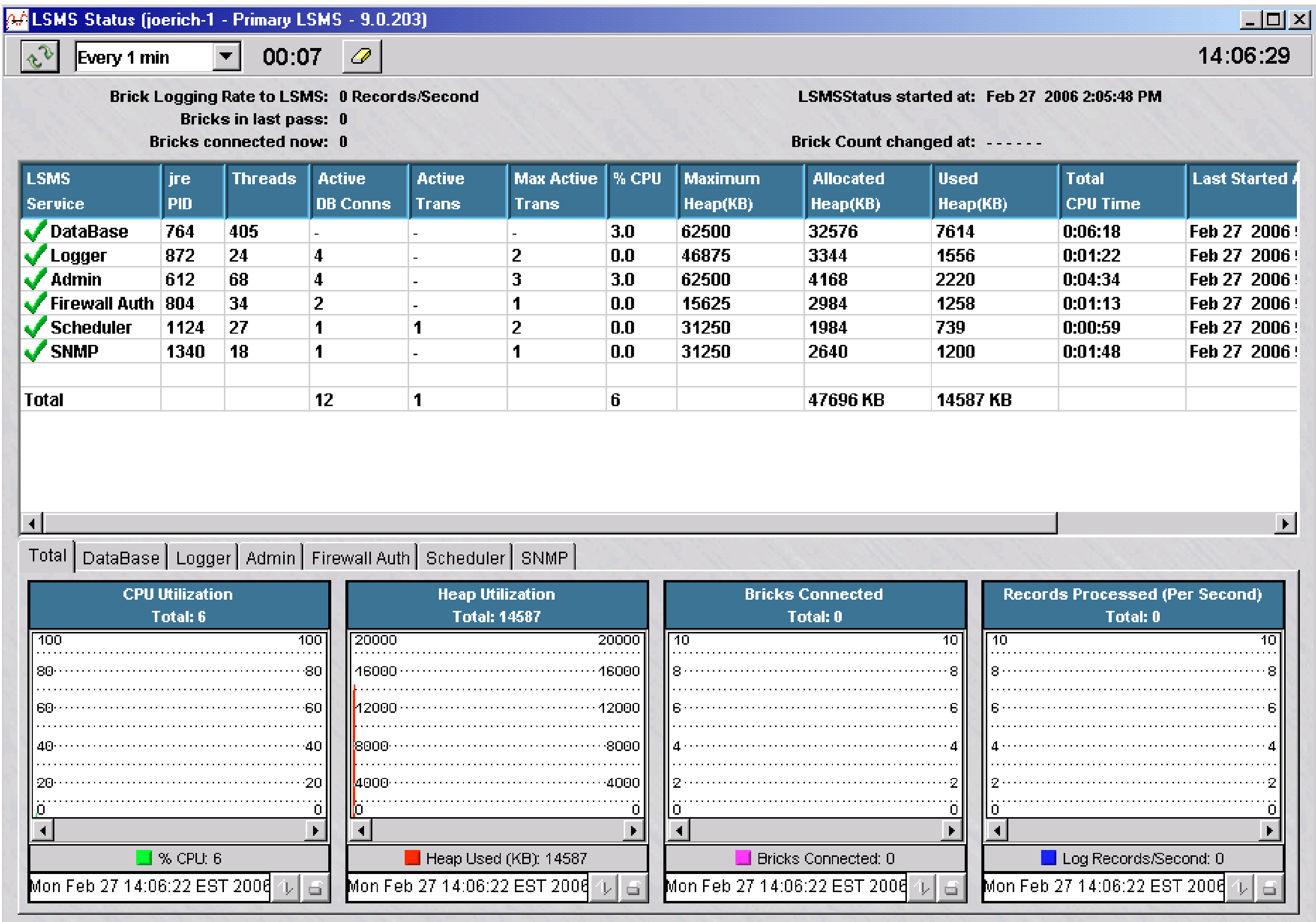
☐

Summary View

Service status view

There are two slightly different presentations available from the LSMS Service Status. When first displayed, the user is shown is the summary view as listed in Figure 7-1, “LSMS Status — Total” (p. 7-3):

Figure 7-1 LSMS Status — Total



In either presentation, the top half of the screen is always the same. Each LSMS service is listed along with its relevant statistics. By default, the display is updated every 30 seconds, but the refresh interval can be modified via the pull-down in the upper left corner of the display.

When the ″Total″ tab is chosen in the lower half of the display, the following statistics are shown graphically:

- CPU Utilization — The total amount of CPU taken by all the LSMS services.
- Heap Utilization — The total memory heap used by all of the LSMS services.

- Bricks Connected — The total numbers of Lucent VPN Firewall *Brick*® devices connected to the LSMS.
- Records Processed per second — The average number of log records processed per second by all the Bricks.

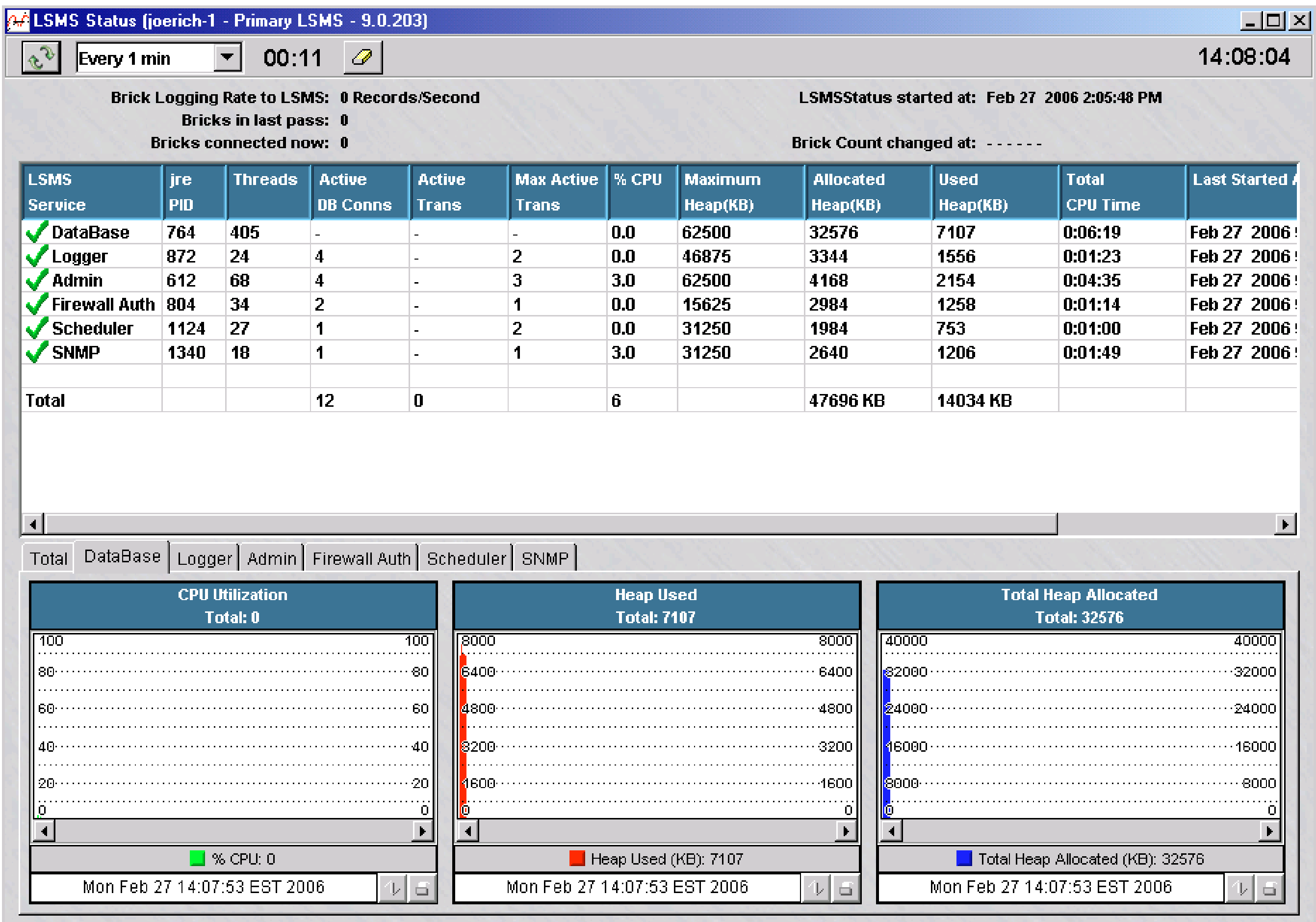
□

Individual Service View

Individual service statistics

At any time, the administrator can opt to select any other tab for any of the individual LSMS services. As mentioned above, the top half of the screen always lists the statistics for each service. However, as you can see below, the graphs presented in the lower half of the display are slightly different than the summary view, as shown in Figure 7-2, “LSMS Status — Single” (p. 7-5).

Figure 7-2 LSMS Status — Single



For each individual LSMS service, graphs are shown for CPU Utilization, Heap Used and Total Heap Allocated.

8 Troubleshooting Resources

Overview

Purpose

This chapter is intended to acquaint the user with the tools that are available for troubleshooting problems with the LSMS and the Lucent VPN Firewall *Brick*® device.

The following areas will be discussed:

- Online help and documentation
- Log Files and reports
- Brick problems
- VPN tunnel problems

Contents

Online Help and Documentation	8-2
Log Files and Reports	8-3
Brick Problems	8-5
VPN Tunnel Problems	8-6

□

Online Help and Documentation

Overview

Once you have logged into the LSMS Navigator or LSMS Remote Navigator, there is a ″Help″ menu selection available on nearly every screen. In that list, you can choose to view:

- Product Manuals - All of the manuals are available in Acrobat's ″pdf″ format. Free Acrobat Reader software is provided on the LSMS CD shipped with the system or may also be obtained at www.adobe.com
- Contents - A listing of all help topics on the system, including an ″Index″ and ″Search″ mechanism similar to those found in Windows help.
- Error Codes - A list of all Brick and LSMS error codes that are reported in the Event Log in the LSMS Logviewer or the Administrative Events Report, along with explanations and / or suggested recovery actions.

☐

Log Files and Reports

LSMS LogViewer

One of the keys in quickly diagnosing any problem is to review the data from the Bricks that is being recorded in the log files on the LSMS.

The user may examine the system events from the LSMS LogViewer or by running a report. The LogViewer can be run locally on the LSMS or from the GUI on an LSMS Remote Navigator. The user can run the LogViewer locally or remotely in ″real time″ mode while an admin logged into the local LSMS standalone host can also review earlier ″historical″ system events.

On a Windows LSMS, go to Start > Programs > Lucent Security Management Server > Utilities > Logs. When the LogViewer window is shown, log in with your LSMS administrator or group administrator ID and password, and choose the desired log.

On a Solaris LSMS, you can bring up the LogViewer from the Solaris menu. Go to Lucent Security Management Server > Utilities > LogViewer. When the LogViewer window is shown, log in with your LSMS administrator or group administrator ID and password, and choose the desired log.

If you are logged in via the LSMS Navigator or the LSMS Remote Navigator, go to the ″Utilities″ menu bar, select System Utilities, and then LSMS LogViewer. Click on the preferred log.

You may view the following LSMS logs:

- *Event Log* - This log records a number of error conditions as well as routine administrative events and VPN tunnel transactions. This is a vital tool for configuration and maintenance issues with Bricks and VPN tunnels.
- *Session Log* - This log contains Brick session records, which record network activity through each of the device interfaces as well as whether a packet was ″passed″ through a Brick or ″dropped″. This is also a very important tool to track the path of a packet if your Brick is not communicating with the LSMS, a VPN tunnel will not come up, or similar situation.
- *User Authentication Log* - This log records all user login attempts. (Administrator logins are recorded in the Event Log).
- *Proactive Monitoring Log* - This log records statistical information about the LSMSs and the Bricks that it is managing.
- *VPN Logs* - These logs contains events and alarms logged about VPN tunnels.

The user may also configure filters and / or use the ″Find″ option on the LogViewer to more closely monitor specific events. For more information on the logs, please review the *Reports, Logs and Alarms Guide*, which is available under ″Help″ on the LSMS Navigator.

Reports

Once logged into the LSMS Navigator or the LSMS Remote Navigator, the user can run a number of different reports. The reports are displayed in a browser. While real-time events may only be observed on the LSMS LogViewer, historical data can be reviewed from the LogViewer or from a report. Filters can also be used to make the reports as concise as possible.

There are five different reports that can be run. For more information on reports, please review the *Reports, Logs and Alarms Guide*, which is available under ″Help″ on the LSMS Navigator.

□

Brick Problems

Troubleshooting a Brick

The Brick is an extremely flexible device. It may be configured to act as a bridge or as a router and perform as a pure firewall or as a VPN tunnel endpoint. You may wish to review the *Technical Overview* document or *Chapters 3 - 7* in the *LSMS Administration Guide* for more information on Brick functionality.

In today's complex internetworking environments, it is always helpful to have a detailed picture of your network topology available when troubleshooting. The Brick may appear to be having a problem, when the actual cause may be due to a failure or misconfiguration of another network element.

The Status Monitor on the LSMS provides a variety of information about all of your Bricks, as well as detailed information on each single brick. For example, you can see if the Brick is ″UP″ (i.e.communicating with the LSMS), which LSMS the Brick is homed to, packet throughput, or other data, the Status Monitor may be accessed from menu bar on the LSMS Navigator, or you may log into it directly. For more information, refer to *Chapter 12 Using the Status Monitor* in the *LSMS Administration Guide*.

While the logs on the LSMS can provide useful messages for troubleshooting Brick problems, it is often critical to observe what is happening on the device itself. In order to monitor Brick activity first hand, a console connection must be set up to the Brick. There are a variety of means available to establish a console connection as outlined in *Chapter 8 Introduction to the Brick Command Line Interface* elsewhere in this manual.

Once a console connection is established to the Brick, the user may enter any of the Brick commands as described in the Brick Command Line Interface portion of this guide. Of particular interest when troubleshooting the Brick is the ″trace packet″ commands. This family of commands can be used to monitor some or all session activity through a specific port. For more information on the use of trace commands, please refer to *Chapter 10 Trace Commands* elsewhere in this manual.

Finally, the admin may also execute from the Brick console ″ping″ and ″traceroute″ commands to and from Bricks or hosts of interest. Please consult *Chapter 5 Maintaining a Brick Configuration* in the *LSMS Administration Guide* for more information on the use of ping and traceroute in Brick troubleshooting.

□

VPN Tunnel Problems

Troubleshooting tunnel problems

When troubleshooting LAN-LAN or IPSec Client - LAN tunnel problems, it is important to gather as many clues as possible from both the LSMS and the Bricks. The most common cause of VPN tunnel failure is a misconfiguration on the LSMS or a routing problem between the tunnel endpoints on the Bricks. In rare cases, it may be necessary to reboot the Brick(s) or restart the LSMS services.

In order to eliminate routing or connectivity issues in your configuration, you may want to try a simple case first. Create and add a basic ruleset to each Brick to allow, say, ″pings″ to pass through to hosts behind each Brick. If that test is successful, you have confirmed that the problem must be strictly VPN related.

On the LSMS, here are some of the items that the administrator can review:

- Rule Configuration - When configuring a Brick, a user must define a Virtual brick Address (VBA) in the Policy Assignment tab for a Brick zone ruleset to be incorporated as part of a tunnel. In addition, you must have at least one rule in the Brick zone ruleset where the ″Action″ field is set to ″VPN″.
- Tunnel Status - For a LAN-LAN tunnel, proceed to the LAN-LAN Tunnel Viewer in the VPN folder within the group where your tunnel is configured. Check that the value of the ″Enabled″ column is ″Yes″; if not, open the LAN-LAN Tunnel Editor and check off the ″Enable Tunnel″ box on the Main tab. Also verify that the ″Status″ of the tunnel is ″Up″.
  For a client tunnel, the admin can confirm whether a given user's IPSec tunnel is up by checking under VPN > Client Tunnel Endpoints. Highlight the tunnel in question and right click and choose the ″Show Tunnels″ option.
- Event Log - As indicated earlier in this chapter, the Event Log portion of the LSMS LogViewer can provide a number of helpful details on VPN tunnel issues. However, by default the system only records a subset of the potential VPN messages in the Event Log. To gather the greatest breadth of detail on VPN messages, you must proceed to the LSMS Configuration Assistant. Under the ″VPN Debugging″ parameter, set the audit level to ″3″.
  After resolving your VPN issue, you may wish to reset to audit level to ″0″ or ″1″ to preserve system resources.
  For more information on the LSMS Configuration Assistant, please refer to *Chapter 10 Using the Configuration Assistant* in the *LSMS Administration Guide*.

On the Brick console, you may wish to review:

- display policy <zone> - Verify that all of the rules (and changes) have been downloaded to the Brick. You may need to ″save & apply″ the ruleset from the LSMS.
- displays a <zone> - Check that there are active security associations (SAs) for your tunnel. If there are no SAs, there may be a tunnel, Brick, or zone ruleset misconfiguration on the LSMS, or the Brick was not able to receive its SAs from the LSMS for some reason.
- display hostgroups <zone> - Verify that the desired hostgroup is present and that it lists all of the hosts configured for this group on the LSMS. If not, you may wish to apply the Brick or the LAN-LAN tunnel.
- trace commands - You can run ″trace packet″ commands to observe the packet activity directly on the Brick.

Please reviews Chapters 9 and 10 elsewhere in this guide for more information on helpful commands to use in the Brick console.

☐

9 Introduction to the Lucent VPN Firewall *Brick*® CLI

Overview

Purpose

The Brick command line interface (CLI) provides a way to issue commands directly to a Brick for query purposes or for troubleshooting. Some methods of establishing a connection to the Brick console are always available, while others can only be established if the brick and the LSMS are communicating.

There are a total of five ways to create a connection to the brick console. Two of the methods may only be used if the Brick and the LSMS are communicating (in other words, the Brick is ″UP″ in the Status Monitor), while the remaining techniques do not depend on whether the Brick connection to the LSMS is up.

Each method provides similar functionality on the Brick; the administrator can simply select the most convenient choice for the circumstances, as follows:

1. Remote Console Connection from the Navigator — From the LSMS Navigator or the LSMS Remote Navigator, an LSMS administrator (or Group Administrator with ″Full″ device privileges) can highlight a Brick, right click the mouse, and choose the ″Open Brick Console″ option. A separate console window is opened and described in further detail on the next page.
2. Remote Console Connection from the Command Line — From the local LSMS host only, an LSMS administrator (only) opens a terminal window or a DOS window. After navigating to the LSMS installation directory, the user runs ″brickcon″ (Windows) or ″brickcon.sh″ (Solaris) and completes the login sequence, as described on page 8-5 and “Remote Console Connection from Solaris” (p. 9-8).
   The commands are entered and executed within the window that you used to run brickcon.
   There are three other methods available to connect to the Brick console. If you use any of these alternatives, it does not matter if the LSMS and Brick are communicating.

3. Local Connection - Anyone with physical access to the Brick can connect a monitor and keyboard to the appropriate ports on the back of the Brick. As soon as the Brick and the monitor are both powered up, the Brick console is displayed and commands may be entered. No password is required, unless it is enabled on the Brick configuration.
4. Remote Dial-in Connection - Connect an external modem to the serial port on a Brick and dial into the Brick from a remote computer equipped with a modem and a terminal emulation program such as HyperTerminal. After establishing a connection to the modem through Hyperterm, you must login to the Brick with the Serial Port Access password (defined on the ″Options″ tab of the Brick) to create a connection to the Brick console.
5. Local Serial Port Connection - Connect a computer equipped with a terminal emulation program such as HyperTerminal to the serial port on the Brick. Similar to the remote dial-in connection, once you connect to the port through Hyperterm, you must login to the brick with the Serial Port Access password (defined on the ″Options″ tab of the Brick) to create a connection to the Brick console.
   The first three methods are described in this chapter. For an explanation of the remote dial-in connection and local serial port connection, refer to Appendix A, “Set up a Remote Dial-In Connection” and Appendix B, “Set up a Direct Serial Port Connection”, respectively.

Contents

Remote Console Connection via the Navigator	9-3
Remote Console Connection from the Command Line	9-5
Remote Console Connection from Windows	9-6
Remote Console Connection from Solaris	9-8
Set up a Local Connection	9-10
Command List Introduction	9-11

Remote Console Connection via the Navigator

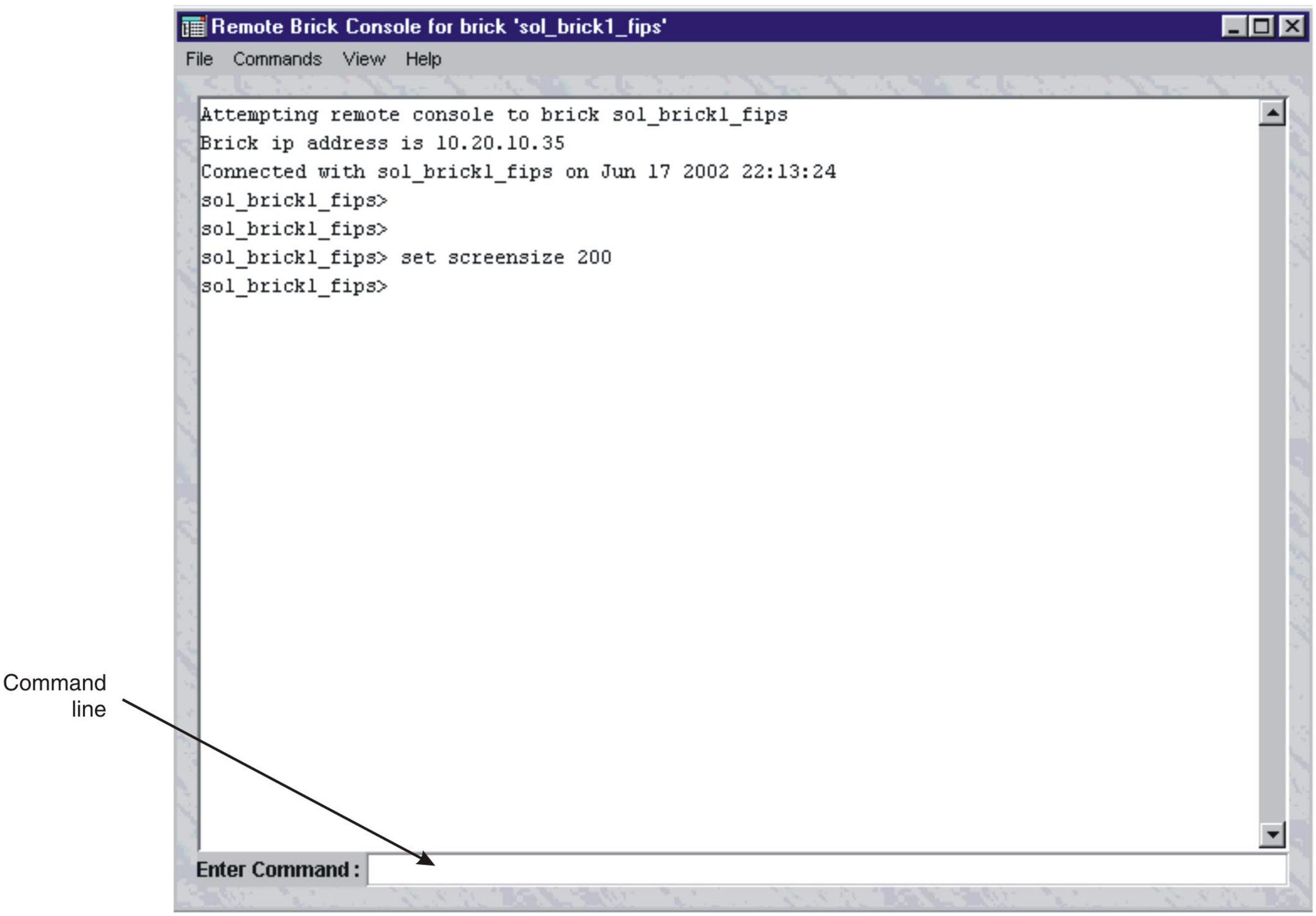
Task

To open a console window on a Brick from the LSMS interface, follow the steps below:

1 With the Navigator window displayed, click the appropriate **Bricks** folder to display the Bricks in the folder.

2 Right-click the Brick you want and select **Open Brick Console** from the pop-up menu. The Brick Console window will appear. It is shown in .

Figure 9-1 Brick Console Window



3 To execute a command, you can do either of the following:

- Enter the command in the command line at the bottom of the window
- Select the command by clicking **Commands** in the menu bar at the top of the window, and selecting the command from the Command menu. (If you do not see the command you are looking for, note that several of the options in the Commands menu contain submenus.)

END OF STEPS

Remote Console Connection from the Command Line

Overview

The remote console connection from the command line feature allows LSMS Administrators (only) to create a Brick console connection from the local LSMS host.

Note that the LSMS must be able to communicate with the Brick to which you are attempting to open a Remote Console session. If the link between the Brick and the LSMS is down, you will need to use a direct connection, a local serial port connection, or a dial-up connection to access the Brick CLI.

Only LSMS Administrators may use the Remote Console feature. Group Administrators do not have permission to access this feature.

Only *one* instance of the remote console application may connect to a Brick at a time. However, you can run multiple instances of the remote console simultaneously from the LSMS host to different Bricks. Also, multiple LSMS Administrators can be logged in, each in their own remote console session.

If another LSMS Administrator attempts to connect to a Brick that is already supporting a remote console session, the new remote console session overrides the existing session, and the existing session is immediately disconnected from the Brick.

The first remote console user is informed that they were disconnected due to a new remote console session to that same Brick. The first user will also be informed of the LSMS Admin ID of the new remote console session that caused them to be disconnected. The new remote console session will also report to the second user the fact that their session just overrode an existing remote console session, and the LSMS Admin ID of the first remote console session.

A remote console session on a particular LSMS can only connect to Bricks managed by that LSMS. The remote console cannot connect to Bricks managed by another LSMS.

The connection from the LSMS to a Brick uses a TCP connection on a TCP destination port already in use from the LSMS to the Brick. This port is preconfigured in the ″LSMS to Brick Services″ service group.

If there are one or more Bricks between the LSMS and the Brick to which the user intends to connect, appropriate rules must exist in all intervening Brick zone rulesets on those bricks such that the TCP connection may pass through these Bricks to get to the intended Brick.

When a remote console session is connected to a Brick, all keyboard input, commands executed, and responses generated at any one of the three console interfaces (serial/dial-in, local, or remote) will be echoed to all other interfaces.

□

Remote Console Connection from Windows

Task

1 From the Windows command line, change the directory to the LSMS installation directory (default option is *C:\\users\\isms\\lmf*).

2 Issue the command `brickcon`

3 You will receive the prompt:

```
Please enter admin ID:
```

4 Enter your admin ID and press Enter.

5 You will receive the prompt:

```
Please enter password:
```

Enter the password for this administrator ID and press Enter.

6 The remote console will display the prompt

```
The bricks on your system are:
```

followed by a list of Brick names accessible from this LSMS.

7 You will receive the prompt:

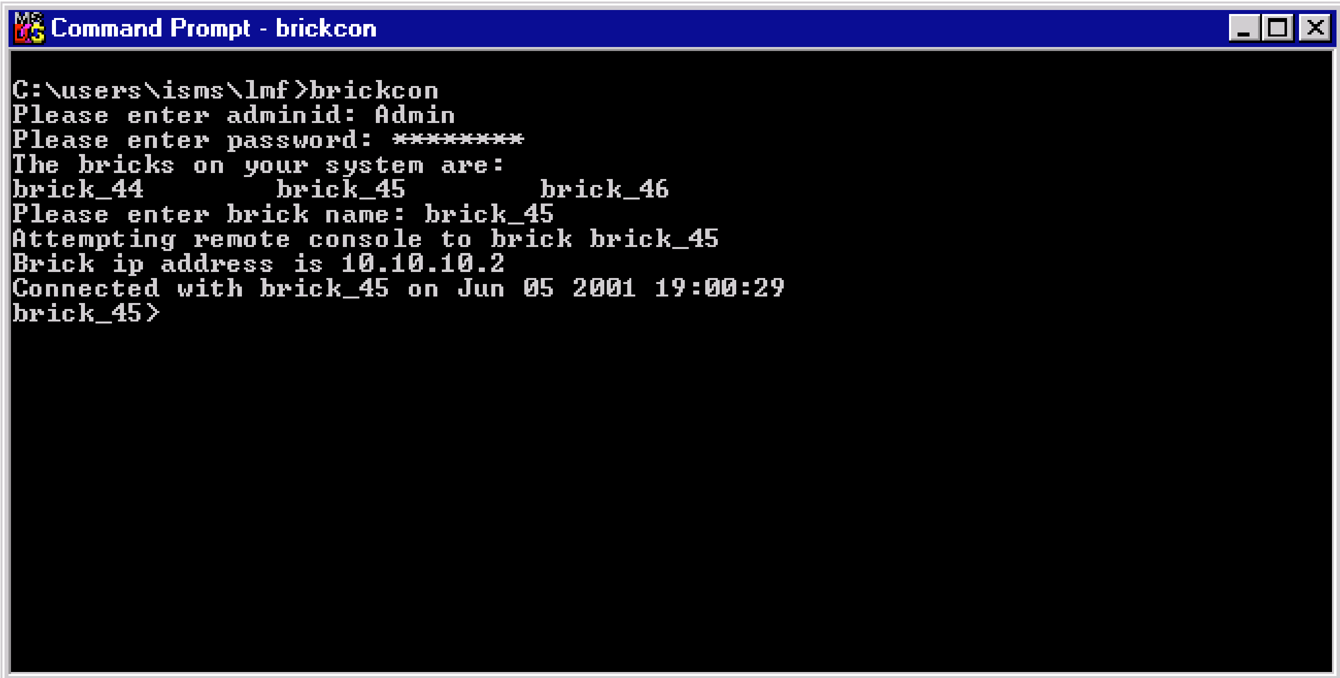
```
Please enter brick name:
```

Enter the name of the Brick you want to open a remote console session to and press Enter. The remote console attempts to connect to the Brick you specified.

8 If the connection attempt is successful, the remote console displays the IP address of the Brick and the date and time the connection was established. The command line

prompt changes to the name of the Brick to which you are connected. See Figure 9-2, “Windows Brick Remote Console Session” (p. 9-7).

Figure 9-2 Windows Brick Remote Console Session



9 If the connection attempt is not successful, the remote console application returns the prompt:

`Unable to find the brick <brickname> in the system.`

You can also issue the brickcon command with all, or some, of its arguments on a single line. The complete syntax of the brickcon command is:

```
brickcon <brickname>
<lsms_admin_id>
<password> [s=script_pathfile] o=output_patfile]
```

Where the optional argument script_pathfile is the path and file name of a script file the Remote Console will execute, and the optional argument output_pathfile is the path and name of an output file to which the remote console will write its output.

Note that if you issue the entire command in a single line in this fashion, you will be typing your password in the clear.

END OF STEPS

Remote Console Connection from Solaris

Task

1. From the command line, change the directory to the LSMS installation directory (default option is */opt/isms/lmf*).

2. Issue the command `brickcon.sh`

3. You will receive the prompt:

   `Please enter admin ID:`

4. Enter your admin ID and press Enter.

5. You will receive the prompt:

   `Please enter password:`

   Enter the password for this administrator ID and press Enter.

6. The remote console will display the prompt

   `The bricks on your system are:`

   followed by a list of brick names accessible from this LSMS.

7. You will receive the prompt:

   `Please enter brick name:`

8. Enter the name of the Brick you want to open a remote console session to and press Enter. The remote console attempts to connect to the specified Brick.

9. If the connection attempt is successful, the remote console displays the IP address of the Brick and the date and time the connection was established. The command line prompt changes to the name of the Brick to which you are connected.

10 If the connection attempt is not successful, the remote console application returns the prompt:

```
Unable to find the brick <brickname> in the system.
```

You can also issue the brickcon command with all, or some, of its arguments on a single line. the complete syntax of the brickcon command is:

```
brickcon <brickname> <lsms_admin_id>
<password> [s=script_pathfile] o=output_patfile]
```

Where the optional argument script_pathfile is the path and file name of a script file the Remote Console will execute, and the optional argument output_pathfile is the path and name of an output file to which the remote console will write its output.

Note that if you issue the entire command in a single line in this fashion, you will be typing your password in the clear.

END OF STEPS

Set up a Local Connection

Introduction

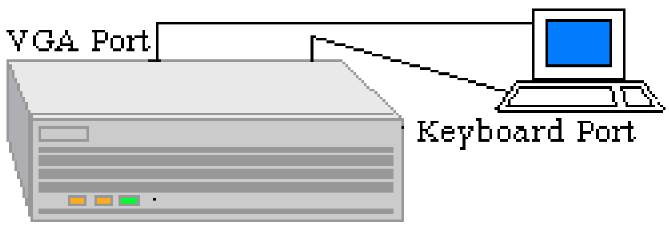
Another way to access the Brick CLI is to set up a local connection. Using a standard monitor cable, connect a monitor and keyboard to the monitor and keyboard ports on the back of a Brick.

A local connection makes it possible to observe the Brick status even if the LSMS is lost.

As illustrated in Figure 9-3, "Local Connection" (p. 9-10), a keyboard and monitor are connected directly to the ports on the back of the Brick.

Note that in this configuration, since you have a direct local connection, the `login` and `logout` commands are not applicable.

Figure 9-3 Local Connection



Procedure

To set up a local connection, do the following:

1 Connect the keyboard and monitor to the keyboard and monitor ports on the back of the Brick. The location of these ports depends on the model of the brick. Refer to the *User's Guide* of the respective Brick model for a description.

2 Turn on the monitor. The Brick prompt appears.

END OF STEPS

Command List Introduction

Overview

Once connected and logged in (local connections do not need to login), commands are issued to a Brick to perform tracing or display statistics such as port status, contents of a Brick zone ruleset security policy, and related information.

All commands, Brick zone ruleset names, and identifiers are case-insensitive. The only case-sensitive text is the Remote Password that is created in the Brick Editor.

Command categories

To see a complete listing of all commands online, enter `help` at the command prompt.

Some of the commands can be grouped as follows:

- `display` commands
  These commands provide information that is relevant to the Brick, such as the policy, host group definition, or session cache of a given Brick zone ruleset.
- `set` commands
  These commands manipulate specific settings that control the viewing of the output, such as screen size, baud rate, status of real-time printing.
- `trace audit` and `trace packet` commands
  These commands control how audit records and real-time incoming and outgoing packets are filtered and displayed.
- `misc` commands
  Other commands provide miscellaneous functions such as rebooting the Brick, disabling brick zone rulesets on the Brick, and refreshing the MAC table.

General Commands

After logging into a Brick, some general commands that can be used are as follows:

<Ctrl> `C`	Terminates viewing real-time output. Can be used if the display is overwhelmed with too much output.
<Esc> `key`	Re-enables the viewing of real-time output.
<Tab> `key`	Use to cycle through a Brick zone ruleset or keyword list.
`Spacebar or` *<Enter>* `key`	Use to complete a partially-entered command.

When viewing non real-time output, 23 lines are displayed on the monitor. To view additional output, press any key.

You can override this default by using the `set screensize` command.

10 Lucent VPN Firewall *Brick*® Device Display Commands

Overview

Purpose

This chapter provides information to perform the following:

1. Issue `display` commands with the correct syntax.
2. Interpret the results of a `display` command.

Overview

Use the `display` commands to view information that is relevant to the Brick you are accessing or the Brick zone rulesets that are configured for the Brick .

For example, you can display a Brick ARP or MAC table, port statistics, and static routes that have been configured for the Brick . With the `display` commands, you can also view information such as a Brick zone ruleset policy, hostgroup definitions, or service group definitions.

The commands in this chapter are listed in alphabetical order. For each command, the chapter provides an overview, description, and explanation of the format, and examples.

To see a complete listing of the display commands online, enter `help display` at the command prompt.

Contents

display arptable	10-3
display clientpolicy	10-4
display configuration	10-5
display encapsulation	10-7
display failover	10-8

display files	10-9
display hostgroups	10-11
display icm	10-12
display interfacestatus	10-13
display lantolantunnels	10-16
display lsms	10-17
display mactable	10-18
display mempools	10-19
display nat	10-20
display partitions	10-21
display policy	10-22
display remoteconsole	10-24
display routes	10-25
display sa	10-27
display servicegroups	10-28
display sessions	10-29
display slamon	10-31
display time	10-32
display vlans	10-33
display zonetable	10-34

display arptable

Overview

The `display arptable` command displays the contents of the Address Resolution Protocol (ARP) table. Every time the Brick needs to resolve an IP address, an ARP request is issued and an entry is written to the ARP table.

Each entry in the ARP table contains the IP address that requires resolution, the MAC address to which the ARP request is sent, and the status of the request (OK or Wait).

Format

The format of the `display arptable` command is:

`display arptable`

Explanation

The `display arptable` command provides additional information that cannot be displayed on the Lucent Security Management Server (LSMS).

You may want to issue this command if the Brick has been configured with static routes and you want to examine the MAC addresses of the associated routers.

A status of ″Wait″ indicates the ARP request is pending. Note that a status of ″Wait″ for the IP Address of the LSMS or the default gateway indicates a connectivity problem. These connections should return status information immediately.

Example

The following is an example `display arptable` command:

```
hr-brick1>display arptable
IP Address      MAC Address        Status
125.92.38.40     0:60:97:e:b2:a1    OK
125.92.38.10     8:0:20:8:6d:98    OK
125.92.38.62     0:60:8:3a:54:11    OK
125.92.38.245    0:10:4b:de:f6:46  OK
Total ARP entries:  4
```

□

display clientpolicy

Overview

The `display clientpolicy` command displays the tunnel policy on the Brick.

Format

The format of the `display clientpolicy` command is:

`display clientpolicy <zone>`

Example

The following is an example of the `display clientpolicy` command:

```
display clientpolicy vpnzone
TEP, DHGRP, PH1Enc, PH1Auth, Proto, PH2Enc, PH2Auth, pfs, comp,
  count
20.20.19.100, Group 1, 3des, sha1, ESP, 3des, sha1, None, None, 10
```

Explanation

The following table summarizes the output fields for this command.:

Remote Tep	Remote Tunnel Endpoint
DH	Diffie-Hellman Group
PH1Enc	ISAKMP Encryption Type
PH1Auth	ISAKMP Authorization Type
Proto	Protocol
PH2Enc	IPSec Encryption Type
PH2Auth	IPSec Authorization Type
PFS	Perfect Forward Secrecy
Comp	Compression
count	Total number of Clients connected

display configuration

Overview

The `display configuration` command displays the contents of the *inferno.ini* file.

This file contains configuration information, such as the Brick name, that was entered in the Lucent Security Management Server (LSMS) graphical user interface (GUI).

Format

The format of the `display configuration` command is:

`display configuration`

Explanation

You may want to issue this command if you need to know the IP address, the number of ports and their IP addresses and network masks, and other characteristics of the Brick.

Note that the Remote Login ID is an encrypted string.

The following table explains VLAN- and Brick failover-related fields:

Field	Values	Explanation
`dom`	Blank if no VLAN assigned, or a letter from a to m	VLAN domain
`pdef`		Port default VLAN ID
`mbr`		Port VLAN membership
`fmt`	u = untagged q = 802.1Q p = preserve a = any d = 802.1Q except default	Format: two letter pair for receive and transmit
`failover`	yes/no	Indicates member of Brick failover pair
`act`	Value in tenths of a second	Brick failover activation Time
`yield`	Value in tenths of a second	Brick failover yield time
`stshint`	auto or port number (0 - 10)	Preferred state sharing link for failover pair

Field	Values	Explanation
`vi<n>`	<n> is a sequential number starting at 0. One per subnet. vid=VLAN ID	VLAN IP assignment
`vp<n>`	<n> is a port number (0 - 10)	VLAN configuration for port<n>

Example

The following is an example `display configuration` command:

```
hr-brick1>display configuration

r=10.10.10.4
ether3=type=i82557
vp3=dom= pdef=1 mbr= fmt=uu
ether2=type=i82557
vp2=dom= pdef=1 mbr= fmt=uu
ether1=type=i82557
vp1=dom= pdef=1 mbr= fmt=uu
ether0=type=i82557
vp0=dom= pdef=1 mbr= fmt=uu
vi0=vid=1 ip=10.10.10.4/24 pvt=n
gateway=gwip=
nics=4
admin=addr=10.10.10.10
vgc=10.10.10.10
RemoteLoginId=91e6a6dbc0004d1c54f0494610414adfb4c5ad45
audit=addr=10.10.10.10
auditwait=no
fwname=brick_one
failover=yes
act=4
yield=15
stshint=auto
```

□

display encapsulation

Overview

The `display encapsulation` command displays information about any enabled LAN-LAN VPN tunnels or client tunnels that are using UDP encapsulation to ″disguise″ their IPSec packets as they traverse through network devices that might otherwise block them.

For more information on UDP encapsulation, please refer to *Chapters 11 and 12* in the *LSMS Policy Guide*.

Format

The format of the `display encapsulation` command is:

`display encapsulation <zone>`

where *<zone>* is the name of the brick zone ruleset that has at least one tunnel using UDP encapsulation.

Explanation

The ″Target Host″ is the IP address or host group available at the far end of the tunnel. The ″Encap Hdr Dst IP″ is the next hop (router address, VBA,etc.) for the packet. The ″Encap Hdr Srv″ is the protocol, source port and destination port used by the encapsulated packet.

Example

The following is an example of the `display encapsulation` command:

```
test_brick> display encapsulation vpnzone
 Target Host                        Encap Hdr DstIP Encap Hdr Srv   Ref
  TagValue
 ----------------------------- -------------- ------------ --- --------
 10.20.30.153                       135.92.38.209   17/30241/501   2
  10054
```

□

display failover

Overview

The `display failover` command displays the current state of a failover Brick pair

Format

The format of the `display failover` command is:

```
display failover
```

Explanation

The display failover command lists the two members of the failover pair by Brick ID, the last four digits of the MAC address. The State column shows each Brick status, either Active or Standby. If the Standby cannot be contacted, as might be the case in an actual failover situation, the message ″NO Standby″ is displayed.

Example

The following is an example `display failover` command:

```
hr-brick1>display failover
Bricks            State          Last Pri            Pwr Id-MAC
=>d3d5          ACTIVE          N/A                84 00a0c9-
  8bd3d5(THIS brick)
  68fc           standby        -7               20 02e03b-0068fc
Links   Status     Active-MAC          Physical-MAC
ether0 rcving    020000-28a338       00a0c9-8bd3d5
ether1 verified   020000-30b304       00a0c9-8bd3f0
ether2 down      020000-bf6e96       00a0c9-8bd408
```

□

display files

Overview

The `display files` command displays the sizes, dates modified, and filenames of all files resident on the Brick.

Format

The format of the `display files` command is:

```
display files
```

Explanation

The display files command lists all files stored in the Brick flash memory and downloaded from the LSMS. The files b.com, tvpc, authinfo, and inferno.ini are copied from floppy when the Brick is created.

Example

The following is an example `display files` command:

```
test_brick> display files
 Size Date (GMT) Name
 0 Jun 14 05:14 bin/
 0 Jun 14 05:14 dev/
 0 Jun 14 05:14 net/
 0 Jun 14 05:14 nvfs/
 0 Jun 14 05:14 prog/
 0 Jun 14 05:14 n/
 0 Jun 14 05:14
 58672 Jun 14 05:14 osinit
 10200 Jun 14 05:14 dosfs
 14560 Jun 14 05:14 dossubs
 4408 Jun 14 05:14 iotrack
 9800 Jun 14 05:14 styx
 2112 Jun 14 05:14 mytime
 66912 Jun 14 05:14 commands
 35804 Jun 13 17:14 b.com
 924 May 08 15:38 authinfo
 493 Jun 13 17:15 inferno.ini
 1241286 Jun 14 14:27 tvpc
 0 Jun 14 14:36 policy/
 0 May 08 15:41 dump.txt
 6 May 08 15:41 tzoffset
 521332 Jun 14 14:28 hspbin
```

□

display hostgroups

Overview

The `display hostgroups` command displays the current set of hostgroup definitions for the specified Brick zone ruleset.

Format

The format of the `display hostgroups` command is:

`display hostgroups <zone>`

where *<zone>* is the name of the Brick zone ruleset that contains the hostgroups.

Explanation

The first column is the name of the hostgroup.

The second column contains the type: normal or dynamic. Normal hostgroups are those that are manually configured to be part of a policy. Dynamic hostgroups are those that are temporary, and are used primarily when the Brick is authenticating users.

The third column contains the IP address or range of IP addresses associated with the hostgroup.

Each IP address or range is printed on its own line.

Example

The following example shows two `display hostgroups` commands:

```
hr-brick1>display hostgroups vpnhostileclient
Hostname            Type     IP Range
privateservers     Normal   10.92.11.10 - 10.92.11.49
hostileclients     Normal   10.92.10.70 - 10.92.10.129
hr-brick1>display  hostgroups administrativezone
Hostname            Type    IP Range
bricks             Normal   10.92.11.10 - 10.92.11.49
hostileclients     Normal   123.34.23.34 - 123.34.45.56
```

□

display icm

Overview

The `display icm` command is used to display session statistics as the Brick uses the ″Intelligent Cache Management″ feature. The feature monitors the cache usage in the Brick to ensure that its resources are not overwhelmed by excessive traffic such as might be seen in the event of an attack.

For more information on this feature, please refer to *Chapter 4 Configuring Brick Ports* in the *LSMS Administration Guide*.

Format

The format of the `display icm` command is:

```
display icm
```

Explanation

A summary is provided as shown below that lists such items as whether the feature is activated on the Brick, the setting for the trigger threshold, as well the number and protocol of the packets passed through the Brick.

Example

```
test_brick> display icm

Current ICM config (disabled). Rpt=0s. Maxidx= 7 flags=00000002
Floor: 65%= 21810360 bytes. Trigger: 80%= 26843520 bytes, Cur: 6528
  bytes
Class name ID DAH Service Pct Bytes Us% Used bytes Sessions
  Unprunable
---------- -- --- --------- --- ---------- --- ---------- -------- ----------
 Other 0***      */*/* 0      0 0   2112     8      0
Drop_Unaud 1 ynn     */*/* 0    0 0      0      0    0
Drop_Audit 2 yyn    */*/* 0    0 0     0       0     0
   ICMP 3 n*n     1/*/* 15   5033160 0    576    2      0
    UDP 6 ***    17/*/* 25   8388600 0    3584   14      0
  TCP_SYN 5 n*y      6/*/* 45   15099480 0    0    0    0
(Residual) 4 n**   17/*/* 20   6710880  0   256   1    0
```

□

display interfacestatus

Overview

The `display interfacestatus` command displays relevant information about traffic that flows through a specified port since the Brick was rebooted last.

You may want to issue this command if you suspect a hardware problem and need to investigate patterns of excessive error messages.

This information is highly dependent upon the make and model of the NIC and the Inferno device driver associated with it. The following two tables are specific to Intel 82557 related cards.

Format

The format for the `display interfacestatus` command is:

```
display interfacestatus <interface#>
```

where:

- *<interface#>* is the number of the port (0 through 11).

If you issue the command without the <interface#> argument, it will return a brief summary of all ports on the Brick.

Explanation

The status reported per port is grouped according to transmit statistics and receive statistics.

Transmit

The following information is printed for transmit (TX):

Good frames	This counter contains the number of frames that were transmitted properly on the link. It is updated only after the actual transmission on the link is completed, and not when the frame was read from memory as is done for the TxCB status.
Exceeded maximum collision errors	This counter contains the number of frames that were not transmitted since they encountered the configured maximum number of collisions.
Late collision errors	The number of collisions detected later than 512 bit-times (i.e. 51.2 microseconds for 10BaseT) into the transmission of the frame.

Underrun errors	The number of frames, which failed to be transmitted or were retransmitted because the system bus (via the DMA) did not keep up with the transmission.
Lost carrier sense	The number of times the carrier sense was lost during transmission.
Deferred frames	The number of frames initially delayed due to activity on the link.
Single collisions	The number of successfully transmitted frames that experienced exactly one collision
Multiple collisions	The number of successfully transmitted frames that experienced multiple collisions
Total collisions	This counter contains the total number of collisions that were encountered while attempting to transmit.

Receives

The following information is printed for receives:

Good frames	The number of good frames received.
CRC errors	This counter contains the number of aligned frames discarded because of a CRC error. This counter increments only once per receive frame. It is mutually exclusive to alignment and short-frame errors.
Alignment errors	This counter contains the number of frames that are both misaligned (i.e., where CRS deasserts on a nonoctal boundary) and contain a CRC error. It is mutually exclusive to the CRC error and short-frame counters.
Resource errors	This counter contains the number of good frames discarded because there were no resources available. Frames intended for a host whose RU is in the No Resources state fall into this category. If the 82557 is configured to Save Bad Frames and the status of the received frame indicates that it is a bad frame, this counter is not updated.
Overrun errors	This counter contains the number of frames known to be lost because the local system bus was not available. If the traffic problem persists for more than one frame, the frames that follow the first are also lost; however, because there is no lost frame indicator, they are not counted.

Collision detect errors	This counter contains the number of frames that encountered collisions during frame reception.
Short frame errors	This counter contains the number of received frames that are shorter than the minimum frame length. It is mutually exclusive to the alignment and short-frame error counters and has a higher priority (i.e., a short frame will always increment only the short-frame counter).

Example

The following is an example `display interfacestatus` command for ether 0:

```
hr-brick1>display interfacestatus 0
Interface 0; Device Driver i82557
TX: good frames:                            5415
   exceeded max collision errors:              2
   late collision errors:                      0
   underrun errors:                            0
   lost carrier sense:                         0
   deferred frames:                          657
   single collisions:                        214
   multiple collisions:                      394
   total collisions:                        1938
RX: good frames:                         1070756
   CRC errors:                              1486
   alignment errors:                        1213
   resource errors:                            0
   overrun errors:                              0
   collision detect error s                     0
   short frame errors                      82325
```

The following example of `display interfacestatus` (with no port number) shows a summary of all the interfaces:

```
test_brick> display interfacestatus
Interface    MAC              Link    Speed      Mode
ether0    0:90:27:16:5b:93  Up     100 Mbps   Full Duplex
ether1    0:90:27:16:59:63  Up     100 Mbps   Full Duplex
ether2    0:90:27:16:3e:f3  Up     10 Mbps    Half Duplex
ether3    0:90:27:16:50:35  Down   -          -
```

☐

display lantolantunnels

Overview

The `display lantolantunnels` command displays the configured LAN-to-LAN tunnel policy on the Brick.

Format

The format of the `display lantolantunnels` command is:

`display lantolantunnels` *<zone> <tunnel name>*

Example

The following is an example of the `display lantolantunnels` command:

```
display lantolantunnels vpnzone nyc-atl-tunnel
Remote Tep, DH, PH1Enc, PH1Auth, Proto, PH2Enc, PH2Auth, Pfs, Comp,
   Enable, Status
20.20.19.100, Group 1, 3des, sha1, ESP, 3des, sha1, No, No, Enabled,
   UP
```

Explanation

The following table summarizes the output fields for this command.:

Remote Tep	Remote Tunnel Endpoint
DH	Diffie-Hellman Group
PH1Enc	ISAKMP Encryption Type
PH1Auth	ISAKMP Authorization Type
Proto	Protocol
PH2Enc	IPSec Encryption Type
PH2Auth	IPSec Authorization Type
PFS	Perfect Forward Secrecy
Comp	Compression
Enable	Enable/Disable tunnel status
Status	Tunnel Status (UP/DOWN)

□

display lsms

Overview

The `display lsms` command returns the IP address of the LSMS to which the Brick is homed.

Format

The format of the `display lsms` command is:

`display lsms`

Explanation

Use display LSMS to determine which LSMS of a redundant pair to which the Brick is homed.

Example

The following is an example `display lsms` command:

```
display lsms
Last LSMS was 10.10.10.5
```

□

display mactable

Overview

The `display mactable` command displays the contents of the Media Access Control (MAC) table for a specified port.

MAC addresses are hardware addresses that are hard-coded in all network interface cards. The MAC table tracks the MAC addresses of all MAC elements that are associated with each network interface.

Format

The format of the `display mactable` command is:

`display mactable` *<port#>*

where *<port#>* is a port number (0 through 10)

The command syntax is as follows:

- If the port number is missing, the MAC table is displayed for all ports.
- If the port number is specified but it is not available, an empty table is displayed.

Explanation

Each entry in the MAC table contains the port of the MAC element, the MAC address of the element, and the status (OK, Old, or Unavail).

A status of *Old* means that the address needs to be refreshed or is in the process of being refreshed.

A status of *Unavail* means that the port is down.

The VLAN column displays the VLAN ID on which the MAC address resides. The Brick MAC addresses are present on every VLAN attached to the Brick. Their VLAN ID is displayed as ″*″.

Example

The following is an example `display mactable` command :

```
hr-brick1>display mactable 0

IF      MAC Address      Status VLAN
local0 0:60:8:c1:91:40 OK      *
ether0 0:a0:d1:3:21:80 OK      1
local1 0:a0:d1:3:1a:41 OK      *
```

display mempools

Overview

The `display mempools` command displays the memory usage information (in bytes) of the main, heap, image, and Brick session cache pools.

For each pool it displays:

- Maximum allocated size in bytes
- Currently allocated size in bytes
- Peak (high water mark) size in bytes,
- Arena size in bytes
- The difference between the number of allocations versus frees.

Format

The format of the `display mempools` command is:

`display mempools`

Explanation

The `display mempools` command provides additional information that cannot be retrieved on the Lucent Security Management Server.

If the Peak Memory size is persistently approaching the Maximum Memory size, it may be an indicator that the Brick is overloaded and not performing efficiently. You may need to acquire additional Bricks for load balancing the traffic.

Example

The following is an example `display mempools` command

```
hr-brick1>display mempools
Pool      Max-Size      Cur-Size      Peak    Arena-Sz   In-Use
Main      20971520      1141472  1296736   1573296   1519
Heap      4194304     194720     210368     262216     1469
Image      1048576     0          0           0         0
SCache     8388608     768        3840     131108     4
```

□

display nat

Overview

The `display nat` command shows any rules using network address translation (NAT) with the ″direct″ type option enabled.

For more information on NAT, please refer to *Chapter 6 Network Address Translation* in the *LSMS Policy Guide*.

Format

The format of the `display nat` command is :

`display nat <zone>`

where *<zone>* is the Brick zone ruleset that has rules configured with NAT and the ″direct″ type specified under the Address Translation tab for the rule.

Explanation

This command will only display NAT entries where the rule(s)uses direct NAT. Direct NAT implies that there are two hostgroups defined, each with the same number of hosts. One group is the ″inside″ private list of IP addresses (Pre-NAT list)and it is mapped to the other group of ″outside″ routable IP addresses (Post-NAT list).

Example

The following is an example of the `display nat` command:

```
test_brick> display nat administrativezone
 Name           RefCt Pre-NAT list                Post-NAT list
 ------------- ----- ------------------------   -------------------------
 Rule_201_2         1 brickLocalAddresses
  brickRemoteAddresses
 Rule_200_3         1 brickRemoteAddresses       brickLocalAddresses
```

□

display partitions

Overview

The display partitions command will show any VLAN partitions defined for a given Brick.

For more information on Brick partitions, refer to the *LSMS Administration Guide*.

Format

The format for this command is simply:

display partitions

Explanation

If the Brick is ″VLAN enabled″, you have the option of creating Brick partitions. A partition allows the Brick to properly route packets from a VLAN using the same private IP address space as another customer or department. The command output lists the partition number and the VLAN IDs associated with it.

Example

The following is an example of the display partitions command:

```
test_brick> display partitions
Partition   VLAN IDs
*Default    *(local)
```

display policy

Overview

The `display policy` command displays the current ruleset for the specified Brick zone ruleset.

Format

The format of the `display policy` command is:

`display policy` *<zone> [dyn]*

where

- *<zone>* is the Brick zone ruleset that contains the policy you want to display. If the Brick zone ruleset was not loaded on the Brick, an error is displayed.
- [dyn] is an optional keyword that can be specified. This keyword will display any dynamic rules of a Brick zone ruleset security policy at the time the command was issued.

Dynamic rules are those that are temporary and used to allow traffic through the Brick for VPN and FTP traffic, and when traffic is forwarded to a proxy server.

Explanation

The output format is an abbreviated list of the Brick zone ruleset current rules, including the load date, the signing date and the name of the signing administrator. The following items are displayed:

- Rule number
- Source host IP address – either an IP address or a hostgroup
- Destination IP address – either an IP address or the name of a hostgroup
- Service – protocol/src-port/dst-port or name of service group
- A – action: + (pass), - (drop), y (proxy)
- D – direction: i (in), o (out), b (both)
- SM – source NAT enabled (source mapping)
- DM – destination NAT enabled (destination mapping)
- PM – port NAT enabled (port mapping)
- DEP – dependency mask enabled
- VPN – VPN enabled

The Signer at the bottom of the display is the last administrator who loaded the security policy.

Example

```
hr-brick1>display policy firewall

Rule#  Source  Destination Service                    A D SM DM PM
  DEP VPN
200      *      LSMS        */*/*                      + o no no no
  no no no
201      LSMS  *            brick_from_SMS_Services   + i no no no
  no
202      *      *            1/3/13                     + o no no no
  no no no
203      *      *            17/500/*                   + i no no no
  no no
204      *      *            17/*/500                   + o no no no
  no no
205      *      *            17/1024/*                  + i no no no
  no no
65535  *      *            */*/*                      - b no no no
  no no no
Load Date: Mar 21 10:08:13 2001
Sign Date: Dec 02 17:30:26 2000
Signer: Admin
```

display remoteconsole

Overview

The `display remoteconsole` command reports the state of the remote Brick console if it is connected and the LSMS Administrator ID associated with that remote console session.

Format

The format of the `display remoteconsole` command is:

```
display remoteconsole
```

Explanation

You can use `display remoteconsole` to from a local Brick console connection to determine if another administrator is connected to the Brick via a remote console session. The command also returns the AdminID of the administrator.

Example

The following is an example `display remoteconsole` command:

```
display remoteconsole
User Admin_22 is connected through remote console.
```

□

display routes

Overview

The `display routes` command displays the set of static routes configured for the Brick.

If a Brick is configured with static routes, it can send traffic to LAN segments that are not connected to the Brick. The routes determine how the Brick routes traffic that are not destined for a local LAN segment.

Format

The format of the `display routes` command is:

`display routes` ***`[<interface# | #>]`***

where ***`[<interface# | #>]`*** can be:

- A port number (0 through 11).
  The command syntax is as follows:
  - If a port number is missing, the routes are displayed for all ports.
  - If a port number is specified, but it is not available, an empty table is displayed.
- #

If # is specified, the routes are displayed for the local port (the firewall Brick zone ruleset).

Explanation

System Administrators can use the LSMS to create and maintain a static routing table. Use the `display routes` command to display the routes that were set in the LSMS graphical user interface.

Each entry in the static routing table contains the port number (IF), the destination IP address, the mask, and the gateway.

The interface column (IF); gives the Brick port of the gateway for the route (example: ether0). If the gateway is a on VLAN other than the default VLAN for that interface, the VLAN number is suffixed to the port number, for example, ″ether0.99.″

Example

The following is an example `display routes` command:

```
hr-brick1>display routes

IF      Destination   Mask             Gateway
ether3 0.0.0.0        255.255.255.0   192.168.14.2
local   192.168.14.0   255.255.255.0    0.0.0.0
local   192.20.250.0   255.255.255.0    0.0.0.0
local   192.168.20.0   255.255.255.0    0.0.0.0
local   192.168.15.0   255.255.255.0    0.0.0.0
ether1  192.168.40.0   255.255.255.0    192.168.20.10
ether1 192.168.50.0   255.255.255.0    192.168.20.10
ether3 135.92.38.0     255.255.255.0    192.168.14.1
```

In this example, 192.168.14.2 is the default gateway for this Brick, set in the Brick Editor Brick tab **Gateway IP Address** field, and is noted as default route ″0.0.0.0″ in the display routes output.

☐

display sa

Overview

The `display sa` command displays the current set of Security Associations (SA) for the specified Brick zone ruleset.

Format

The format of the `display sa` command is:

`display sa <zone>`

where:

- *<zone>* is the Brick zone ruleset that contains the current set of Security Associations you want to display. If the Brick zone ruleset was not loaded on the Brick, an error is displayed.

Explanation

In LAN-LAN VPNs that use Internet Key Exchange (IKE), the administrators of the Brick zone rulesets at both ends of the VPN must manually create and negotiate Security Associations (SAs) to specify the type of authentication and encryption to be used in the VPN.

Use the `display sa` command to display the security associations for VPN Key Exchange that were created using the LSMS graphical user interface. The output consists of nine free-form columns.

Example

The following is an example `display sa` command:

```
hr-brick1>display sa vpnhostileclient

SPI User Name Source   Destination  Prot AH  ESP TEP      Sec/KBytes
4096 efg      *       125.92.38.100 esp md5 des 125.92.   0/0
                                                    10.241
```

□

display servicegroups

Overview

The *display servicegroups* command displays the current set of service group definitions that are currently in use for a given Brick zone ruleset.

A listing of all the default service groups is not included in the display.

Format

The format of the `display servicegroups` command is:

`display servicegroups <zone>`

where

- *<zone>* is the Brick zone ruleset that contains the current set of service group definitions you want to display. If the Brick zone ruleset was not loaded on the Brick, an error is displayed.

Explanation

The first column contains the name of the service. The second column contains the definitions of the service.

This data can also be displayed on the LSMS.

Example

The following is an example `display servicegroups` command:

```
hr-brick1>display servicegroups administrativezone

Service Name Definitions
brickServices tcp/9000-9004/*
       tcp/900/*
       tcp/*/910
```

□

display sessions

Overview

The `display sessions` command displays the current session cache for the specified Brick zone ruleset.

Format

The format of the `display sessions` command is:

```
display sessions <zone>
[<ipfilter>]
```

where:

- *<zone>* is the Brick zone ruleset that contains the current session cache you want to display. If the Brick zone ruleset was not loaded on the brick, an error is displayed.
- *[<ipfilter>]* is an optional argument specifying an IP address that matches either the source or destination IP addresses.

Explanation

The output format is an abbreviated list of the Brick zone ruleset current sessions.

The ″A″ (action) column can be ″+″ for pass, ″-″ for drop and ″y″ for proxy. (PKT) and bytes (B) coming into the Brick zone ruleset (FWD), going out of the Brick zone ruleset (REV), (V) indicates a VPN, and (E) represents UDP VPN Encapsulation.

Example

The following is an example `display sessions` command:

```
hr-brick1>display sessions administrativezone
Source          Destination Service         A Rule# FWD-PKT/B
  REV-PKT/B
135.92.38.220     135.92.38.40   6/9001/24505 + 1      225/
  16228      134/5540
135.92.38.40       135.92.38.220   6/910/1032   + 2      1/
  40         0/0
135.92.38.76       135.92.38.255   17/137/137   - 65535 39/
  3042        0/0
135.92.38.251     135.92.38.255   17/138/138   - 65535 1/
  248         0/0
135.92.38.79       135.92.38.255   17/138/138   - 65535 1/
  229         0/0
135.92.38.246     135.92.38.255   17/138/138   -  65535 1/
  248         0/0
135.92.38.         255.255.255.25   9/0/0        - 65535 2/
  2416          0/0
```

□

display slamon

Overview

If you are a service provider and you are guaranteeing certain minimum service levels through a Service Level Agreement (SLA), you can activate the ″Enable SLA Probe″ option for LAN-LAN tunnels.

For information on this feature, please refer to *Chapter 11 LAN-LAN Tunnels* in the *LSMS Policy Guide*.

Format

The format of the `display slamon` command is:

`display slamon <zone>`

where `<zone>` is the Brick zone ruleset on the Initiator Brick where the SLA probe has been enabled.

Explanation

The command output shows the Probe Source and Destination VBAs as well as the number of probes sent by the source and received by the destination.

Note that while there are two endpoints in a LAN-LAN tunnel, the display slamon command will only provide results when executed on the Initiator Brick (usually Endpoint 1) in the tunnel.

Example

The following is an example of the `display slamon` comand:

```
test_brick> display slamon vpnzone
Probe ID        Probe Src         Probe Dst     Sent Recvd    MaxRTT
-------- -------------- -------------- ---- ----- ------
   10710       10.20.10.133      10.20.10.77      8    8      1
```

□

display time

Overview

The `display time` command displays the current time on the Brick relative to Greenwich Mean Time (GMT).

Format

The format of the `display time` command is:

`display time`

Explanation

When the Brick boots, the LSMS machine sends the Brick its own time indexed to the GMT. The Brick requests an update every hour to ensure that the pair's time is synchronized to the second.

Example

The following is an example `display time` command:

```
test_brick> display time
The current time is Jun 14 2002, 19:10:02 GMT
Last booted at Jun 14 2002, 14:31:24. Active since
Jun 14 2002, 14:31:24.
LSMS is at -4:00 from GMT.
Brick local time is -4:00 from GMT.
```

□

display vlans

Overview

The `display vlans` command displays a list of all VLANs on the Brick.

Format

The format of the `display time` command is:

```
display vlans
```

Explanation

The `display vlans` command lists all VLANs with their IP address, subnet mask, and member ports, including default ports.

Example

The following is an example `display vlans` command:

```
hr-brick1>display vlans
VLAN IP Address  Mask            Ports
1     10.10.1.1   255.255.255.0    1
2     10.10.67.2  255.255.255.0    1-2
100   10.10.44.2  255.255.255.0    0-2
a.7              255.255.255.192  3
```

☐

display zonetable

Overview

The `display zonetable` command prints the Policy Assignment Table of the Brick, including the load date, the signing date and the signing administrator.

Format

The format of the `display zonetable` command is:

display zonetable

Explanation

For each Brick, any applied Brick zone ruleset is associated with one or more physical ports. A ruleset can be assigned to more than one port on the same brick, or on different Bricks.

This information is then stored in the Policy Assignment Table, which the Brick uses to determine which security policy to apply to incoming packets. The Signer at the bottom of the display is the last administrator who loaded the Policy Assignment Table.

Example

The following is an example `display zonetable` command:

```
hr-brick1>display zonetable

Ifc     Address Range    Zone   VLAN  VBA
e0     *                *        *
e1     *                *        *
e2     *                *        *
e3     *                *        *
lcl  firewall
Load Date: Thu June 21 11:26:18 2002
Sign Date: Fri Jue 15 16:17:56 2002
Signer:    Admin
Softw vers: 7.2.444; VPN card n; Status: active
Starcast zone: n; MAC moves: y
```

□

11 Lucent VPN Firewall *Brick*® Device Trace Commands

Overview

Purpose

This chapter provides information to perform the following:

1. Issue `trace` commands with the correct syntax.
2. Interpret the results of a `trace` command.

Overview

Use the `trace` commands to control the filtering and display of real-time log records and incoming and outgoing packets as they are processed by the Brick.

For example, you can create a filter to display packets that originate from a specific IP address or zone. Or, a filter can be created to display log records that logged dropped packets only.

> **Important!** *PERFORMANCE CONSIDERATIONS* For performance reasons, it's best to issue the `trace audit on` and `trace packet on` commands one at a time. If both commands are issued at the same time, network performance may be severely hampered.
>
> Trace packet is not for use in a live network. The use of trace packet may cause the Brick CPU to increase and may result in drop packets. Trace packet should be used in a controlled environment such as a maintenance window.

The commands in this chapter are listed in alphabetical order. For each command, the chapter provides an overview, a description and explanation of the format, and examples.

To see a complete listing of the display commands online, enter `help trace audit` or `help trace packet` at the command prompt.

Contents

trace arp	11-3
trace audit delete	11-4
trace audit filter	11-5
trace audit list	11-7
trace audit modify	11-8
trace audit off	11-10
trace audit on	11-11
trace nonip	11-12
trace packet delete	11-13
trace packet filter	11-14
trace packet list	11-16
trace packet modify	11-17
trace packet off	11-19
trace packet on	11-20

trace arp

Overview

The `trace arp` command turns on or off the tracing of Address Resolution Protocol (ARP) packets on the brick console.

Format

The format of the `trace arp on` command is:

```
trace arp <on|off> [yes|no}
```

where:

The *<on|off>* argument turns trace arp on or off. Issued without the on or off argument, trace arp toggles the current state.

The optional *[yes|no]* argument refers to echo of traced packets to the screen.

Explanation

The `trace arp` command enables the administrator to see Address Resolution Protocol requests arriving at the Brick ports.

Example

The following is an example `trace arp` command:

```
trace arp on
tracing of arp enabled
```

□

trace audit delete

Overview

The `trace audit delete` command deletes a filter that was defined with the `trace audit filter` command.

Format

The format of the `trace audit delete` command is:

`trace audit delete <id>`

where:

- *<id>* is the number of the filter you wish to delete. To see the list of filters that can be deleted, *issue the* `trace audit list` *command*.

Explanation

If the filter does not exist, no action is taken.

Example

The following is an example `trace audit delete` command:

```
hr-brick1>trace audit delete 1

filter 1 deleted.
```

□

trace audit filter

Overview

The `trace audit filter` command allows you to define a filter that will display log records that match only the fields you specified in the filter.

A filter is defined by a filter list of six possible values:

- Zone
- Source IP address
- Destination IP address
- Protocol
- Direction
- Action

After defining a filter, enable the filter and start viewing the records by issuing the `trace audit on` command. Ensure that the `set printing` command is set to `on`.

Format

The format of the `trace audit filter` command is:

`trace audit filter <filter_list>`

where *<filter_list>* is one or more of the following parameters:

Parameter	Explanation	Example
z	Zone name	z=administrativezone
s	Source IP address	s=104.23.32.123
d	Destination IP address	d=123.34.134.2
p	Protocol/dst-port/src-port	p=17/138/1137
r	Direction ("in" or "out")	r=in
a	Action ("drop", "pass" or "proxy")	a=drop

Explanation

A wildcard (*) matches everything for the specified parameter. Parameters that are not defined are assumed to be a wildcard (*).

There is a limit of 20 audit filters. In addition, the zone name can be abbreviated and is completed from the list of known zones after the command is entered.

Example

The following is an example `trace audit filter` command:

```
hr-brick1>trace audit filter z=adm
filter 1 defined: z=administrativezone,s=*,d=*,p=*/*/*, r=*,a=*
```

□

trace audit list

Overview

The `trace audit list` command displays the currently defined filters, as defined with the `trace audit filter` command.

Format

The format of the `trace audit list` command is:

`trace audit list`

Explanation

This command is particularly useful to identify the filters when modifying, deleting, enabling, or disabling a filter.

In the output, the ″E″ column specifies whether the filter is enabled or not. The other columns are defined in the `trace audit filter` command description.

Example

The following is an example `trace audit list` command:

```
hr-brick1>trace audit list
# E Zone              Source       Destination Protocol Dir A
1 Y administrativezone*            *           */*/*    i   *
2 N *                 104.23.34.24 *           */*/*    *   *
```

□

trace audit modify

Overview

The trace `audit modify` command modifies a filter that was defined with the `trace audit filter` command.

All six possible values of a filter list — zone, source IP address, destination IP address, protocol, direction, and action, can be modified.

Format

The format of the `trace audit modify` command is:

`trace audit modify` *`<id> <filter_list>`*

where:

- *`<id>`* is the number of the filter you wish to modify. To see the list of filters that can be modified, *issue the* `trace audit list` *command*.
- *`<filter_list>`* is one or more of the following parameters:

Parameter	Explanation	Example
`z`	Zone name	z=administrativezone
`s`	Source IP address	s=104.23.32.123
`d`	Destination IP address	d=123.34.134.2
`p`	Protocol/dst-port/src-port	p=17/138/1137
`r`	Direction (″in″ or ″out″)	r=in
`a`	Action (″drop″, ″pass″ or ″proxy″)	a=drop

Explanation

If a parameter is present, then it replaces the value in the definition. If the parameter is not present, then the definition is unchanged. Specifying a wildcard matches everything for the specified parameter.

In addition, the zone name can be abbreviated and is completed from the list of known zones after the command is entered.

Example

The following is an example `trace audit modify` command:

```
hr-brick1>trace audit modify 1 r=in, a=*

filter 1 modified.
```

□

trace audit off

Overview

The `trace audit off` command stops the display of audit records for all filters or a particular filter.

Before a filter can be disabled, it must have been enabled with the `trace audit on` command.

Format

The format of the `trace audit off` command is:

`trace audit off [<id> |a |p]`

where:

- *<id>* is the number of the filter you wish to disable. To see the list of filters that can be disabled, *issue the* `trace audit list` *command*. If an <id> is not present, then all defined session filters are disabled.
- *a* disables the tracing of all Administrative Events audit records.
- *p* disables the tracing of the Proactive Monitoring records.

Explanation

Note that this command does not remove the filter but just stops the display of the audit records.

Example

The following is an example `trace audit off` command:

```
hr-brick1>trace audit off

all filters disabled.
```

□

trace audit on

Overview

The `trace audit on` command enables the filter and starts the display of the audit records. The filter can be disabled, and the display of the audit records stopped, with the `trace audit off` command.

Due to performance reasons, the `trace audit on` and `trace packet on` commands should be used separately. If a trace packet filter has been enabled and you want to enable a trace audit filter, issue the `trace packet off` command.

Format

The format of the `trace audit on` command is:

`trace audit on [<id> |a |p]`

where:

- `<id>` is the number of the filter you wish to enable. To see the list of filters that can be enabled, *issue the* `trace audit list` *command*. If an <id> is not present, then all defined session filters are enabled.
- *`a`* permits the tracing of all Administrative Events audit records.
- *`p`* permits the tracing of the Proactive Monitoring records.

Explanation

The formats for audit records are the raw formats dumped directly to the logger. Notice that each line is preceded by an ″A>>″ to identify it as an audit record.

Example

The following is an example `trace audit on` command:

```
hr-brick1>trace audit on 1

filter 1 enabled.
A>>
   0:administrativezone:OUT:875c2628:875c26ff:17:138:138:Drop:ether0::0:65535:
A>>
   0:administrativezone:OUT:875c2614:875c26ff:17:137:137:Drop:ether0::0:65535:
A>>1:administrativezone:OUT:875c2628:875c26ff:17:138:138:1:0:233:0:0
A>>1:administrativezone:OUT:875c2614:875c26ff:17:137:137:2:0:156:0:0
```

□

trace nonip

Overview

The `trace nonip` command enables tracing of all non-IP packets.

Format

The format of the `trace nonip on` command is:

`trace nonip` ***[on|off] [y|n]***

where:

- The optional second argument [y|n]outputs the contents of the entire packet in hexadecimal format.

Explanation

The `trace nonip` command is used to debug non-IP packet flow issues. Input is shown whether the particular packet is bridged or not.

Example

The following is an example `trace nonip` command for an IPX packet received on port 0.

```
hr-brick1>trace nonip

0 non d=0601882166ad s=0000947545b4 ln=40 dsap=ff ssap=ff
ctl=000601882166ad0000947545b40028ffff002800110000face000000000001
```

☐

trace packet delete

Overview

The `trace packet delete` command deletes a filter that was defined with the `trace packet filter` command.

Format

The format of the `trace packet delete` command is:

`trace packet delete <id>`

where:

- *<id>* is the number of the filter you wish to delete. To see the list of filters that can be deleted, *issue the* `trace packet list` *command*.

Example

The following is an example `trace packet delete` command:

```
hr-brick1>trace packet delete 1

filter 1 deleted.
```

trace packet filter

Overview

The `trace packet filter` command defines a filter to display incoming and outgoing packets in real-time. The `trace packet filter` command does not display Address Resolution Protocol (ARP) packets. Use the `trace arp` command to trace ARP packets.

A filter is defined by a filter list of six possible values — interface, source IP address, destination IP address, protocol, direction, and format.

After defining a filter, you must enable the filter and start displaying the packets by issuing the `trace packet on` command.

Format

The format of the `trace packet filter` command is:

`trace packet filter` *<filter_list>*

where:

- *<filter_list>* is one or more of the following parameters:

Parameter	Explanation	Example
`i`	Brick port number (0-11) or # for local. If * is specified, the local port is not included.	i=2
`s`	Source IP address	s=104.23.32.123
`d`	Destination IP address	d=123.34.134.2
`p`	Protocol/dst-port/src-port	p=17/138/1137
`r`	Direction (″in″ or ″out″)	r=in
`f`	Format options (a,l,m, or c). `a`=dumps the hexadecimal form of the binary data of the packets matching the filter on the Brick local and remote consoles. `l`=packages the binary data in audit records and ships them to the LSMS. `m`=prints the hardware MAC addresses of the packet and the IP identification field. `c`=prints additional contents of the IP packet based on the protocol.	f=m

Explanation

A wildcard (*) matches everything for the specified parameter. Parameters which are not defined are assumed to be a wildcard (*).

For performance reasons, there is a limit of five packet filters.

Example

The following is an example `trace packet filter` command:

```
hr-brick1>trace packet filter i=1

filter 1 defined. i=1, s=*, d=*, p=*/*/*, r=in, f=*
```

□

trace packet list

Overview

The `trace packet list` command displays the currently defined filters as defined with the `trace packet filter` command.

Format

The format of the `trace packet list` command is:

`trace packet list`

Explanation

This command is particularly useful to identify the filters when modifying, deleting, enabling, or disabling a packet filter.

In the output, the ″E″ column specifies whether the filter is enabled or not. The ″IF″ column specifies the Brick port number. The other columns are explained in the `trace packet filter` command description.

Example

The following is an example `trace packet list` command:

```
hr-brick1>trace packet list

# E IF Source       Destination Protocol Format
1 N 0  *            *           */*/*    a
2 N *  104.23.34.24 *           */*/*    l
```

☐

trace packet modify

Overview

The `trace packet modify` command modifies a filter that was defined with the `trace packet filter` command.

All six possible values of a filter list — interface, source IP address, destination IP address, protocol, direction, and format can be modified.

Format

The format of the `trace packet modify` command is:

`trace packet modify <id> <filter_list>`

where:

- *<id>* is the number of the filter you wish to modify. To see the list of filters that can be modified, issue the `trace packet list` command.
- *<filter_list>* is one or more of the following parameters:

Parameter	Explanation	Example
i	Brick port number (0-11) or # for local. If * is specified, the local port is not included.	i=2
s	Source IP address	s=104.23.32.123
d	Destination IP address	d=123.34.134.2
p	Protocol/dst-port/src-port	p=17/138/1137
r	Direction (″in″ or ″out″)	r=in
f	Format options (a,l,m, or c). a=dumps the hexadecimal form of the binary data of the packets matching the filter on the Brick local and remote consoles. l=packages the binary data in audit records and ships them to the LSMS. m=prints the hardware MAC addresses of the packet and the IP identification field. c=prints additional contents of the IP packet based on the protocol.	f=m

Explanation

If a parameter is present, then it replaces the value in the definition. If the parameter is not present, then the definition is unchanged.

Specifying a wildcard matches everything for the specified parameter.

Example

The following is an example `trace packet modify` command:

```
hr-brick1>trace packet modify 1 r=in, f=*

filter 1 modified.
```

□

trace packet off

Overview

The `trace packet off` command disables and stops the display of audit records for all filters or a particular filter.

Before a filter can be disabled, it must have been enabled with the `trace packet on` command.

Format

The format of the `trace packet off` command is:

`trace packet off` ***`[<id>]`***

where:

- *`<id>`* is the number of the filter you wish to disable. To see the list of filters that can be disabled, *issue the* `trace packet list` *command*. If an <id> is not present, then all packet tracing is disabled.

Explanation

Note that this command does not remove the filter but just turns off the real-time printing of the incoming and outgoing packets.

Example

The following is an example `trace packet off` command:

```
hr-brick1>trace packet off

all filters disabled.
```

□

trace packet on

Overview

The `trace packet on` command enables a filter created with the `trace packet filter` command and starts the display of the packets. The filter can be disabled and the display of the packets can be stopped with the `trace packet off` command.

For performance reasons, the `trace packet on` and `trace audit on` commands should be used separately. If a trace audit filter has been enabled and you want to enable a trace packet filter, issue the `trace audit off` command.

Format

The format of the `trace packet on` command is:

trace packet on *[<id>]*

where:

- *<id>* is the number of the filter you wish to enable. To see the list of filters that can be viewed, *issue the* `trace packet list` *command*. If an <id> is not present, then all filters are enabled.

Explanation

Each line in the output is preceded by an ″P>>interface#″ or ″P<<interface#″ to identify it as a packet entering or exiting the specified Brick port.

″>>″ means the packet is entering the Brick port and ″<<″ means the packet is exiting the Brick port.

The number is the Brick port. The next two values are the source and destination addresses respectively. The next item shows the service (protocol/dst-port/src-port) followed by the packet size (length).

For TCP packets, four additional items are printed: tcp-flags (″A″ - ack number present; ″R″ - reset; ″S″ - syn; ″F″ -fin), the sequence number, the TCP length, and the acknowledgement number.

If the ″m″ option is specified, the next line contains the MAC addresses of the source and destination, respectively, with the IP identification number.

If the ″c″ option is specified, the first 20 bytes of the payload are displayed in the network order along with its ASCII representation. See “trace packet filter” (p. 11-14) of “trace packet modify” (p. 11-17) for a complete explanation of the available parameters and options.

Note that in TCP and UDP packet records, the content begins with their payload. For all other types of protocols, the content begins with the protocol header. When the payload length is zero, no content is shown.

Example

The following is an example `trace packet on` command:

```
2k_brick1> trace packet on
all filters enabled.
2k_brick1> P<<1 10.20.30.153 10.20.30.101 ICMP/8/0 60
P>>1 10.20.30.101 10.20.30.153 ICMP/0/0 60
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 68 A:3178338170:28:64412
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 244
  A:3178338198:204:64412
P<<1 10.20.30.153 10.20.30.101 ICMP/8/0 60
P>>1 10.20.30.101 10.20.30.153 ICMP/0/0 60
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 68 A:3178338402:28:64412
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 40 A:3178338430:0:64414
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 228
  A:3178338430:188:64414
P<<1 10.20.30.153 10.20.30.101 ICMP/8/0 60
P>>1 10.20.30.101 10.20.30.153 ICMP/0/0 60
P>>1 10.20.30.101 10.20.30.153 ICMP/0/0 60
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 244
  A:3178338618:204:64414
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 68 A:3178338822:28:64414
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 40 A:3178338850:0:64416
P<<1 10.20.30.153 10.20.30.101 ICMP/8/0 60
P>>1 10.20.30.101 10.20.30.153 ICMP/0/0 60
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 228
  A:3178338850:188:64416
P<<1 10.20.30.153 10.20.30.101 ICMP/8/0 60
P>>1 10.20.30.101 10.20.30.153 ICMP/0/0 60
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 244
  A:3178339038:204:6441
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 68 A:3178339242:28:64416
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 40 A:3178339270:0:64418
P<<1 10.20.30.153 10.20.30.101 ICMP/8/0 60
P>>1 10.20.30.101 10.20.30.153 ICMP/0/0 60
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 228 A:3178339270:188:644
P<<1 10.20.30.153 10.20.30.101 ICMP/8/0 60
P>>1 10.20.30.101 10.20.30.153 ICMP/0/0 60
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 68 A:3178339458:28:64418
P>># 10.20.30.114 10.20.30.8 TCP/9000/29303 40 A:3178339486:0:64420
P<<1 10.20.30.153 10.20.30.101 ICMP/8/0 60
P>>1 10.20.30.101 10.20.30.153 ICMP/0/0 60
```

Important! If activating this command results in an unmanageable flood of trace records, pressing ″CTRL C″ will halt the output. It is equivalent to typing ″set printing off″, so to reenable tracing, type ″set printing on″.

12 Lucent VPN Firewall*Brick*® Device Set Commands

Overview

Purpose

This chapter provides information to perform the following:

1. Issue `set` commands with the correct syntax.
2. Interpret the results of a `set` command.

Overview

Use the `set` commands to control settings that govern the viewing of the output.

For example, you can turn on or off the viewing of real-time packets and log records as the Brick is processing them. Other commands allow you to set the number of lines that are displayed on the screen and the baud rate of the modem.

The commands in this chapter are listed in alphabetical order. For each command, the chapter provides an overview, a description and explanation of the format, and examples.

To see a complete listing of the display commands online, enter **help set** at the command prompt.

Contents

set baudrate	12-2
set errors	12-3
set printing	12-4
set screensize	12-5
set throttle	12-6

set baudrate

Overview

The `set baudrate` command displays or sets the baudrate of the Brick serial port. The default is 115kb/seconds.

It is recommended that this command be issued only from a monitor and keyboard that are directly plugged into the ports of a Brick.

Format

The format of the `set baudrate` command is:

`set baudrate [<number>]`

where:

- number must be a positive decimal number and a legal serial baud rate.

Explanation

If the baud rate needs to be changed and you are accessing a Brick through a remote connection, this command should be issued before a remote login session is attemped.

If the `set baudrate` command is issued without an argument, then the current modem baudrate is displayed. If an argument is specified, the baudrate will be reset to the new rate.

Example

The following are example `set baudrate` commands:

```
hr-brick1>set baudrate
baudrate = 19200
hr-brick1>set baudrate 9600
baudrate = 9600
```

☐

set errors

Overview

The `set errors` command turns on or off a Brick periodic reporting of error conditions, depending on the argument supplied, or the current state of error reporting if no argument is supplied.

Format

The format of the `set errors` command is:

`set errors` [on|off]

Explanation

Issued with no argument, the set errors command toggles the current state. So, if error reporting is on, issuing set errors with no argument will turn it off.

Example

The following are example `set errors` commands:

```
set errors
error trace setting: on
set errors off
error trace setting off
```

□

set printing

Overview

The `set printing` command enables or disables the real-time display of packets or log records.

Before this command has any effect, a filter to display packets or log records must have been defined and enabled. See `trace packet filter` and `trace audit filter` commands respectively.

Format

The format of the `set printing` command is:

`set printing` *`[on | off]`*

where:

- *`on`* enables the real-time display of log records or packets. This is the initial default setting. When *`on`* is specified, it is equivalent to typing the `<Esc>` key.
- off disables the real-time display of log records or packets. When *`off`* is specified, it is equivalent to entering `<Ctrl>C`.

Explanation

If the `set printing` command is issued without an argument, then the status displays the current setting (i.e., either ″on″ or ″off″).

Example

The following are example `set printing` commands:

```
hr-brick1>set printing
printing = off

hr-brick1>set printing on
printing set on.
```

□

set screensize

Overview

The `set screensize` command sets the number of lines in which the output is displayed. This command has no effect on the display of real-time data, such as packet and log record traces, as defined with `trace packet filter` and `trace audit filter` commands.

If the number of lines printed from a normal command exceeds the actual screensize, printing pauses until a key is pressed.

Format

The format of the `set screensize` command is:

```
set screensize [<lines>]
```

where:

- *<lines>* must be a decimal number greater or equal to 0. Setting the screensize to 0 lines causes an infinite screensize (no scroll mode).

Explanation

If `set screensize` command is issued without a lines argument, then the current value is displayed.

The initial value is 23 - leaving the 24th line for the status line ("Enter any key to continue:").

Example

The following are example `set screensize` commands:

```
hr-brick1>set screensize
screensize = 23

hr-brick1>set screensize 0
screensize = 0 (no scroll mode)
```

□

set throttle

Overview

The `set throttle` command sets the rate at which log error messages are generated.

This command is especially useful to slow down the generation of error messages if voluminous error messages are creating a Denial-of-Service attack situation.

Format

The format of the `set throttle` command is:

`set throttle` *<time_period>*

where:

- *<time_period>* is any positive whole number (1 or greater). It signifies the time period between messages in seconds.

Explanation

Throttling error messages may prevent an attacker from creating a ″denial of service″ on the LSMS in order to mask other harmful or potentially damaging activity.

For example, if repeated unauthorized connection attempts flood the LSMS, throttling would reduce the error messages for the time period that is specified.

Example

The following are example `set throttle` commands:

```
hr-brick1>set throttle 10
message throttle set to 10 seconds.
```

□

13 Lucent VPN Firewall *Brick*® Device General Commands

Overview

Purpose

Use the general commands to log into a Brick, refresh MAC and ARP tables, send modem commands, reboot a Brick, and other functions.

The commands in this chapter are listed in alphabetical order. For each command, the chapter provides an overview, a description and explanation of the format, and examples.

This chapter provides information to perform the following:

- Issue miscellaneous commands with the correct syntax.
- Interpret the results of a command.

Contents

failover yield	13-2
help	13-3
login	13-4
logout	13-5
modem	13-6
reboot	13-7
refresh	13-8
repeat	13-10

failover yield

Overview

The `failover yield` command initiates failover of a Brick failover pair from the Brick CLI.

Format

The format of the `failover yield` command is:

```
failover yield [force]
```

where the [force] option overrides the Halt All Traffic If Audit Fails setting.

Explanation

The `failover yield` command, when issued from the active Brick, switches the Brick to standby mode. If the **Halt All Traffic If Audit Fails** checkbox on the Brick Editor Options tab is set, and log events would be lost, the Brick will not switch. The optional force parameter will override this reluctance. The `failover yield force` command is the equivalent of the `Initiate Failover` command issued from the **Utilities**  **Brick** menu in the LSMS Navigator window.

Example

The following is an example `failover yield` command:

```
brick_45> failover yield
Enter Ôy’ to confirm failover (‘n’ to cancel):
invoking failover...
```

□

help

Overview

The `help` command lists all commands that are described in this chapter. Entering help without any arguments displays the entire list.

Alternatively, `help` can be followed by one of the commands to display help for that command only. For example, entering `help trace` will display help for all trace commands.

Entering `help` by itself displays the following information:

″Most keywords, zone names, and commands can be automatically completed by <space>, <tab>, and <return> keys. Also, the <tab> key can be used to cycle through legitimate tokens at the current position.″

- The ″display″ commands generally print information about tables, policies, and sessions.
- The ″trace″ commands permit the tracing of either log records or packets.
- The ″set″ commands permit the setting and viewing of the screensize, the remote port baudrate, and the toggling of trace related printing.

Format

The format of the `help` command is either:

`help`

or

`help` *<cmd>*

Explanation

The `help` command provides an online explanation of all commands that can be entered to a Brick.

Example

The following is an example `help` command:

```
hr-brick1> help display configuration
Prints the inferno.ini file.
```

□

login

Overview

The `login` command enables an Administrator to log on to a particular Brick.

The `login` command is used with local serial port connections or dial-up connections only. No login is required if you are accessing the Brick command line using a locally connected monitor and keyboard. Login using the Remote Brick Console feature is accomplished with the `brickcon` command.

After logging in, any of the commands that are documented in this section can be entered until a `logout` command is issued.

Format

The format of the login command is:

`login <remote-password>`

where

- *`<remote-password>`* is the Remote Password that you entered and verified in the Bricks Editor Options tab Serial Port Access area. See "Create a Serial Port Access Password" (p. A-9) in Appendix A, "Set up a Remote Dial-In Connection" or in Appendix B, "Set up a Direct Serial Port Connection" for details. When entering the Remote Password, it is echoed to the screen in asterisks.

Explanation

Unless a successful remote password is entered, commands cannot be issued to the Brick. When the command succeeds, an log record is written to the Administrative Events log file.

Example

The following is an example `login` command:

```
Signon to brick> login ******
**************************************************
** Remote Port Login to Lucent Security Product **
** Type help to get list of commands.           **
**************************************************
July 19 2001, 14:43:44 GMT - Login from remote port successful
hr-brick1>
```

logout

Overview

The `logout` command terminates the command line session on a Brick.

After logging out, if you need to log back in, you can enter three carriage returns, one right after another.

Format

The format of the logout command is:

`logout`

Explanation

The logout command is used to end the current session and disconnect from the Brick.

Once the command is successfully executed, the message

`Remote port logging out.... Bye!`

is displayed.

Example

The following is an example `logout` command:

```
hr-brick1> logout
Feb 26 2001, 14:44:45 GMT - Remote Port Logging out.... Bye!
Signon to brick>
```

modem

Overview

The `modem` command sends a Hayes AT command directly to the modem that is attached to the Brick.

It is recommended that this command be issued only from a monitor and keyboard that are directly plugged into the ports of a Brick.

Format

The format of the `modem` command is:

`modem <string>`

where

- *`string`* is a Hayes AT command string (e.g., ATZ, ATDT).

Explanation

If a Hayes AT command needs to be sent to a Brick and you are accessing the Brick through a remote connection, this command should be issued before a remote login session is attempted.

Example

The following is an example `modem` command:

```
hr-brick1>modem ATZ
```

□

reboot

Overview

The reboot command reboots the Brick five seconds after the command is accepted.

After the reboot is complete, a new login session must be initiated.

Format

The format of the reboot command is:

reboot *<”message text”>*

where:

- *<”message text”>* is an optional message, that, if supplied, appears in the log record of the Administrative Events log file.

Explanation

The `reboot` command is used to restart a Brick.

You may want to reboot a Brick any time you believe the Brick is inoperable and is not responding or resetting itself.

Once the command is successfully executed, the message

`reboot of brick in 5 seconds.`

is displayed.

Example

The following is an example `reboot` command:

```
hr-brick1> reboot ”replaced cable on ether2”
reboot of brick in 5 seconds.
```

The log record in the Administrative Events log would then look like:

`4:b:hr-brick1:182147:13:Remote request-”replaced cable on ether2”`

□

refresh

Overview

The `refresh` command forces the Brick to refresh the specified Media Access Control (MAC) table or Address Resolution Protocol (ARP) table.

MAC table particulars

The MAC table tracks the MAC addresses of all MAC elements that are associated with each network interface of a Brick.

ARP table particulars

Every time the Brick needs to resolve an IP address, an ARP request is issued and an entry is written to the ARP table.

Format

For MAC table refreshes:

```
refresh mac
```

For ARP table refreshes:

```
refresh arp [IPADDRESS]
```

where *`[IPADDRESS]`* is a specific IP address to be refreshed in the ARP table retained by the Brick. (This argument is optional.)

Explanation

The `refresh` command is used to refresh the MAC or ARP tables of a Brick.

You may want to display the tables first with either the `display mactable` or `display arptable` command.

Once the command is successfully executed, the message

```
MAC tables cleared.
```

—or—

```
ARP table cleared.
```

is displayed.

Example

The following is an example `refresh` command:

```
hr-brick1> refresh mac
MAC tables cleared.
```

□

repeat

Overview

The `repeat` command repeats the last command that was issued.

Format

The format of the `repeat` command is:

```
repeat
```

Explanation

This command repeats the last command that was issued in addition to any arguments that were specified.

Example

The following is an example `repeat` command:

```
hr-brick1>display time
The current time is Jun 12 2001, 22:03:15 GMT
Last booted at Jun 12 2001, 18:20:47. Active since Jun 12 2001,
  18:20:47
hr-brick1>repeat
hr-brick1>display time
The current time is Jun 12 2001, 22:03:15 GMT
Last booted at Jun 12 2001, 18:20:47. Active since Jun 12 2001,
  18:20:47
```

□

Appendix A: Set up a Remote Dial-In Connection

Overview

Purpose

If you intend to access the Lucent VPN Firewall *Brick*®device CLI by dialing into a brick from a remote computer, you need to install and configure a modem at both ends of the connection.

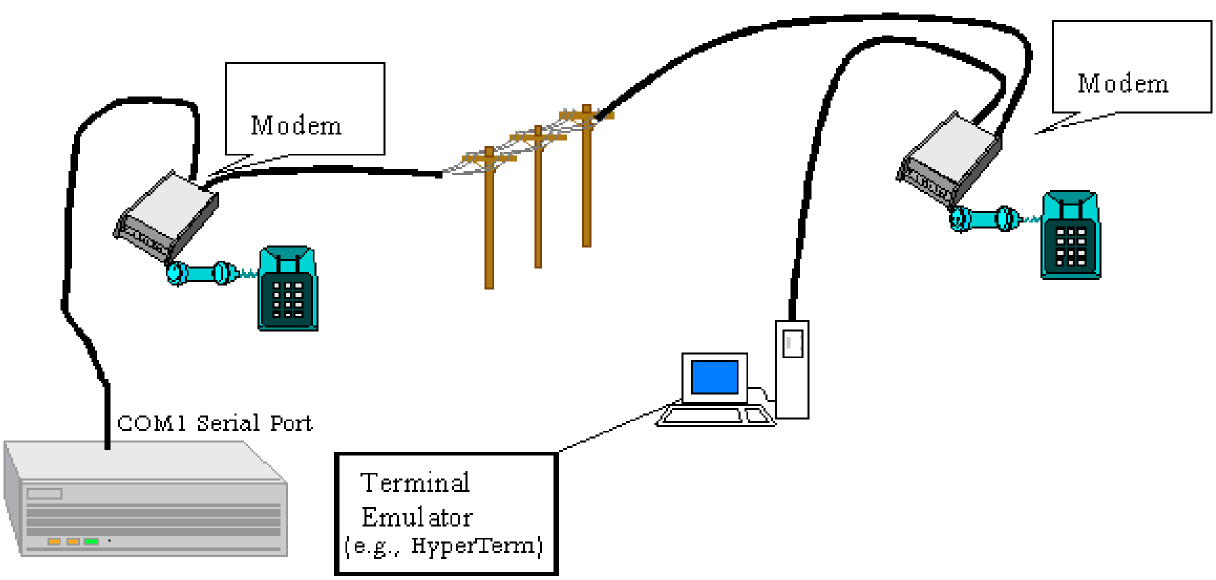
Notes

Important! *LSMS Remote Console* If a Brick connection to the LSMS is still operational, use the Remote Console feature described in Chapter 9, "Introduction to the Lucent VPN Firewall *Brick*® CLI" to access the Brick command line.

As Figure A-1, "Remote Dial-in Connection" (p. A-2) illustrates:

- On the Brick end, connect a Hayes-compatible modem to the COM1 serial port on the back of the Brick.
- On the remote computer end, configure a dial-up, Hayes-compatible modem and use a terminal emulation program, such as HyperTerminal, to dial into the Brick and access the Brick command line interface.

Figure A-1 Remote Dial-in Connection



Contents

Modem Setup on the Brick	A-3
Modem Setup on a Remote Computer	A-4
Create a Serial Port Access Password	A-9
Dial Up and Log Into a Brick	A-11

Modem Setup on the Brick

Task

To set up a modem on a Brick, do the following:

1 Connect a Hayes-compatible modem to the COM1 Serial port on the back of the Brick. The location of the serial port depends on the model of the Brick. See the *User's Guide*of the respective Brick model for a description.

2 Set the DIP switch on the modem to `Auto Answer` mode. Consult the modem manufacturer's documentation for instructions.

3 Enable **Data Terminal Ready** (DTR) override mode on the modem. Consult the modem manufacturer's documentation for instructions.

END OF STEPS

□

Modem Setup on a Remote Computer

When to use

This procedure explains how to configure a HyperTerminal session for dialing into a Brick from a remote computer running Windows 95/98 or Windows NT platform. Like the modem that is connected to the Brick, the modem that is installed on the remote computer end must also be a Hayes-compatible modem.

To configure a HyperTerminal session for remote access, do the following:

1 Verify that the modem and modem drivers are installed on the remote computer. If they are not installed, consult the modem manufacturer's documentation for instructions.

2 Start a terminal emulation program, such as HyperTerminal, as follows:

Start a terminal emulation program, such as HyperTerminal, as follows:

1. Select **Start ▶ Programs ▶ Accessories ▶ HyperTerminal.**
2. Select **HyperTerminal** from the list.

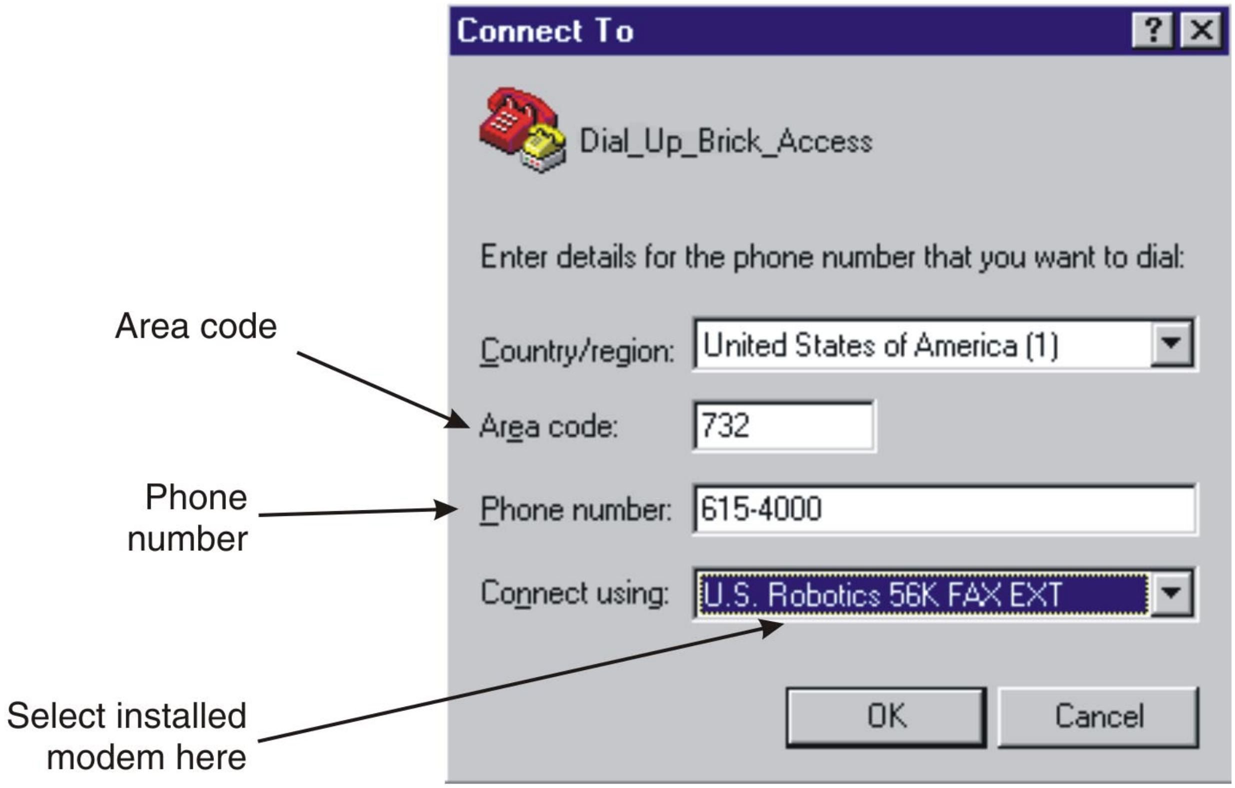
3 Enter a name for the connection (as shown in Figure A-2, "Entering Name for Hyperterminal Connection" (p. A-4)) and optionally select an icon to represent it, then click **OK**.

Figure A-2 Entering Name for Hyperterminal Connection

4 Select the modem (see Figure A-3, "Connect To Window in Hyper Terminal" (p. A-5)) that was installed in Step 1.

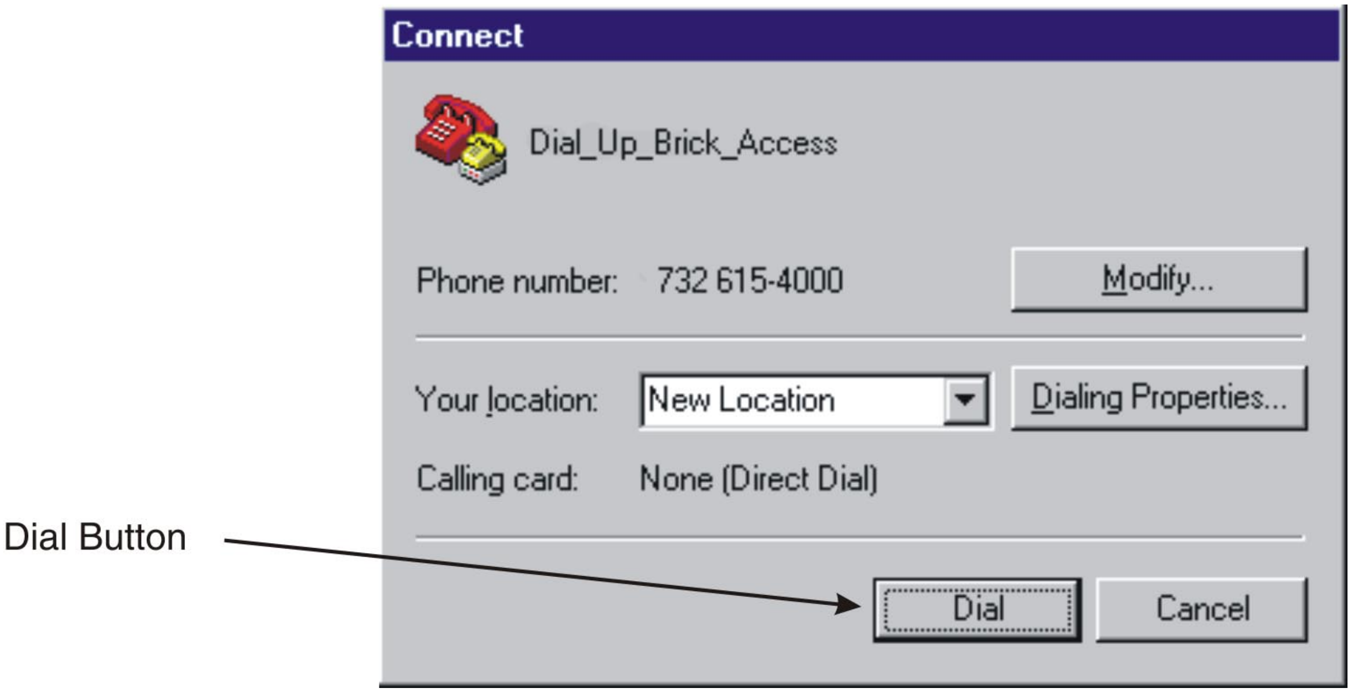
5 In the same window, enter the area code and phone number of the modem that is connected to the Brick and click **OK**.

Figure A-3 Connect To Window in Hyper Terminal



6 At this point, you can dial into the Brick by clicking the **Dial** button in the next window that appears (see Figure A-4, "Connect Window in HyperTerminal" (p. A-6)).

Figure A-4 Connect Window in HyperTerminal



7 As a last step, ensure that the session is using **Auto Detect** as the emulation type.

To double-check this:

1. In the HyperTerminal window, select **File** ▶ **Properties**.
2. Click the **Settings** tab as shown in Figure A-6, "Setting Emulation Type for Remote Computer Modem" (p. A-8).
3. Select **Auto Detect** in the Emulation field.

As shown in Figure A-5, "Configuring Port Settings for Remote Computer Modem" (p. A-7), configure the modem on the remote computer to these settings:

- 115200 bits per second. This is the Brick default.
- 8-N-1 (8 data bits, no parity, 1 stop bit)
- No flow control

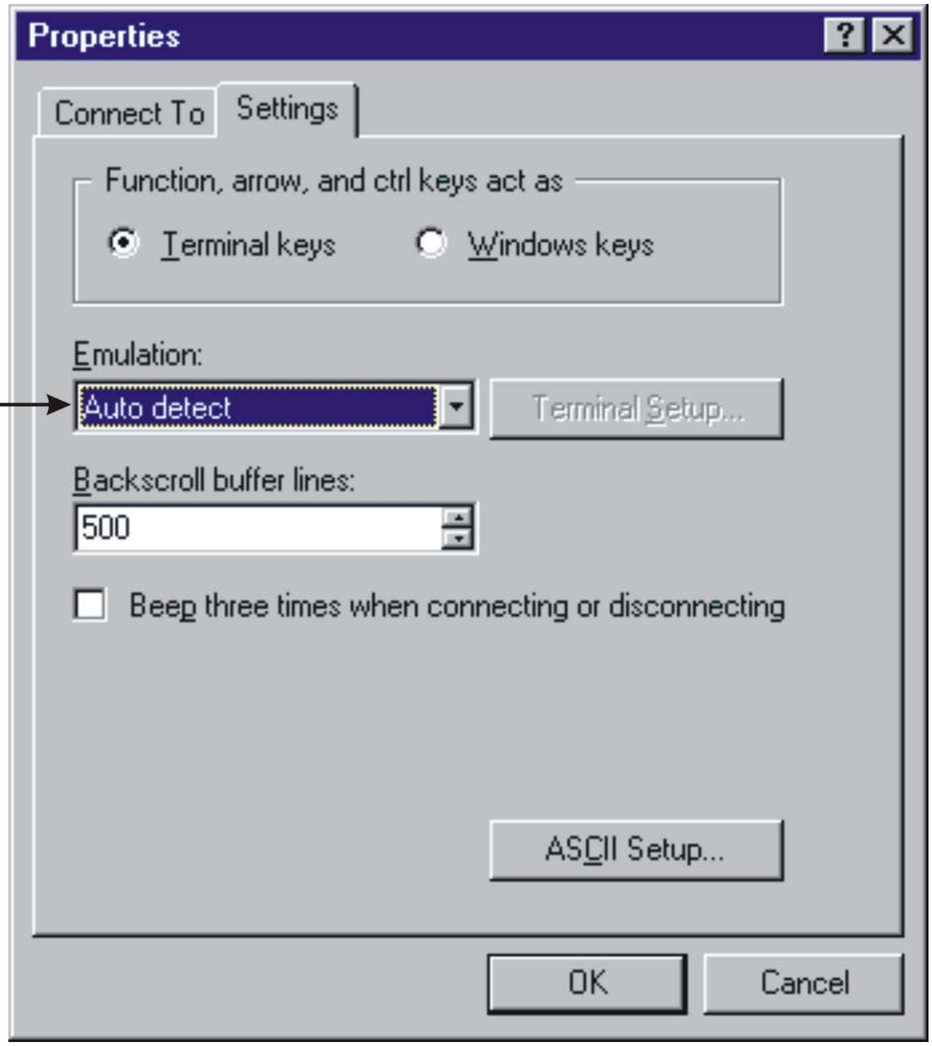
Figure A-5 Configuring Port Settings for Remote Computer Modem

8 Click **OK** to save the port settings.

9 Also, ensure that the emulation type is set to **Auto Detect**. To double-check this, select **File ▶ Properties** and click the **Settings** tab as shown in Figure A-6, "Setting Emulation Type for Remote Computer Modem" (p. A-8).

Figure A-6 Setting Emulation Type for Remote Computer Modem

Select Emulation



END OF STEPS

☐

Create a Serial Port Access Password

Overview

If you are accessing the Brick command line interface by means of a dial-in connection, you will need a Serial Port Access Password.

A Serial Port Access Password is required for each Brick that you want to access via a dial-up connection. To create a Serial Port Access Password, display the Brick Editor **Options** tab in the LSMS graphical user interface.

For a complete description of all fields in the Brick Editor, refer to the *LSMS Administration Guide*.

Procedure

To create a Serial Port Access Password, follow the steps below:

1. Display the Brick Editor. You may either right-click the Bricks folder and select **New Brick** to create a new Brick , or double-click a Brick name to edit an existing Brick.

2. When the Brick Editor appears click **Options** to display the Options tab. If you are creating a new brick, you will have to enter a `Brick Name` and `Brick IP Address/Mask` before you will be able to proceed to the Options tab.

3. In the area labeled **Serial Port Access,** select the **Enable Serial Port** checkbox. The **Password** and **Verify Password** fields become active.

4. Enter the password in the **Password** field, then enter it a second time in the `Verify Password` field.

5. Select **Save and Apply** from the File menu.

6. For an explanation on how to configure and activate a Brick with **Make/Package Floppy,** refer to the *LSMS Administration Guide*.

7 Reboot the Brick.

END OF STEPS

Dial Up and Log Into a Brick

Procedure

For remote dial-in and external direct connections only, to login into a Brick:

1 Dial into the Brick using the Brick modem number. This can be done from HyperTerminal's Call menu or via the ATDT <dial string> command.

2 After the connection is made, the Brick sends the prompt:

```
**************************************************
** Lucent Managed Security Product (enter 3>CR>)**
```

3 After entering three carriage returns, the Brick sends the following message:

```
Signon to brick>
```

4 At this point, enter the login command and the Serial Port Access Password as created in the Brick Editor (see "Create a Serial Port Access Password" (p. A-9)).

```
 Signon to brick>
login
    ******
  **************************************************
 ** Remote Port Login to Lucent Security Product **
 **          Type help to get list of commands.  **
 **************************************************
 July 19 2001, 14:43:44 GMT - Login from remote port successful
 hr-brick1>
```

5 Upon successful completion of this command, a log record is written to the Administrative Events log. See Chapter 4, “Types of LSMS logs” , in the *Reports, Alarms, and Logs Guide* for details on this log file.

END OF STEPS

Appendix B: Set up a Direct Serial Port Connection

Overview

Purpose

Accessing the Lucent VPN Firewall*Brick*® device command line interface (CLI) can be accomplished by connecting a computer equipped with a terminal emulation program such as HyperTerminal to a serial port on the back of a Brick.

As illustrated in Figure B-1 , the serial connection can be a null modem serial cable (RS-232) or two regular serial cables coupled with a null modem connection.

Contents

Set Up a Local Serial Port Connection	B-2
Create a Serial Port Access Password	B-5
Log In to a Brick	B-7

Set Up a Local Serial Port Connection

Task

To set up an local serial port connection, do the following:

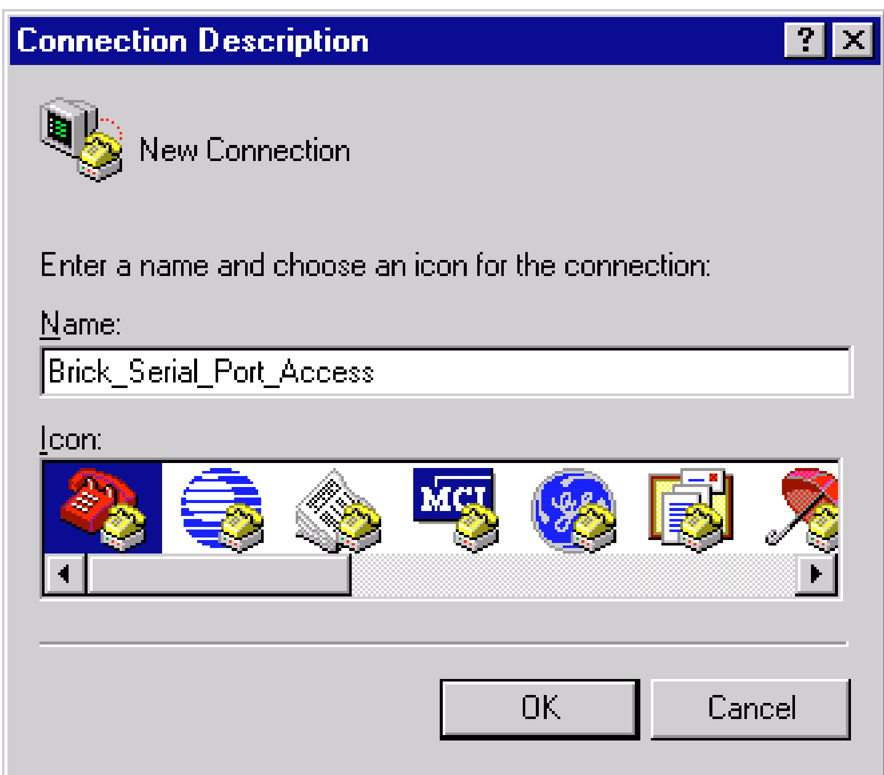
1 Connect the serial cable to the COM1 serial port on the back of the Brick.

2 On the computer, start a terminal emulation program, such as HyperTerminal, by doing the following:

1. Select **Start** ▶ **Programs** ▶ **Accessories** ▶ **HyperTerminal.**
2. Select **HyperTerminal** from the list.

3 Enter a name for the connection, as shown in Figure B-1, "Entering Name for Hyperterminal Connection" (p. B-2), and optionally select an icon to represent it, then click **OK**.

Figure B-1 Entering Name for Hyperterminal Connection

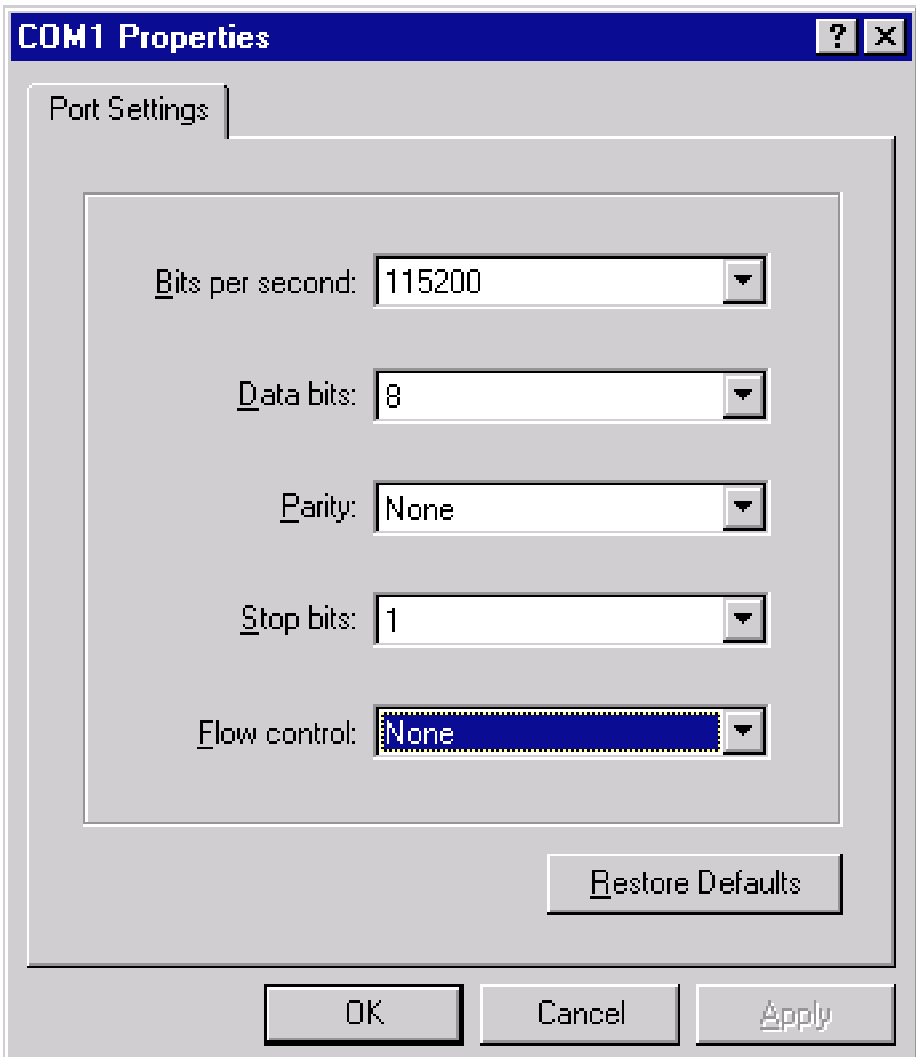


4 Select **COM1** and click **OK**.

5 In the next window (see Figure B-2, “Configuring Port Settings for Remote Computer Modem” (p. B-3)), configure the COM1 port with these settings:

- 115200 bits per second.
- 8-N-1 (8 data bits, no parity, 1 stop bit).
- No flow control

Figure B-2 Configuring Port Settings for Remote Computer Modem



6 Click **OK** to save the port settings.

Also, ensure that the emulation type is set to **Auto Detect**. To double-check this, select **File ▶ Properties** and click the **Settings** tab.

END OF STEPS

Create a Serial Port Access Password

Overview

If you are accessing the Brick CLI by means of a local serial port connection, you will need a Serial Port Access Password.

A Serial Port Access Password is required for each Brick that you want to access with the Brick CLI. To create a Serial Port Access Password, you have to display the Brick Editor in the LSMS graphical user interface.

For a complete description of all fields in the Brick Editor, refer to the *LSMS Administrator Guide*.

Procedure

To create a Serial Port Access Password, follow the steps below:

1. Display the Brick Editor. You may either right-click the Bricks folder and select **New Brick** to create a new Brick, or double-click a Brick name to edit an existing Brick.

2. When the Brick Editor appears click **Options** to display the Options tab. If you are creating a new Brick, you will have to enter a `Brick Name` and `Brick IP Address/Mask` before you will be able to proceed to the Options tab.

3. In the area labeled **Serial Port Access**, select the **Enable Serial Port** checkbox. The **Password** and **Verify Password** fields become active.

4. Enter the password in the **Password** field, then enter it a second time in the **Verify Password** field.

5. Select **Save and Apply** from the File menu.

6. For an explanation on how to configure and activate a Brick with **Make/Package Floppy**, refer to the *LSMS Administrator Guide*.

7 Reboot the Brick.

END OF STEPS

Log In to a Brick

Procedure

For external direct connections only, to log in into a Brick:

1. Dial into the Brick using the Brick modem number. This can be done from HyperTerminal's Call menu or via the ATDT <dial string> command.

2. After the connection is made, the Brick sends the prompt:

```
****************************************************
** Lucent Managed Security Product (enter 3 <CR>)**
```

3. After entering three carriage returns, the Brick sends the following message:

```
Signon to brick>
```

4. At this point, enter the login command and the Remote Password as created in the Brick Editor (see "Create a Serial Port Access Password" (p. B-5)).

5. Upon successful completion of this command, an log record is written to the Administrative Events log. See Chapter 4, ″Types of LSMS Logs″ in the *Reports, Alarms, and Logs Guide* for details on this log file.

```
Signon to brick> login ******
```

```
**************************************************
** Remote Port Login to Lucent Security Product **
** Type help to get list of commands. **
**************************************************
July 19 2001, 14:43:44 GMT - Login from remote port successful
hr-brick1>
```

END OF STEPS

Index

B Brick devices

supported, 1-2

Bricks viewer/editor, A-9, B-5

D dbsetup, 6-8