VitalQIP®

SNMP Module

Version 2.2

User's Guide

190-409-038R7.1
Issue 1
August 2007

Apache

This product includes software developed by the Apache Software Foundation (http:// www.apache.org/).

Contents

About this manual

Purpose

Welcome to the Simple Network Management Protocol (SNMP) Module – a powerful network device and service management tool. The SNMP Module, a valuable enhancement to the base VitalQIP™ product, allows you to monitor the status and usage of your Domain Name System (DNS) and Dynamic Host Configuration Protocol (DHCP) servers.

Refer to this section for the audience, organization, and typographical conventions used in this manual. This section also describes the package contents, how to order additional manuals, and how to obtain technical support.

Intended audience

This manual is intended for SNMP Module users who plan to manage and administer an IP network address infrastructure. The reader is expected to understand basic networking concepts and have a working knowledge of the operating system on which the SNMP Module is running.

Two types of groups interact with the SNMP Module:

1. **SNMP Module administrators**—The Information Technology (IT) professionals who install, configure, and administer the SNMP Module product.
2. **SNMP Module users**—The IT professionals who use the SNMP Module as a service-level monitoring and capacity tool.

How to use this information product

The manual is organized as follows:

Chapter 1: SNMP Support for the Lucent DHCP and DNS Servers

This chapter describes how Simple Network Management Protocol (SNMP) support is provided to the Lucent DHCP and Lucent DNS servers.

It also describes how these servers use SNMP MIB variables and it provides summary tables of the Lucent DHCP and Lucent DNS server SNMP MIB variables.

Chapter 2: Install and configure the SNMP Module

This chapter describes the tasks required to install and configure the Lucent SNMP DHCP and DNS Agents.

Chapter 3: Start the SNMP Master Agent

This chapter describes how to start the SNMP Agent and how to verify SNMP operations.

Appendix A: Increase SNMP Module security

This appendix describes adding additional security to the SNMP Module by limiting SNMPv1 and SNMPv2c access to MIB variables. It also describes adding additional security using SNMPv3 authentication and privacy protocols.

Conventions used

The following table lists the typographical conventions used throughout this manual.

Typographical conventions

Convention	Meaning	Example
boldface	Names of items on windows. Names of commands and routines. Names of buttons you should click.	Select the **Client** check box. The **qip_getappllst** routine returns the entire list of existing applications. Click **OK**.
Arial boldface	Names of keys on the keyboard to be pressed.	Press **Enter** to continue.
`courier font`	Input that you should enter from your keyboard.	Run the following command: `c:\setup.exe`
<angle brackets>	Variables for which you must substitute another value.	*http://<VitalQIP_server_IP_address_or_name>*
italics	Names of manuals and emphasis.	Refer to the ***VitalQIP Administrator Reference Manual*** for more information.
Arial italic	Directories, paths, file names, e-mail addresses, and Uniform Resource Locators (URLs).	The VitalQIP web site is *http://qip.lucent.com.*
click	Click the left button on your mouse once.	To delete the object, click **Delete**.
right-click	Click the right button on your mouse.	Right-click on a service.

Convention	Meaning	Example
double-click	Double-click the left button on your mouse.	Double-click the book icon.

Related information

Use the following manuals with this product:

- *VitalQIP Administrator Reference Manual* (part number: 190-409-042)

  This guide describes planning and configuring your network, information about the VitalQIP interface, advanced DNS and DHCP configurations, and troubleshooting.

- *SNMP Module User's Guide* (part number: 190-409-038)

  This guide describes how to install and use the Simple Network Management Protocol (SNMP) Module. The SNMP Module provides a standard way for management products to monitor network devices and services. The application is purchased separately and requires a license key.

Technical support

If you need assistance with SNMP Module, you can contact Technical Support via phone or email. Refer to the following table for a list of phone numbers, addresses, and email addresses:

Table 3-1 Technical support information

Region	Address	Contact information
North, Central, and South America	Alcatel-Lucent 400 Lapp Road Malvern, PA 19355 USA	Phone: 1-866-LUCENT8 (582-3688) Option 1, Option 2 Web: **https://support.lucent.com**
Europe, Middle East, Africa, and China	Alcatel-Lucent Chiltern House Sterling Court Broad Lane Bracknell, RG12 9GU UK	Phone: 00 800 00 LUCENT or +353 1 692 4579 E-mail: *emeacallcenter@alcatel-lucent.com* Web: **https://support.lucent.com**
Central and South America	Alcatel-Lucent Calle 10, No. 145 San Pedro de los Pinos, 01180 Ciudad de Mexico Mexico	Mexico 01 800 123 8705 or (52) 55 5278 7005 Brazil 0800 89 19325 or (55) 193707 7900 Argentina 0800 666 1687 Venezuela 0 800 1004136 Costa Rica 0800-012-2222 or 1800 58 58877 For other local CALA numbers, consult the web site **https://support.lucent.com** or contact your local sales representative.

Region	Address	Contact information
Asia Pacific	Alcatel-Lucent Australia 68 Waterloo Road North Ryde NSW 2113 Australia	Phone: 1800-458-236 (toll free from within Australia) (IDD) 800-5823-6888 (toll free from Asia Pacific - Hong Kong, Indonesia, South Korea, Malaysia, New Zealand, Philippines, Singapore, Taiwan, and Thailand) (613) 9614-8530 (toll call from any country) E-mail: *apactss@alcatel-lucent.com*

Training Support

Alcatel-Lucent University offers cost-effective educational programs that support the VitalQIP product. Our offerings also include courses on the underlying technology for the VitalQIP products (for example, DNS and DHCP). Our classes blend presentation, discussion, and hands-on exercises to reinforce learning. Students acquire in-depth knowledge and gain expertise by practicing with our products in a controlled, instructor-facilitated setting. If you have any questions, please contact us at 1 888 LUCENT8, option 2, option 2.

How to order

Customers can order additional VitalQIP manuals online at **http://www.lucentdocs.com/cgi-bin/CIC_store.cgi**. Select **VitalQIP** from the Product Line list and click **Go**.

How to comment

To comment on this document, go to the Online Comment Form or e-mail your comments to the Comments Hotline (**comments@alcatel-lucent.com**).

1 SNMP Support for the Lucent DHCP and DNS Servers

Overview

Purpose

This chapter describes how Simple Network Management Protocol (SNMP) support is provided to the Lucent DHCP and Lucent DNS servers.

This information presents the following topics.

Introduction to SNMP support for the Lucent DHCP and DNS

SNMP provides an industry-standard protocol used by a number of Network Management products, such as HP Openview, to manage devices and services on the network. SNMP provides a standard way for management products to monitor network devices and services. The Lucent Management Information Base (MIB) is SNMPV1, SNMPV2c, and SNMPV3 compliant.

Lucent Technologies provides SNMP support to our Lucent DNS and Lucent DHCP servers, enabling the collection and monitoring of statistics and general operational information through the use of SNMP MIB variables. The Lucent DNS and DHCP servers can provide information through these MIB variables. The variables offered by Lucent have been designed to be used in conjunction with MIB-2, to allow monitoring of Lucent DHCP and DNS name services via SNMP.

Lucent DHCP MIB variables

Alcatel-Lucent has modified the Lucent DHCP server on all supported platforms to optionally support SNMP. The statistical information gathered by the DHCP server through normal operations can be accessed through the Lucent MIB variables.

Alcatel-Lucent has implemented portions of the DHCP MIB objects defined by the DHCP Working Group of the Internet Engineering Task Force (IETF) in a proposed draft. In particular, there is support for Bootp and DHCP counter and statistics groups. The supported DHCP server MIB variables, are grouped in categories listed in the following tables. Refer to Table 1-1 for a description of each variable.

Table 1-1 Summary of SNMP MIB variables for the Lucent DHCP server

Function	MIB Variable(s)	Description
Operational information		
Server Information	*dhcpServSystemDescr*	Provides a textual description of the server. This value includes the full name and version identification of the server.
Server Status	*dhcpServSystemStatus*	Status of the DHCP server: 0 – Starting server up 1 – Server is running 2 – Server is stopping 3 – Server is halted 4 – Server is reloading its configuration **Note:** Once the server has been completely stopped, no status can be returned from this variable.
Number of seconds since service was started	*dhcpServSystemUpTime*	This value is the time elapsed (in seconds) since it started.
Number of seconds since service was last reset (config files were re-read)	*dhcpServSystemResetTime*	This value is the time elapsed (in seconds) since the last time the name server was “reset”.

Function	MIB Variable(s)	Description
Counter information by server		
Number of Used Subnets (in use)	*dhcpServCountUsedSubnets*	The number of subnets managed by the server (for example, configured), from which the server has issued at least one lease.
Number of Unused Subnets (not in use)	*dhcpServCountUnusedSubnets*	The number of subnets managed by the server, from which the server has issued no leases.
Number of exhausted/full Subnets	*dhcpServCountFullSubnets*	The number of subnets managed by the server, in which all defined addresses have been leased to clients. Subnets containing unavailable leases are not represented in this counter.
Counter information for Bootp packets		
Number of Bootp Request Packets received	*dhcpServBootpCountRequests*	The number of packets received that contain a Message Type of 1 (BOOTREQUEST) in the first octet and do not contain option number 53 (DHCP Message Type) in the options.
Number of Invalid Bootp Request Packets received	*dhcpServBootpCountInvalids*	The number of packets received that do not contain a Message Type of 1 (BOOTREQUEST) in the first octet or are not valid BOOTP packets (for example, too short, invalid field in packet header)
Number of Bootp Packets sent	*dhcpServBootpCountReplies*	The number of packets sent that contain a Message Type of 2 (BOOTREPLY) in the first octet and do not contain option number 53 (DHCP Message Type) in the options.

Function	MIB Variable(s)	Description
Number of Bootp Packets dropped with unknown clients	*dhcpServBootpCountDroppedUnknownClients*	The number of BOOTP packets dropped due to the server not recognizing or not providing service to the hardware address received in the incoming packet.
Number of Bootp packets dropped because this server cannot serve addresses to this subnet	*dhcpServBootpCountDroppedNotServing Subnet*	The number of BOOTP packets dropped due to the server not being configured or not able to serve addresses on the subnet from which this message was received.
Counter information for DHCP packets		
Number of DHCP Discover Packets received	*dhcpServDhcpCountDiscovers*	The number of DHCPDISCOVER (option 53 with value 1) packets received.
Number of DHCP Request Packets received	*dhcpServDhcpCountRequests*	The number of DHCPREQUEST (option 53 with value 3) packets received.
Number of DHCP Release Packets received	*dhcpServDhcpCountReleases*	The number of DHCPRELEASE (option 53 with value 7) packets received.
Number of DHCP Decline Packets received	*dhcpServDhcpCountDeclines*	The number of DHCPDECLINE (option 53 with value 4) packets received.
Number of DHCP Inform Packets received	*dhcpServDhcpCountInforms*	The number of DHCPINFORM (option 53 with value 8) packets received.

Function	MIB Variable(s)	Description
Number of Invalid DHCP packets received	*dhcpServDhcpCountInvalids*	The number of DHCP packets received whose DHCP message type (option 53) is not understood or handled by the server.
Number of DHCP Offers sent	*dhcpServDhcpCountOffers*	The number of DHCPOFFER (option 53 with value 2) packets sent.
Number of DHCP Acks sent	*dhcpServDhcpCountAcks*	The number of DHCPACK (option 53 with value 5) packets sent.
Number of DHCP Nacks sent	*dhcpServDhcpCountNacks*	The number of DHCPNACK (option 53 with value 6) packets sent.
Number of DHCP Packets dropped with unknown clients	*dhcpServDhcpCountDroppedUnknownClient*	The number of DHCP packets dropped due to the server not recognizing or not providing service to the client ID and/or hardware address received in the incoming packet.
Number of DHCP Packets dropped because this server cannot serve addresses to this subnet	*dhcpServDhcpCountDroppedNotServingSubnet*	The number of DHCP packets dropped due to the server not being configured or not able to serve addresses on the subnet from which this message was received.

Function	MIB Variable(s)	Description
Statistical/performance information for Bootp packets		
Minimum Amount of Time between receiving two Bootp packets	*dhcpServBootpStatMinArrivalInterval*	The minimum amount of time between receiving two BOOTP messages. A message is received at the server when the server is able to begin processing the message. This typically occurs immediately after the message is read into server memory. If no messages have been received, or the time interval is less than 1 ms, this object contains a zero value. The value is in milliseconds.
Maximum Amount of Time between receiving two Bootp packets	*dhcpServBootpStatMaxArrivalInterval*	The maximum amount of time between receiving two BOOTP messages. A message is received at the server when the server is able to begin processing the message. This typically occurs immediately after the message is read into server memory. If no messages have been received, this object contains a zero value. The value is in milliseconds.
Number of Seconds since the last Bootp Packet was received	*dhcpServBootpStatLastArrivalTime*	The number of seconds since the last valid BOOTP message was received by the server. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value.

Function	MIB Variable(s)	Description
Minimum Response Time to Bootp packets	*dhcpServBootpStatMinResponseTime*	The smallest time interval measured as the difference between the arrival of a BOOTP message at the server and the successful transmission of the response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received, or the interval is less than 1 ms, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Maximum Response Time to Bootp packets	*dhcpServBootpStatMaxResponseTime*	The largest time interval measured as the difference between the arrival of a BOOTP message at the server and the successful transmission of the response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Sum of the Response Times for Bootp packets	*dhcpServBootpStatSumResponseTime*	The sum of the response time intervals (in milliseconds), where a response time interval is measured as the difference between the arrival of a BOOTP message at the server and the successful transmission of the response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value. The value is in milliseconds.
Statistical/performance information for DHCP packets		
Minimum Amount of Time between receiving two DHCP packets	*dhcpServDhcpStatMinArrivalInterval*	The minimum amount of time between receiving two DHCP messages. A message is received at the server when the server is able to begin processing the message. This typically occurs immediately after the message is read into server memory. If no messages have been received, or the interval is less than 1 ms, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Maximum Amount of Time between receiving two DHCP packets	*dhcpServDhcpStatMaxArrivalInterval*	The maximum amount of time between receiving two DHCP messages. A message is received at the server when the server is able to begin processing the message. This typically occurs immediately after the message is read into server memory. If no messages have been received, this object contains a zero value. The value is in milliseconds.
Number of Seconds since the last DHCP Packet was received	*dhcpServDhcpStatLastArrivalTime*	The number of seconds since the last valid DHCP message was received by the server. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value.

Function	MIB Variable(s)	Description
Minimum Response Time to DHCP packets	*dhcpServDhcpStatMinResponseTime*	The smallest time interval measured as the difference between the arrival of a DHCP message at the server and the successful transmission of the response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received, or the interval is less than 1 ms, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Maximum Response Time to DHCP packets	*dhcpServDhcpStatMaxResponseTime*	The largest time interval measured as the difference between the arrival of a DHCP message at the server and the successful transmission of the response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Sum of the Response Times for DHCP packets	*dhcpServDhcpStatSumResponseTime*	The sum of the response time intervals (in milliseconds), where a response time interval is measured as the difference between the arrival of a DHCP message at the server and the successful transmission of the response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value. Value is in milliseconds.

Function	MIB Variable(s)	Description
DHCP and Bootp statistics by subnet and/or address pool		
	dhcpServRangeEntry ::= SEQUENCE *{* *dhcpServRangeSubnetAddr,* *dhcpServRangeSubnetMask,* *dhcpServRangeStart,* *dhcpServRangeEnd,* *dhcpServRangeInUse,* *dhcpServRangeOutstandingOffers,* *dhcpServRangeUnavailable,* *dhcpServRangeType,* *dhcpServRangeUnused* *}*	Note that a table is provided that will allow all of the following information to be accessed not only by subnet, but also by address pool (scope).
Subnet Address of the Address Pool (scope) of the table entry that is being referenced	*dhcpServRangeSubnetAddr*	The IP address defining this subnet.
Subnet Mask of the Address Pool (scope) of the table entry that is being referenced	*dhcpServRangeSubnetMask*	The subnet mask associated with this subnet.
Starting IP Address of the Address Pool of the table entry that is being referenced	*dhcpServRangeStart*	The starting IP Address of the Address pool range for this entry within the table.
Ending IP Address of the Address Pool of the table entry that is being referenced	*dhcpServRangeEnd*	The ending IP Address of the Address pool range for this entry within the table.

Function	MIB Variable(s)	Description
Number of Addresses in this range (of the table entry) that are Used (in use)	*dhcpServRangeInUse*	The number of addresses in this range that are currently in use. This number includes address leases that have not expired, and addresses that have been reserved (by the server through configuration).
Number of Addresses in this range (of the table entry) that have outstanding Offers pending	*dhcpServRangeOutstandingOffers*	The number of outstanding DHCPOFFER messages for this range is reported with this value. An offer is outstanding if the server has sent a DHCPOFFER message to a client, but has not yet received a DHCPREQUEST message from the client, nor h as the server-specific timeout (limiting the time in which a client can respond to the offer message) for the offer message expired.
Number of Addresses in this range (of the table entry) that are unavailable	*dhcpServRangeUnavailable*	The number of IP Addresses within this range that are marked by the DHCP server as unavailable. An address is marked as unavailable by the DHCP server when an acknowledgement of the address conflict detection (PING) is received. In addition, the DHCP server will mark leases as unavailable if a DHCPDECLINE is received from the DHCP client.

Function	MIB Variable(s)	Description
The Type of (IP) Address range of the table entry	*dhcpServRangeType*	DHCP Server Client Lease Type: 1 – Manual Bootp 2 – Automatic Bootp 3 – Manual DHCP 4 – Automatic DHCP 5 – Dynamic DHCP
Number of Addresses in this range (of the table entry) that are unused or available for assignment	*dhcpServRangeUnused*	The number of addresses in this range that are currently unused. This number excludes address leases that have not expired, and addresses that have been reserved (by the server through configuration).
DHCP Server Failover Configuration		
Important! These MIB variables are only available when you have a DHCP Failover server configuration.		
Indicates the IP address of the partner server.	*dhcpServFailoverPartnerAddr*	Shows the failover server IP address for a queried primary server, or the primary server IP address for a queried failover server. If no failover server is defined, this has a null value.

Function	MIB Variable(s)	Description
Indicates the status of the partner server	*dhcpServFailoverPartnerStatus*	This variable inidcates the last known state of the queried server's partner. The following values indicate the status of the partner server: **unknown (0)** - The status of the partner server identified in the dhcpServFailoverPartnerAddr variable, as defined in this server's dhcpd.pcy file, is unknown. This value is valid for primary and secondary partner servers. **syncing (1)** - The partner server identified in the dhcpFailoverPartnerAddr variable is exchanging lease data with the server maintaining the partner MIB variables. This value is valid for primary and secondary partner servers. **active (2)** - The partner server identified in the dhcpFailoverPartnerAddr variable is running and is giving out leases on its configured subnets. This value is only valid when querying a secondary server for the status of its primary partner.

Function	MIB Variable(s)	Description
	dhcpServFailoverPartnerStatus (continued)	**inactive (3)** - The partner server identified in the dhcpFailoverPartnerAddr variable is not responding to poll messages and not giving out leases on its configured subnets. This value is only valid when querying a secondary server for the status of its primary partner. When querying a primary server for the status of its secondary partner, only the 0 and 1 values are used. The primary server does not poll the secondary server. As a result, the primary server does not know if the secondary server is running after the primary server has synchronized with the secondary server. After the primary server has synchronized with the secondary server, the partner status value for the secondary server is set to the unknown (0) status.
Indicates the function of the partner server	*dhcpServFailoverPartnerType*	The type of partner server. The following values indicate the type of partner server: **primary (1)** - The partner server is a primary server. **secondary (2)** - The partner server is a secondary server.
Indicates the time of the last poll or response	*dhcpServFailoverPartnerPolltime*	A timestamp documenting the time of receipt for the last poll message from a secondary partner server or last poll resonse from each primary server.

Function	MIB Variable(s)	Description
DHCP server SNMP traps		
The DHCP server has started	*dhcpServerStarted*	The DHCP server has been started.
The DHCP server has stopped	*dhcpServerStopped*	The DHCP server has been stopped.
The DHCP server has reloaded its configuration	*dhcpServerReload*	The DHCP server has been told to reload its configuration.
The DHCP server has determined that a subnet has been depleted of addresses that satisfy the configured ForceClass server policy value.	*dhcpServerSubnetDepleted*	The DHCP server has used all the addresses within a subnet that satisfy the configured ForceClass server policy value, and has received a discover request for which it cannot offer a lease. This trap will be generated for each such discover request that cannot be offered a lease. The address of the depleted subnet is included in the trap text.
The DHCP server receives a bad packet.	*dhcpServerBadPacket*	The DHCP server has received a malformed packet.
The DHCP Failover server has taken control of some address space	*dhcpServerFailoverActive*	This trap is issued by the secondary server. A primary server is down and its scopes will be serviced by this failover server.
The DHCP Failover server has returned control to the primary server for some address space	*dhcpServerFailoverReturnedControl*	This trap is issued by the secondary server. The failover server has returned control to its primary partner.

Function	MIB Variable(s)	Description
Indicates the number of leases issued by the server has exceeded the specified threshold value set for the subnet.	*dhcpServerSubnetThresholdExceeded*	The trap is issued by the LucentDHCP server when the percentage of used addresses in a subnet has exceeded the value of the threshold defined for the subnet or global server threshold, assuming that a subnet-specific value is not specified with a subnet in the server configuration file.
Indicates the number of leases issued by the server fallen below the threshold value set for the subnet.	*dhcpServerSubnetThresholdDescent*	The trap is issued by the LucentDHCP server when the percentage of used addresses in a subnet has fallen below the value of the threshold defined for the subnet or global server threshold, assuming that a subnet-specific value is not specified with a subnet in the server configuration file.
Indicates the lease request is from an unknown client.	*dhcpServerDropUnknownClient*	The trap is issued when an unregistered client attempts to obtain a DHCP lease.
Indicates the address for which the server wants to provide a lease is unavailable.	*dhcpServerPingResponseReceived*	The address that the server wanted to provide is not available as indicated by a ping response. This can indicate unauthorized use of the address or the network.

Lucent DNS MIB variables

The following MIB variable definitions are Alcatel-Lucent extensions to the standard DNS MIB. They are used to count statistics that are not covered in the DNS MIB defined by RFC1611. Together, the supported DNS server MIB variables, fall within the following categories. Refer to Table 1-2 for a description of each MIB variable.

Table 1-2 Summary of SNMP MIB variables for the Lucent DNS server

Function	MIB Variable(s)	Description
Operational information		
Server information	*dnsServSystemDescr*	Provides a textual description of the server. This value includes the full name and version identification of the server.
Server Status	*dnsServSystemStatus*	The current status of the server: 1 – Some other state that is not listed in 2–4 2 – The service is being reset 3 – The service is initializing 4 – The service is running **Note:** Once the server has been completely stopped, no status can be returned from this variable.
Number of seconds since service was started	*dnsServConfigUpTime*	If the server has a persistent state (for example, a process), this value will be the time elapsed (in seconds) since it started. For software without a persistent state, this value will be zero.
Number of seconds since service was last reset (config files were re-read)	*dnsServConfigResetTime*	This value is the time elapsed (in seconds) since the last time the name server was “reset”.

Function	MIB Variable(s)	Description
Recursion state of the server	*dnsServConfigRecurs*	***For DNS 4.0:*** Defaults to 0 but does not necessarily reflect the correct value due to the introduction of views in BIND9. ***For DNS 3.1:*** Returns the recursion state of the server: 1 – (available) performs recursion on requests from clients. 0– (unavailable) recursion is not available. Note: Starting the server with the –r recursion flag will effect this value.

Function	MIB Variable(s)	Description
Round Robin Status of the server	*dnsServConfigRoundRobin*	***For DNS 4.0:*** The current state of Lucent qddns option “rrset-order” : (on) 1 - default random cyclic ordering. (off) 0 - fixed ordering ***For DNS 3.1:*** The current state of Round Robin within the service: on (1) - means DNSSEC is aware of the new round robin status. This mode is activated if roundrobin yes; is specified and rrset-order is not specified in the named.conf file. off (0) - means round robin is off. This mode is activated if roundrobin no; is specified in the named.conf file. old (2) - means classic round robin. This mode is activated if roundrobin old; is specified and rrset-order is not specified in the named.conf file.
Counter information by OP Code/class/resource record type		
	dnsServCounterEntry ::= *SEQUENCE {* *dnsServCounterOpCode,* *dnsServCounterQClass,* *dnsServCounterQType,* *dnsServCounterTransport,* *dnsServCounterRequests,* *dnsServCounterResponses* *}*	Note that a table is provided that will allow request and response information counters to be accessed by OpCode, Class, Type, and Transport.

Function	MIB Variable(s)	Description
The DNS OP Code of this table entry	*dnsServCounterOpCode*	The DNS OP Code being counted in this row of the table: 0 – A standard query (QUERY) 1 – (obsolete) An inverse query (IQUERY) 2 – A server status request (STATUS) 4 - A notify (NOTIFY) 5 - A dynamic update (UPDATE)
The DNS Class of this table entry	*dnsServCounterQClass*	The class of record being counted in this row of the table. 1 – 'IN' the Internet 2 – 'CS' the CSNET class (Obsolete) 3 – 'CH' the CHAOS class 4 – 'HS' Hesiod
The DNS Record Type of this table entry	*dnsServCounterQType*	
	Resource Record type of record being counted in this table row:	
Important! Variables in this section with an asterisk (*) at the end of their description are available for use with DNS 4.0. However their use and definition in DNS 4.0 differs from what is provided here. See the *DNS 4.0 Release Notes* for information on how these variables are used in DNS 4.0.		

Function	MIB Variable(s)	Description
	1 – Host address (A) 2 – Authoritative server (NS) 3 – Mail destination (MD) 4 – Mail forwarder (MF) 5 – Canonical name (CNAME) 6 – Start of authority zone (SOA) 7 – Mailbox domain name (MB) 8 – Mail group member (MG) 9 – Mail rename name (MR) 10 – Null resource record (NULL) 11 – Well known service (WKS) 12 – Domain name pointer (PTR) 13 – Host information (HINFO) 14 – Mailbox information (MWFO) 15 – Mail routing information (MX) 16 – Text strings (TXT) 17 – Responsible person (RP) 18 – AFS cell database (AFSDB) 19 – X_25 calling address (X25) 20 – ISDN calling address (ISDN) 21 – Router (RT) 22 – NSAP address (NSAP) 23 – Reverse NSAP lookup (deprecated) (NSAP-PTR) 24 – Security signature (SIG) 25 – Security key (KEY) 26 – X.400 mail mapping (PX)) 27 – Geographical position (withdrawn) (GPOS)	28 – Ip6 Address (AAAA) 29 – Location Information (LOC) 30 – Next domain (security) (NXT) 31 – Endpoint identifier (EID) 32 – Nimrod Locator (NIMLOC) 33 – Server Selection (SRV) 34 – ATM Address (ATMA) 35 – Naming Authority Pointer (NAPTR) 36 - Key Exchanger (KX) 37 - Certificate (CERT) 38 - IPv6 Host Address (A6) 39 - Name Redirection (DNAME) 40 - Kitchen Sink (SINK) 41 - EDNS0 Option (OPT) 42 - Lists of Address Prefixes (APL) 43- Delegation Signer (DS) 249 - Transaction Key(TKEY) 250 - Transition Signature - (TSIG) 251 - Incremental Transfer - (IXFR) 252 – A request for a transfer of an entire zone (AXFR) 253 – A request for mailbox-related records (MAILB, MB, MG, or MR) 254 – A request for mail agent RRs (MAILA (Obsolete—see MX) 255 – A request for any records (ANY)
The transport layer used for the records of this table entry	*dnsServCounterTransport*	The transport that was used for these queries. 1 – The queries reported on this row were sent using UDP. 2 – The queries reported on this row were sent using TCP. 3 – The queries reported on this row were sent using a transport that was neither TCP nor UDP.

Function	MIB Variable(s)	Description
Number of queries that have been recorded in this table entry	*dnsServCounterRequests*	Number of requests (queries) that have been recorded in this row of the table. The counter information is accessed as follows: `dnsServCounterRequests.<opcode>.<class>.<type>.<transport>` The count of requested queries for IN A records over UDP by the server would be: `dnsServCounterRequests.0.1.1.`. See also "Counter information by OP Code/class/resource record type" (p. 1-26). *
Number of responses that have been recorded in this table entry	*dnsServCounterResponses*	Number of responses made by the server since initialization for the kind of query identified on this row of the table. The counter information is accessed as follows: `dnsServCounterResponses.<opcode>.<class>.<type>.<transport>` The count of query responses for IN A records over UDP by the server would be: `dnsServCounterResponses.0.1.1.1`. See also "Counter information by OP Code/class/resource record type" (p. 1-26). *
Number of queries received	*dnsServCounterRQ*	The number of queries received by this name server. *
Number of responses received	*dnsServCounterRR*	The number of responses received by this name server. *
Number of inverse queries received	*dnsServCounterRIQ*	The number of inverse queries received by this name server. *

Function	MIB Variable(s)	Description
Number of 'no such domain'' answers received	*dnsServCounterRNXD*	The number of 'no such domain' answers received by this name server. *
Number of queries received that required further processing	*dnsServCounterRFwdQ*	The number of queries received by this name server that required further processing. *
Number of responses received that answered the original query	*dnsServCounterRFwdR*	The number of responses received by this name server that answered the original query. *
Number of duplicate queries received	*dnsServCounterRDupQ*	The number of duplicate queries received by this name server. *
Number of duplicate responses received	*dnsServCounterRDupR*	The number of duplicate responses received by this name server. *
Number of SERVFAIL responses received	*dnsServCounterRFail*	The number of SERVFAIL responses received by this name server. *
Number of FORMERR responses received	*dnsServCounterRFErr*	The number of FORMERR responses received by this name server. *
Number of error responses received by this name server that were not SERVFAIL or FORMERR	*dnsServCounterRErr*	The number of error responses received by this name server that were not SERVFAIL or FORMERR. *
Number of queries received by this name server on TCP connections	*dnsServCounterRTCP*	The number of queries received by this name server on TCP connections. *
Number of zone transfer requests received by this name server	*dnsServCounterRAXFR*	The number of zone transfer requests received by this name server. *
Number of lame delegations received	*dnsServCounterRLame*	The number of lame delegations received by this name server. *

Function	MIB Variable(s)	Description
Number of packets received with IP options	*dnsServCounterROpts*	The number of packets received with IP options by this name server. *
Number of system queries sent by this name server	*dnsServCounterSSysQ*	The number of system queries sent by this name server. *
Number of answers sent	*dnsServCounterSAns*	The number of answers sent by this name server. *
Number of queries forwarded	*dnsServCounterSFwdQ*	The number of queries forwarded by this name server. *
Number of responses forwarded	*dnsServCounterSFwdR*	The number of responses forwarded by this name server. *
Number of duplicate queries sent	*dnsServCounterSDupQ*	The number of duplicate queries sent by this name server. *
Number of SERVFAIL responses sent	*dnsServCounterSFail*	The number of SERVFAIL responses sent by this name server. *
Number of FORMERR responses sent	*dnsServCounterSFErr*	The number of FORMERR responses sent by this name server. *
Number of *sendto()* system calls that failed	*dnsServCounterSErr*	The number of *sendto()* system calls that failed for this name server. *
Number of queries received by this name server that were not from name servers	*dnsServCounterRnotNsQ*	The number of queries received by this name server that were not from name servers. This number is not counted in DNS 4.0.
Number of non-authoritative answers sent	*dnsServCounterSnaAns*	The number of non-authoritative answers sent by this name server. *
Number of 'no such domain' answers sent	*dnsServCounterSNXD*	The number of 'no such domain' answers sent by this name server. *
Number of unapproved queries received	dnsServCounterRUQ	The number of unapproved queries recieved. *

Function	MIB Variable(s)	Description
Number of unapproved recursive queries received	*dnsServCounterRURQ*	The number of unapproved recursive query packets received by this name server. *
Number of unapproved AXFR or IXFR requests received	*dnsServCounterRUXFR*	The number of unapproved AXFR or IXFR requests received by this name server. *
Number of unapproved dynamic update requests received	*dnsServCounterRUUpd*	Number of unapproved dynamic update requests received.
Number of resource records added by RFC 2136 dynamic updates.	*dnsServCounterDUAdded*	The number of resource records added to this name server by RFC 2136 dynamic updates. *
Number of resource records deleted by RFC2136 dynamci updates.	*dnsServCounterDUDeleted*	The number of resource records deleted from this name server by RFC2136 dynamic updates.*
Performance/statistic counters by server		
Minimum amount of time between receiving two DNS requests	*dnsServStatMinArrivalInterval*	The minimum amount of time between receiving two DNS request messages. A message is received at the server when the server is able to begin processing the message. This typically occurs immediately after the message is read into server memory. If no messages have been received or the interval is less than 1 ms, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Maximum amount of time between receiving two DNS requests	*dnsServStatMaxArrivalInterval*	The maximum amount of time between receiving two DNS request messages. A message is received at the server when the server is able to begin processing the message. This typically occurs immediately after the message is read into server memory. If no messages have been received, this object contains a zero value. The value is in milliseconds.
Number of seconds since the last DNS request was received	*dnsServStatLastArrivalTime*	The number of seconds since the last valid DNS request message was received by the server. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value.

Function	MIB Variable(s)	Description
Minimum response time to authoritative DNS requests	*dnsServStatAuthMinResponseTime*	The smallest time interval measured as the difference between the arrival of a DNS message at the server and the successful transmission of an authoritative response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received or the interval is less than 1 ms, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Maximum response time to authoritative DNS requests	*dnsServStatAuthMaxResponseTime*	The largest time interval measured as the difference between the arrival of a DNS message at the server and the successful transmission of an authoritative response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Sum of the response times for authoritative DNS requests	*dnsServStatAuthSumResponseTime*	The sum of the response time intervals (in milliseconds), where a response time interval is measured as the difference between the arrival of a DNS message at the server and the successful transmission of an authoritative response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Minimum response time to non-authoritative DNS requests	*dnsServStatNonAuthMinResponse Time*	The smallest time interval measured as the difference between the arrival of a DNS message at the server and the successful transmission of a non-authoritative response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received or the interval is less than 1 ms, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Maximum response time to non-authoritative DNS requests	*dnsServStatNonAuthMaxResponseTime*	The largest time interval measured as the difference between the arrival of a DNS message at the server and the successful transmission of a non-authoritative response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value. The value is in milliseconds.

Function	MIB Variable(s)	Description
Sum of the response times for non-authoritative DNS requests	*dnsServStatNonAuthSumResponseTime*	The sum of the response time intervals (in milliseconds), where a response time interval is measured as the difference between the arrival of a DNS message at the server and the successful transmission of an non-authoritative response to that message. A message is received at the server when the server is able to begin processing the message. A message is transmitted after the server has no further use for the message. Note that the operating system may still have the message queued internally. The operating system queue time is not to be considered as part of the response time. Invalid messages do not cause this value to change. If no valid messages have been received, this object contains a zero value. The value is in milliseconds.
DNS traps		
The DNS Server has started	*dnsServerStarted*	Sent by the DNS server when it started.
The DNS Server has stopped	*dnsServerStopped*	Sent by the DNS server during a normal, smooth, shutdown.
The DNS has reloaded its configuration	*dnsServerReload*	Sent by the DNS server when it has been reloaded.

Function	MIB Variable(s)	Description
The DNS has detected an error in its configuration files	*dnsServerConfigError*	Sent by the DNS server when an error occurred while processing the DNS zone files. This trap will be generated only if the following errors occur: The server cannot find a file or directory that has been specified in the configuration. The server has rejected a zone that it is trying to load, due to errors.
The DNS has dumped its database to disk	*dnsServerDumped*	Sent by the DNS server when the DNS database files have been dumped to disk.

Defining Trap OIDs

Purpose

This section describes the components that comprise a trap.

`IpspgDhcpTrapEntry` and `IpspgDnsTrapEntry` define Alcatel-Lucent's Enterprise-Specific Trap contents.

> **Important!** Traps are defined for both DNS and DHCP. The examples in this section are for DNS traps, but apply identically to DHCP traps.

The following trap entry is from `named.mib`.

```
IpspgDnsTrapEntry ::= SEQUENCE {
   ipspgDnsTrIndex      INTEGER,
   ipspgDnsTrSequence   Counter32,
   ipspgDnsTrId         INTEGER,
   ipspgDnsTrText       DisplayString(SIZE(0..80)),
   ipspgDnsTrPriority   INTEGER,
   ipspgDnsTrClass      INTEGER,
   ipspgDnsTrType       INTEGER,
   ipspgDnsTrTime       Counter32,
   ipspgDnsTrSuspect    DisplayString(SIZE(0..32)),
   ipspgDnsTrDiagId     INTEGER
```

Currently used variables are as follows:

```
ipspgDnsTrIndex
ipspgDnsTrSequence
ipspgDnsTrId
ipspgDnsTrText
ipspgDnsTrPriority
ipspgDnsTrTime
ipspgDnsTrSuspect
```

Variables that are not used and are set to the ipspgDnsTrIndex are:

```
ipspgDnsTrClass,
ipspgDnsTrType,
ipspgDnsTrDiagId
```

> **Important!** Equivalent DHCP variables that are not used are set to the ipspgDhcpTrIndex.

Identifying a trap OID of `ipspgDnsTrIndex` identifies if it is a DNS trap. A trap OID of `ipspgDhcpTrIndex` identifies if it is a DHCP trap. For example, if the OID of `ipspgDnsTrIndex` is:

```
1.3.6.1.4.1.1751.1.48.2.2.1.1
```

- `ipspgDnsTrIndex` indicates which trap is received.
- `ipspgDnsTrSuspect` tells you the hostname of the servers.
- `ipspgDnsTrText` contains the textual description of the traps. So ipspgDnsTrIndex,

Variable Descriptions

The following is a list of the variables used when defining traps:

The DHCP trap names are identical, except for the substitution of `Dhcp` for `Dns` in the variable name. They function identically except as noted.

- `ipspgDnsTrIndex` tells which trap is received. For named, it is 1 to 5. 1 means named has started, 2 mens stopped etc. Please look at the end of the named.mib and look for NOTIFICATION-TYPE. There are five notification types. For DHCP there are 11 traps. Please look at *dhcp.mib.*
- `ipspgDnsTrSequence` tells how many times that specific traps is received. For example, for qddns 4.0, if you dump db using the command 'rndc dumpdb', you'll see the number will increment with each command.
- `ipspgDnsTrId` is always 1 at this time.
- `ipspgDnsTrText` contains the text message of the trap e.g. 'Lucent DNS started', 'Lucent DNS stopped' etc.
- `ipspgDnsTrPriority` contains `1` or `2` for `named`. A `1` indicates information type, `2` indicates a warning. Start, stop, reload, db dump are all priority 1 (informational) and if there are any errors in the config file, the priority is 2 (warning). qddns 4.0 sends another priority 2 trap when it finds CNAME and other data error in a zone or a file.

  Both the DNS and DHCP servers send traps with follwing priorities:

  - Priority 1 (inform) for start, stop, reload, failover returned control, subnet threshold descended and ping resonse received.
  - Priority 2 (warning) a bad packet is received.
  - Priority 3 (minor) when a subnet threshold is exceeded.
  - Priority 4 (major) when a subnet is depleted.
  - Priority 5 (critical) when a failover is activated.
- `ipspgDnsTrTime` indicates the time when the trap has occurred. It contains the number of seconds since UNIX epoch (midnight UTC of January 1, 1970).

  For example 1187282583 is 'Thu Aug 16 12:43:03 2007' local time of the sub-agent host.
- `ipspgDnsTrSuspect` is the hostname where the sub-agent (named or dhcpd) is running.

□

2 Install and configure the SNMP Module

Overview

Purpose

This chapter describes how to install the SNMP Module on Windows and UNIX platforms. It concludes with instructions on how to configure the SNMP Module.

This information presents the following topics.

Introduction

The installation of the SNMP Module involves the installation and configuration of the Lucent SNMP DHCP and DNS Agents. To successfully install and configure the agents, you must follow these steps:

1 Obtain the Alcatel-Lucent SNMP Distribution media.

2 Complete the prerequisites.

3 Back up your system before proceeding.

4 Install the DHCP and DNS SNMP Agents. (The SNMP Master Agent and utilities are installed and configured as part of this process.)

5 Start the Master SNMP Agent.

6 Verify the configuration and installation using built-in SNMP Utilities.

These tasks are described in detail in this chapter.

□

Obtain distribution media

Prior to the installation, you must obtain the Alcatel-Lucent SNMP Distribution media, See the SNMP Module Release Notes for information about obtaining the distribution media.

□

Prerequisites

See the SNMP Release Notes for information about the most current prerequisites for this product.

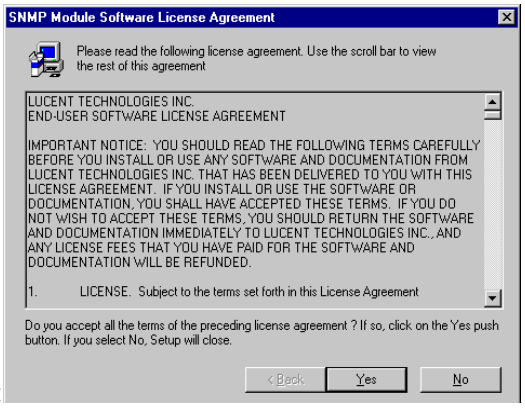
□

Install the SNMP Module on Windows

Perform the following steps to install the SNMP Module on Windows:

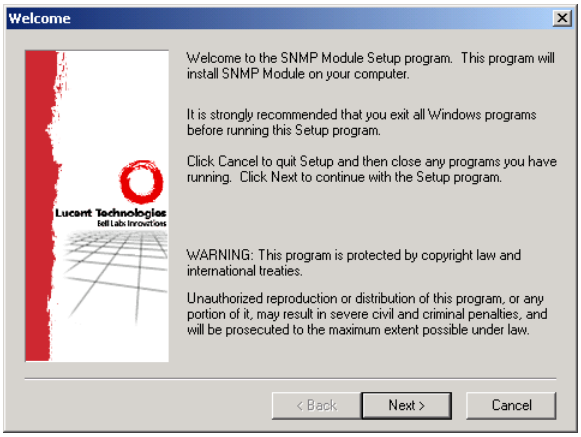
1 Exit all Windows programs that are currently running.

2 Click **Start**. Select the **Run** menu selection. Enter the drive letter and directory where the installation software has been loaded (for example, *x:\setup.exe*) and click **OK**. The SNMP Module Software License Agreement window opens, as shown in the SNMP Module Software License Agreement:

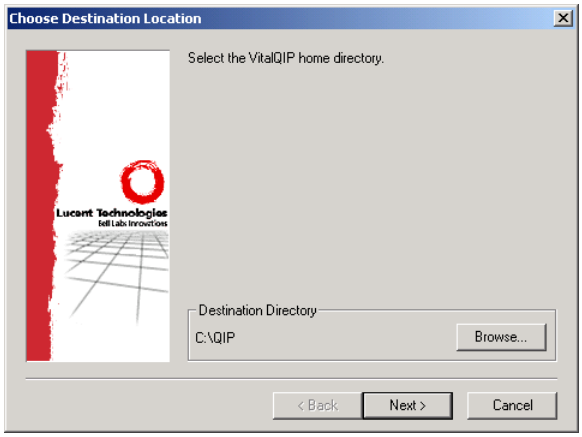


:

3 After reviewing the license agreement, click **Yes** to accept it. The Welcome window opens, as shown below:



4 After reviewing the Welcome window, click **Next**. The Choose Destination Location window opens, as shown below:

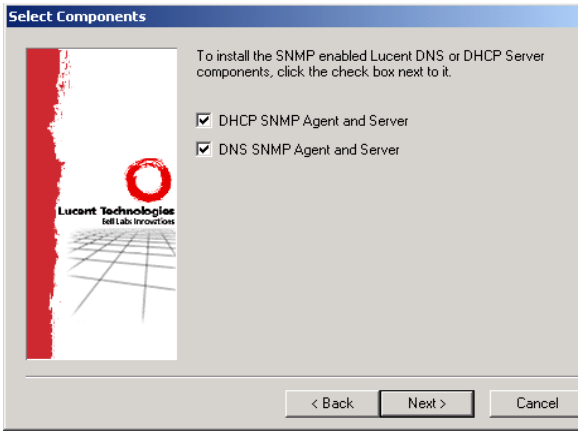


5 The Choose Destination Location window displays the directory where the VitalQIP software will be updated with the SNMP Module. Click **Browse** to select a different directory.

6 Click **Next** when the destination directory displayed indicates the appropriate location of the VitalQIP software. The installation checks your system to see if the DNS or DHCP SNMP Agents can be installed.

- If no DNS or DHCP server is installed, the message **A Lucent DNS or DHCP Server must be installed on this system prior to the installation of the SNMP Module** appears in a dialog box.
- If a version of the DNS server that might not be compatible with the SNMP install set is detected, the message **A compatible version of the Lucent DNS server was not found. The located version of the Lucent DNS Server (<version number>) must be 3.1.11 or greater. Installation of the Lucent DNS SNMP Enabled server will not be allowed** appears in a dialog box.
- If a version of the DHCP server that might not be compatible with the SNMP install set is detected, the message **A compatible version of the Lucent DHCP server was not found. The located version of the Lucent DHCP Server (<version number>) must be 5.3.8 or greater. Installation of the Lucent DHCP SNMP Enabled server will not be allowed** appears in a dialog box. Click **OK** to exit the warning message.

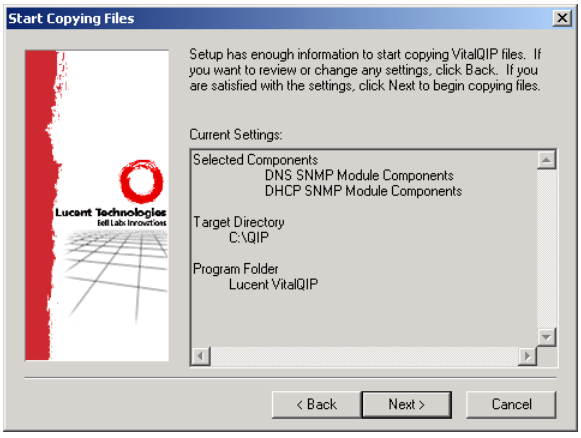
7 Click **Next** in the Choose Destination Location window and the Select Components window opens, as shown below:



The Select Components window displays the options that are available for installation on this system.

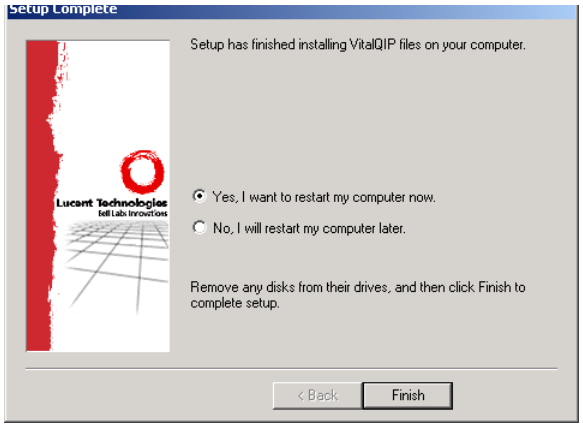
Important! This screen displays only a single DHCP or DNS SNMP Agent and Server check box if a single server was found.

8 Select or clear the check boxes to select or deselect the DHCP or DNS SNMP Agent components to be installed. Click **Next** when you have made your selections. The Start Copying Files window opens, as shown below:



9 After verifying all settings, click **Next**. Various messages and windows appear as the VitalQIP files are copied to your hard drive.

10 When the installation is completed, the Setup Complete window opens, as shown below:



11 Click **Finish** and reboot your server.

□

Environment changes

The following environment variables are defined as part of the installation:

1 The SNMP directory is added to the path.

```
path=%QIPHOME%\snmp\bin;%path%
```

2 The following path is defined for the file *snmpinfo.dat*, which is used by the SNMP Utilities:

```
SR_MGR_CONF_DIR = %QIPHOME%\snmp\config
```

3 The following path is defined for the Master Agent Configuration file *snmpd.cnf*:

```
SR_AGT_CONF_DIR = %QIPHOME%\snmp\config
```

The installation on Windows 2000/2003 creates the directory structure shown in Table 2-1:

Table 2-1 Directory structure changes

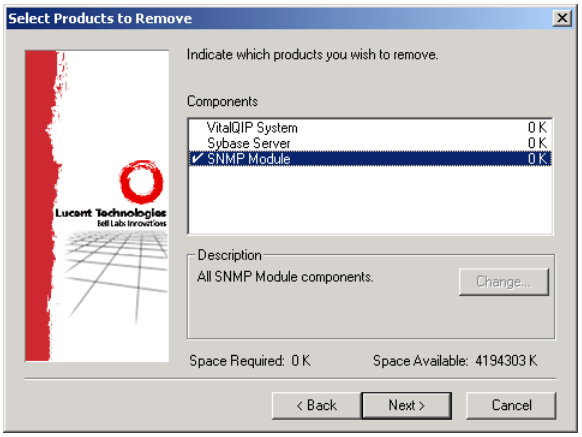
Directory	Description
%QIPHOME%\snmp\config	The directory containing the SNMP Agent configuration files.
%QIPHOME%\snmp\bin	The directory containing the SNMP Master Agent and Utilities.

Uninstall the SNMP Module from Windows

If you need to reinstall the SNMP Module, you should uninstall it first. Uninstall the SNMP Module as follows:

1 Under the Windows **Control Panel**, click **Add/Remove Programs**. The Add/Remove Programs Properties window opens.

2 Select **SNMP Module**, and click **Change/Remove**. The Select Products to Remove window opens, as shown below, asking you which components you want to remove:



3 Click **Change** to view the select sub-components screen.

4 Select **Lucent SNMP Module** then click **Continue**. A check mark appears, indicating the component to be uninstalled.

5 Click **Next** when you return to the Select Products to Remove Screen.

6 Depending on what you want to uninstall, you will be required to name the location of the files. Proceed with the instructions shown on the screen. Click **OK** at the warning screen to stop services (after stopping any SNMP services).

7 Click **Next** when the uninstall summary appears.

8 Click **Finish** when the Uninstall complete window appears, and reboot your machine.

□

Install SNMP Module on a UNIX platform

Perform the following steps to install the SNMP Module on a UNIX platform:

> **Important!** Before executing the load command, ensure the `cshrc` or `shrc` command is run to set the correct *$QIPHOME* environment.

1 To install the Lucent DHCP/DNS SNMP agents, execute the following command:

```
./qipsnmp-load
```

Executing this command displays the environment variables you are using. For example, they may appear as follows:

```
06/23/1999 15:00:02 : Start qipsnmp-load process
06/23/1999 15:00:02 : CheckUser: USER=root
06/23/1999 15:00:02 : CheckOSType: LOCAL_OSTYPE=Solaris-2.x
06/23/1999 15:00:02 : CheckEnvironment:
06/23/1999 15:00:02 : QIPHOME=/opt1/qipsnmp
```

2 If the displayed variables appear incorrect, verify that you have set the *$QIPHOME* environment as noted above.

3 Press **Enter** to continue. The License Agreement opens. Accept the License Agreement terms and the VitalQIP SNMP Product Component Load menu is displayed, as shown below:

```
################################################################################

                    Lucent Technologies, Inc.
                Copyright (c) 1999, All Rights Reserved
                     VitalQIP SNMP Version 1.0
################################################################################

                          Product Component Load
################################################################################

          1)  Media to load from         = file
          2)  Load Path                  = /home/tmp/
          3)  Load DHCP                  = yes
          4)  Load DNS                   = yes
          x)  Exit

          Are these options correct?
          Enter the option number you want to change or enter y to install:
```

Refer to Table 2-2 for descriptions of the options shown in VitalQIP SNMP Product Component Load Menu. Change the options as required.

Table 2-2 VitalQIP SNMP product component load menu options

Option	Prompt	Default	Description
1	Media to load from	file	Enter **file** when loading VitalQIP software from the FTP site or **cdrom** when loading from a CD-ROM.
2	Load Path	/home/tmp	Enter the current directory path from where the VitalQIP SNMP Software will be loaded.
3	Load DHCP	yes	Enter **y** if you want to install the DHCP SNMP Agent.
4	Load DNS	yes	Enter **y** if you wish to install the DNS SNMP Agent.

When you enter **y** at the **Are these options correct?** prompt, the general Lucent license is displayed and the product is installed. If you are prompted to kill running services, enter **y** to continue.

□

Changes made by the installation

The following changes occur as part of the installation:

1 If DHCP is selected, the library *libqsidhcpsnmp.so* is copied to *$QIPHOME/usr/lib*. The earliest version of the Lucent DHCP Server (dhcpd) that must already exist is Version 5.4, Build 18.

2 If DNS is selected, the library *libqsidnssnmp.so* is copied to *$QIPHOME/usr/lib*. The earliest version of Lucent DNS Server (*named*) that must already exist is 3.1, Build 27 or 4.0 Build 17.

In either event, the SNMP-enabling modules are copied to *$QIPHOME/snmp*.

3 The current *shrc* and *cshrc* environment files, which already exist in *$QIPHOME/etc*, are modified to include the following environment variables:

```
PATH=$QIPHOME/snmp/bin:$PATH
SR_MGR_CONF_DIR=$QIPHOME/snmp/config
SR_AGT_CONF_DIR=$QIPHOME/snmp/config
```

4 The log and template files created by this installation are located in *$QIPHOME/log*.

5 The installation on UNIX creates the directory structure shown in Table 2-3:

Table 2-3 Directory structure changes

Directory	Description
$QIPHOME/snmp/config	The directory containing the SNMP Agent configuration files.
$QIPHOME/snmp/bin	The directory containing the SNMP Master Agent and Utilities.

6 Start or restart the named and/or DHCP Services.

□

Configure the SNMP Master Agent

Before using the SNMP Module, you must configure the Master Agent as described in this section.

□

Configure additional user names

At installation time, the SNMP Master Agent configuration file (*snmpd.cnf)* is set up with a default user name of `Guest`. To use SNMPv3, a user name must be specified within an SNMPv3 Protocol Data Unit (PDU).

You may want to define additional user names in the SNMP Master Agent configuration file. If so, you can open the *snmpd.cnf* file for your specific platform in an editor.

1 If you have a UNIX platform, enter the following commands:

```
cd $QIPHOME/snmp/config
vi  snmpd.cnf
```

2 If you have a Windows platform, open a command window and enter the following command (or open the file in Notepad):

```
edit %QIPHOME%\snmp\config\snmpd.cnf
```

3 When the *snmpd.cnf* file is open, find the *usmUserEntry* section, which specifies the default user name of `Guest`:

```
usmUserEntry  localSnmpID Guest usmNoAuthProtocol\
usmNoPrivProtocol nonVolatile -
```

Additional user name entries can be made in this section by adding a new line in the format above, substituting the default `Guest` with the new user name. See Appendix A, “Increase SNMP Module security” to set security and privacy levels.

4 When new user names are added to this configuration file, security modifications must be made in the *vacmSecurityToGroupEntry* section to view the MIB:

```
vacmSecurityToGroupEntry  usm Guest SystemAdmin nonVolatile
```

```
See Appendix A, “Increase SNMP Module security” for additional
   information
```

Important! It is optional to retain the default user name `Guest`. If the default user name is not removed, SNMPv3 access using the `Guest` user name remains available.

5 You must restart the Master Agent to complete the Master Agent configuration changes.

Important! If you want to stop the SNMP Master Agent while a DNS or DHCP Server that is SNMP-enabled is running, you must stop the SNMP-enabled service(s) before stopping SNMP. Then start sthe SNMP Master Agent and then the SNMP-enabled service(s).

□

Configure notifications/traps

You can configure the Network Management station destination IP Address(es) and parameters to handle the reception of Notifications/Traps from SNMP-enabled DHCP and DNS Servers.

Important! If the Management Station is running on the same machine as the DHCP or DNS Servers, ***do not*** make any changes. The default Trap destination is set to local host.

Important! If the Network Management station is running on a different machine than the DHCP or DNS Servers to be managed, add additional lines to *snmpd.cnf* containing the IP addresses of the Management Stations in dotted decimal format. Follow the procedure to restart the Master Agent.

Multiple Management Station destinations can be defined by adding additional lines in the *snmpd.cnf* file, as detailed in the following sections.

Configure additional SNMPv1/v2c notification traps

To send SNMPv1/v2 traps to a different Management Station destination, open the *snmpd.cnf* file in an editor and follow these steps:

1 Locate the following line in the *snmpTargetAddrEntry* section:

```
snmpTargetAddrEntry  31 snmpUDPDomain 127.0.0.1:0 100 3 Console \
    v1ExampleParams nonVolatile 255.255.255.255:0
```

Important! For SNMPv2, the line will contain `v2cExampleParams` instead.

2 Copy and paste this line and then change the new `31` entry to a unique value.

3 Change the loopback address (127.0.0.1) to the IP address of the Management Station you want to receive the trap.

Important! The Network Mask entry (default: 255.255.255.255:0) can also be modified at this time. However, Alcatel-Lucent recommends that you do not change this value without first consulting your account representative.

4 Change the Description tag `Console` in the copied string to a unique Description tag.

5 Locate the following line in the *snmpNotifyEntry* section:

```
snmpNotifyEntry  31 Console trap nonVolatile
```

6 Copy and paste this line and then change the new `31` entry to the value that you assigned the *snmpTargetAddrEntry* line in Step 2 above.

7 Change the Description tag `Console` in the copied string to the unique value you selected in Step 4.

8 Save the changes to the *snmpd.cnf* file.

9 Stop the DHCP and/or DNS server.

10 Stop the SNMP Master Agent.

11 Restart the Master Agent, then the DHCP and/or the DNS server.

□

Configure additional SNMPv3 notification traps

To send SNMPv3 traps to a different Management Station destionation, open the *snmpd.cnf* file in an editor and follow these steps:

1 Locate the following line in the *snmpTargetAddrEntry* section:

```
snmpTargetAddrEntry  33 snmpUDPDomain 127.0.0.1:0 100 3 TrapSink \
    v3ExampleParams nonVolatile 255.255.255.255:0
```

2 Copy and paste this line and then change the second `33` entry to a unique value.

3 Change the loopback address (127.0.0.1) to the IP address of the Management Station you want to receive the trap.

Important! The Network Mask entry (default: 255.255.255.255:0) can also be modified at this time.

4 Change the Description tag `TrapSink` in the copied string to a unique description tag of your choice.

5 Locate the following line in the *snmpNotifyEntry* section:

```
snmpNotifyEntry  32 TrapSink trap nonVolatile
```

6 Copy and paste this line and then change the second 32 to the value that you assigned the *snmpTargetAddrEntry* line in Step 2 above.

7 Change the second TrapSink to the unique tag value in Step 4.

8 Save the changes to the *snmpd.cnf* file.

9 Stop the DHCP and/or DNS server.

10 Stop the SNMP Master Agent.

11 Restart the Master Agent, then the DHCP and/or the DNS server.

□

3 Start the SNMP Master Agent

Overview

Purpose

This chapter describes how to start the SNMP Agent and how to verify the SNMP MIB access of the DHCP/DNS server . You must perform these tasks before using the SNMP Module.

This information presents the following topics.

Start the SNMP Master Agent on UNIX

Important! Ensure the *cshrc* or *shrc* command is run to set the correct environment.

You must always start the SNMP Master Agent *before* starting either the DHCP or DNS server. If the SNMP Master agent is restarted, then the DHCP and/or DNS server must also be restarted.

Use the following command to start the SNMP Master Agent on the remote server.

```
cd $QIPHOME/etc
./qip-snmp-startup
```

If either the DHCP or DNS servers are started via the system startup script, ensure the script performs the startup in the following sequence, with the startup of your DHCP and DNS servers coming ***LAST*** (after the SNMP Master Agent):

1. Ensure the environment is set as performed by the *$QIPHOME/etc cshrc* (or *shrc*) file.

2. Start the SNMP Master Agent by using the *$QIPHOME/etc/qip-snmp-startup* file.

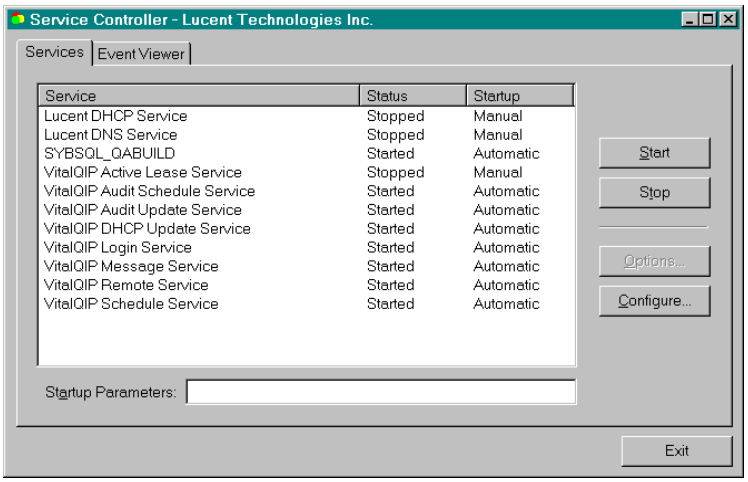
3. Start the DHCP or DNS server.

□

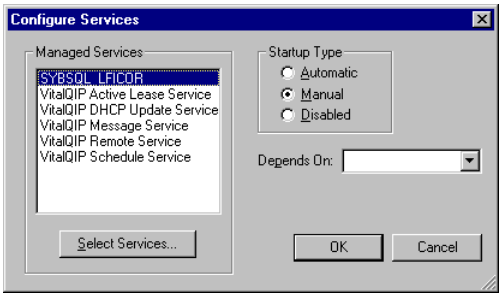
Start the SNMP Master Agent on Windows

Refer to the following steps to start the SNMP Research Master Agent on Windows. This process assumes that the installation was successful and the Registry entries were created.

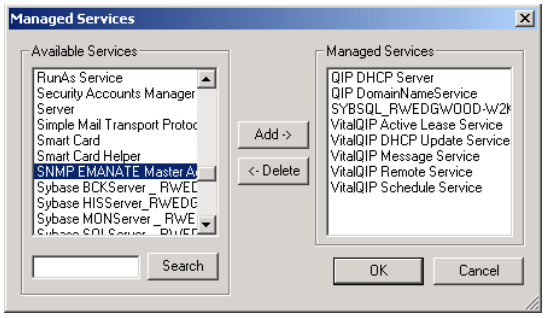
1 Start the VitalQIP Service Controller application. (This can be found in the **Start|Programs|Alcatel-Lucent VitalQIP** program group.) The Service Controller window opens as shown below:



2 Click **Configure…**. The Configure Services window opens as shown below:



3 In the Configure Services window, click **Select Services**. The Managed Services window opens as shown below:



4 In the Managed Services window, click **Search**. This populates the **Available Services** list box. Scroll down the list and highlight the **SNMP EMANATE Master Agent** entry.

5 Add this entry to the **Managed Services** list box by clicking **Add**. Click **OK** when you are done.

6 Click **OK** in the Configure Services window.

7 Start the SNMP EMANATE Master Agent by highlighting the service and clicking **Start** on the Service Controller.

8 Start the Lucent DHCP Server by highlighting the service and clicking **Start** on the Service Controller.

9 Start the Lucent DNS Server by highlighting the service and clicking **Start** on the Service Controller.

> **Important!** The SNMP Master Agent must always be started *before* starting either the DHCP or DNS server. If the master agent is restarted, then the DHCP and/or DNS server must also be restarted. You may want to create a service dependency; see your system administrator or Windows documentation for the appropriate steps.

□

Verification of DHCP/DNS server SNMP MIB access

Your installation of SNMP includes utilities to verify SNMP operations, and to ensure that the SNMP Module has been properly installed and configured according to the instructions in this manual.

To verify the operation of SNMP, complete the following steps for either UNIX or Windows. Then, read the following information about "Verification Passed" or "Verification Failed" to determine your SNMP verification.

Important! If you attempt to query the SNMP Agent for DHCP values before the initial DCHP file generation, a DHCP server error may occur. Before querying the SNMP Agent for DHCP values, ensure the DHCP Server is configured and running For information on DHCP file generation, refer to the *VitalQIP Administrator Reference Manual*.

□

Verification on a UNIX platform

To verify DHCP/DNS SNMP MIB access on a UNIX platform, follow these steps:

1 Login to the target system and obtain a terminal.

2 Ensure the *cshrc* or *shrc* command is run to set the correct $QIPHOME environment.

3 Verify that the SNMP Master Agent is running by issuing the following command:

```
ps -ef | grep snmpdm
```

4 If the SNMP Master Agent is *not* running, start it, as described in "Start the SNMP Master Agent on UNIX " (p. 3-2).

5 Verify that the DHCP and DNS servers are started. If they are not started, start them according to the instructions in the *VitalQIP User's Guide for UNIX*.

6 For DNS, run the test script file:

```
$QIPHOME/snmp/bin/TestDNS
```

Important! This test requires that *localhost* represents the current Hostname

The following output displays for a successful test:

```
SUCCESS:
Dns Server Subagent Test
dnsServSystemDescr.0 = Lucent QDDNS <VERSION_INFO>
```

The following output displays for an unsuccessful test:

```
FAILURE:
Error code set in packet - No such variable name.  Index:  1.
```

7 For DHCP, run the test script file:

```
$QIPHOME/snmp/bin/TestDHCP
```

Important! This test requires that *localhost* represents the current Hostname

The following output displays for a successful test:

```
SUCCESS:
DHCP Server Subagent Test
dhcpServSystemDescr.0 =  (Version: <VERSION_INFO> - Lucent
   DHCP Server)
```

The following output displays for an unsuccessful test:

```
FAILURE:
Error code set in packet - No such variable name.  Index:  1.
```

Verification on Windows

To verify the DHCP/DNS Server SNMP access on a Windows platform, follow these steps:

1 Verify that the SNMP Master Agent is running.

2 From the Control Panel, select **Services**.

3 Scroll through the list of services and verify that SNMP EMANATE Master Agent is in the Started state.

4 Verify that the DHCP and DNS servers are started. If they are not started, start them according to the instructions in the *VitalQIP User's Guide for Windows*.

5 From a MS-DOS Command window:

6 For DNS run the test batch file from the home directory:

```
%QIPHOME%\snmp\bin\TestDNS
```

Important! This test requires that *localhost* represents the current Hostname

The following output displays for a successful test:

```
SUCCESS:
dnsServSystemDescr.0 = Lucent QDDNS <VERSION_INFO>
```

The following output displays for an unsuccessful test:

```
FAILURE:
Error code set in packet - No such variable name.  Index:  1.
```

7 For DHCP run the test batch file from the home directory:

```
%QIPHOME%\snmp\bin\TestDHCP
```

Important! This test requires that *localhost* represents the current Hostname

The following output displays for a successful test:

```
SUCCESS:
dhcpServSystemDescr.0 =  (Version: <VERSION_INFO> - Lucent
   DHCP Server)
```

The following output displays for an unsuccessful test:

```
FAILURE:
Error code set in packet - No such variable name.  Index:  1.
```

If Verification Passes

If verification passes, the DHCP or DNS service description and version are returned.

If Verification Fails

If verification fails, the DHCP or DNS service description and version are NOT returned.

□

Additional information

The Lucent DHCP and DNS MIB files are automatically installed on your system during the installation of the SNMP Module. These files can be used to load into a network management system, such as HP Openview.

On Windows NT/2000, the file locations are:

```
%QIPHOME%\snmp\config\dhcp.mib (DHCP MIB)
%QIPHOME%\snmp\config\named.mib (DNS MIB)
```

On UNIX, the file locations are:

```
$QIPHOME/snmp/config/dhcp.mib (DHCP MIB)
$QIPHOME/snmp/config/named.mib (DNS MIB)
```

The Lucent DHCP and DNS MIBs are vendor specific.

□

Appendix A: Increase SNMP Module security

Overview

Purpose

This appendix covers adding additional security to the SNMP Module. Additional security can be added by limiting SNMPv1 and SNMPv2c access to MIB variables. User name and passwords can be protected in SNMPv3.

This information presents the following topics.

Limit SNMPv1 and SNMPv2c access to MIB variables

MIB variables can be queried by anyone who knows the default community. The The SNMP Module restricts "public" access so the majority of the configuration tree is unavailable to "public" access. This restricted "public" access may not be enough in some environments.

In cases where restricted "public" access is not enough, modifying the *<VitalQIP_directory>/snmp/config/snmpd.cnf* file can prevent the "public" community from receiving answers to queries. To do so, search the *<VitalQIP_directory>/snmp/config/snmpd.cnf* file for two "Anyone" entries under the *acmAccessEntry* section and modify them as follows:

```
vacmAccessEntry Anyone  - snmpv1 noAuthNoPriv \
    exact - - CfgProt nonVolatile
vacmAccessEntry Anyone - snmpv2c noAuthNoPriv \
exact - - CfgProt nonVolatile
```

It is not recommended to delete the "public" community. Additional communities can be defined as described in the procedures below.

To change the community name string, follow these steps:

1. With a text editor, open the *<VitalQIP_directory>/SNMP/config/snmpd.cnf* file.

2. In the *snmpd.cnf* file, define a group and its associated access right in the **vacmAccessEntry** section. The following example uses "group1" for the new group:

```
#Entry type: vacmAccessEntry
#Format:  vacmGroupName  (text)
#         vacmAccessContextPrefix  (text)
#         vacmAccessSecurityModel  (snmpv1, snmpv2c, snmpv2s, usm)
#         vacmAccessSecurityLevel  (noAuthNoPriv, authNoPriv, authPriv)
#         vacmAccessContextMatch  (exact, prefix)
#         vacmAccessReadViewName  (text)
#         vacmAccessWriteViewName  (text)
#         vacmAccessNotifyViewName  (text)
#         vacmAccessStorageType  (nonVolatile, permanent, readOnly)
vacmAccessEntry  group1 - snmpv1 noAuthNoPriv exact All All All nonVolatile
vacmAccessEntry  group1 - snmpv2c noAuthNoPriv exact All All All nonVolatile
```

3. Associate the community name to the group in the **vacmSecurityToGroupEntry** section. The following example uses "CommunA" as the community name:

```
#Entry type: vacmSecurityToGroupEntry
#Format:  vacmSecurityModel  (snmpv1, snmpv2c, snmpv2s, usm)
#         vacmSecurityName  (text)
#         vacmGroupName  (text)
#         vacmSecurityToGroupStorageType  (nonVolatile, permanent,
   readOnly)
vacmSecurityToGroupEntry  snmpv1 CommunA group1 nonVolatile
vacmSecurityToGroupEntry  snmpv2c CommunA group1 nonVolatile
```

4 To add the new community (in the above example CommunA), add an entry to the **snmpCommunityEntry** section. The **snmpCommunityEntry** section contains a list of all active communities. Increment the **snmpCommunityIndex**. For SNMP v1/v2c, the **snmpCommunityName** and the **snmpCommunitySecurityName** are the same, and both values should contain the new community name. The **snmpCommunityTransportTag** contains a dash (-) unless an Access Control List (ACL) value is required. In the following example, an ACL value called `CommunALocation` is used (and is set up in the next step):

```
##Entry type: snmpCommunityEntry
#Format: snmpCommunityIndex (text)
#        snmpCommunityName (text)
#        snmpCommunitySecurityName (text)
#        snmpCommunityContextEngineID (octectString)
#        snmpCommunityContextName (text)
#        snmpCommunityTransportTag (text)
#        snmpCommunityStorageType (nonVolatile, permenant, readOnly)
snmpCommunityEntry tOOOOOOO public public localSnmpID - - nonVolatile
snmpCommunityEntry tOOOOOO1 CommunA CommunA localSnmpID - \
   CommunALocation nonVolatile
```

5 ***Optional.*** To further restrict the use of this community from a specific workstation in the **snmpTargetAddrEntry** section, add a line similar to the following example (the example uses "`CommunALocation`" as the name of the ACL value):

```
#Entry type: snmpTargetAddrEntry
#Format:  snmpTargetAddrName  (text)
#         snmpTargetAddrTDomain  (snmpUDPDomain, snmpIPXDomain, etc.)
#         snmpTargetAddrTAddress  (transport address,such as
#    192.147.142.254:0)
#         snmpTargetAddrTimeout  (integer)
#         snmpTargetAddrRetryCount  (integer)
#         snmpTargetAddrTagList  (text)
#         snmpTargetAddrParams  (text)
#         snmpTargetAddrStorageType  (nonVolatile, permanent, readOnly)
#         snmpTargetAddrTMask  (transport mask, i.e. 255.255.255.255:0)
```

```
#         snmpTargetAddrMMS  (integer)
snmpTargetAddrEntry  81 snmpUDPDomain 10.55.255.100:0 100 3 CommunALocation \
    v1ExampleParams nonVolatile 255.255.255.255:0 2048
snmpTargetAddrEntry  82 snmpUDPDomain 10.55.255.100:0 100 3 CommunALocation \
    v2cExampleParams nonVolatile 255.255.255.255:0 2048
```

6 Save the *snmpd.cnf* file.

7 Restart the master agent and services.

□

SNMPv3 security

SNMPv3 offers additional security features that are not available in earlier versions of the SNMP protocol. SNMPv3 supports the use of a user name and password protection. Authentication and privacy protocols are also supported to further protect the SNMP packets on the network. Setting these protocols provides additional security if SNMP packets are not filtered by a firewall.

Setting user name and password protection for SNMPv3

Basic SNMPv3 security can be added by setting up user name and password protections. This does not involve the use of authentication and privacy protocols. For more information about setting up SNMPv3 security with authentication and privacy protocols, see "Set authentication and privacy protocols for SNMPv3" (p. A-11). Otherwise, follow the steps in the section to establish basic user name and password protection.

Gather information and considerations

Before setting the user name and password protection, gather the following::

- User name
- User Group
- Station location/name
- IP address of monitoring console
- Engine ID

Important! The engine ID is different for each host. The engine ID can be obtained by querying the server with

```
getone -v1 <IP_address_of_station_querying> public
   snmpEngineID.0.
```

The snmpEngineID is written to the configuration files as a colon delimited value, such as 00:00:00:63:00:00:00:A1:0A:00:00:03.

Also, consider if the user should have Get Access privileges, have Set Access privileges, and receive traps.

Set user privileges

To set user privileges, follow these steps:

1 With a text editor, open the *<VitalQIP_directory>/snmp/config/snmpd.cnf* file on the DHCP/DNS server running the Master Agent.

2 In the *snmpd.cnf* file, add the following lines:

```
usmUserEntry localSnmpID <user_name> usmNoAuthProtocol usmNoPrivProtocol \
   nonVolatile \
    <station_location/name> - -
vacmAccessEntry <group> - usm noAuthNoPriv exact All - All nonVolatile
vacmSecurityToGroupEntry usm <user_name> <group> nonVolatile
vacmViewTreeFamilyEntry All iso - included nonVolatile
snmpTargetAddrEntry <arbitrary_tag> snmpUDPDomain \
   <monitoring_console's_IP_addresss>:0 \
     100 3 <station_name/location> v3ExampleParams nonVolatile \
   255.255.255.255:0 2048
```

Important! In most cases, the **vacmViewTreeFamilyEntry** line is already in the *snmpd.cnf* file.

When the SNMP Master Agent is restarted, all lines are rewritten under the correct headings. It does not matter where the lines are entered in the file.

User privileges can vary from those shown above. User permissions can be set by changing the parameters of the **vacmAccessEntry** line. Change the parameters as follows; in all cases, the "All" value sets the privilege and "-" denies the privilege:

- For the Get Access privilege, set the sixth parameter.
- For the Set Access privilege, set the seventh parameter.
- For the Trap Sending privilege, set the eighth parameter.

3 If the user is required to receive traps, set the trap Sending privilege to **All** and add the following lines:

```
snmpNotifyEntry <arbitrary_tag> <station_name/location> nonVolatile
snmpTargetParamsEntry v3ExampleParams 3 usm <user_name> noAuthNoPriv \
   nonVolatile
```

4 Save the *snmpd.cnf* file.

Restart the Master Agent

If the user is set to send traps, the trap receiving station can be set to collect and decode traps from the user's DHCP/DNS server. To configure the receiving station, follow these steps:

1 With a text editor, open the *<VitalQIP_directory>/snmp/config/mgr.cnf* file on monitor console running the Master Agent.

2 In the *mgr.cnf* file, add the following lines:

```
usmUserEntry <engine_ID> <user_name> \
usmNoAuthProtocol usmNoPrivProtocol nonVolatile - - -
```

3 Save the *mgr.cnf* file.

4 Start TRAPRCV or other trap catching utility.

□

Set authentication and privacy protocols for SNMPv3

SNMPv3 authentication and privacy protocols can be used to secure SNMP queries. Authentication and privacy protocols are a more advanced security feature than user name and password protection. These protocols prevent unauthorized users from altering in-transit SNMP messages generated on behalf of an authorized user and prevent another user from assuming the identity of another user that has the appropriate authorizations. For in-depth information about these protocols, refer to RFC 2574. RFC 2574 can be obtained at **http://www.ietf.org**

To set up SNMPv3 authentication and privacy protocols, use the steps in this section.

Gather information

Before beginning, gather the following information:

- User name
- User group
- Authentication protocol - None, MD5, or SHA
- Privacy Protocol - None or DES
- The MIB tree levels available for viewing - None, All, or another view defined in **vacmViewTreeFamilyEntry** section
- Monitoring station location
- Monitoring station ID tag
- Monitoring station IP address
- SNMPv3 user parameter tag
- Engine ID

Important! The engine ID is different for each host. The engine ID can be obtained by querying the server with

```
getone -v1 <IP_address_of_station_querying> public snmpEngineID.0.
```

The snmpEngineID is written to the configuration files as a colon delimited value, such as 00:00:00:63:00:00:00:A1:0A:00:00:03.

Set up the user

To set up a user, follow these steps:

1 With a text editor, open the *<VitalQIP_directory>/snmp/config/snmpd.cnf* file.

2 Add or modify the following sections:

```
usmUserEntry snmpEngineID <user_name> \
usmNoAuthProtocol|usmHMACMD5AuthProtocol|usmHMACSHAAuthProtocol \
   usmNoPrivProtocol|usmDESPrivProtocol nonVolatile \
   <monitoring_station_location> \
   <authentication_password_or_-_for_no_password> \
   <privacy_password_or_-_for_none>
vacmAccessEntry <name_of_user_group> <context_prefix_leave_as_a_-> usm \
   noAuthNoPriv|authNoPriv|authPriv exact <view_level_for_get> \
   <view_level_for_set> \
   <view_level_for_traps> nonVolatile
vacmSecurityToGroupEntry usm <user_defined_in_usmUserEntry> \
   <group_defined_in_vacmAccessEntry> \
   nonVolatile
```

Important! The authentication and privacy passwords are encrypted when **snmpd** is started.

Once one group is set up, several users with the same authentication and privacy levels can be associated to it.

Take care when adding lines to the **vacmAccessEntry** section of *snmpd.cnf*. This section sets the minimum security allowed for user groups. If you assign an **authPriv** user to a group that has **noAuthNoPriv** attribute, that user will be able to query using both the defined **authPriv** passwords and NULL passwords.

The following example shows a user set up to view the entire MIB variable tree:

```
usmUserEntry localSnmpID User usmHMACMD5AuthProtocol usmDESPrivProtocol \
   nonVolatile - AuthPass \
   PrivPass
vacmAccessEntry Group - usm AuthPriv exact All - - nonVolatile
vacmSecurityToGroupEntry usm User Group nonVolatile
```

3 If needed, you can restrict a user to only perform queries from a specific workstation by adding the monitoring station's location in the **usmUserEntry** section. The **snmpTargetAddrEntry** section must be modified to include the monitoring station's location as follows:

```
snmpTargetAddrEntry <monitoring_station_ID_tag> snmpUDPDomain \
   <monitoring_station_IP_address>:0 100 3 <monitoring_station_location> \
   <v3_user_parameter_tag>\
   nonVolatile 255.255.255.255:0
```

The following example is of a user who is limited to reading the MIB variables from the workstation 10.55.0.4. The example uses MD5 authentication and DES privacy protocols.

```
usmUserEntry localSnmpID User usmHMACMD5AuthProtocol usmDESPrivProtocol \
```

```
   nonVolatile \
   Workstation AuthPass PrivPass
vacmAccessEntry Group - usm AuthPriv exact All - - nonVolatile
vacmSecurityToGroupEntry usm User Group nonVolatile
snmpTargetAddrEntry 98 snmpUDPDomain 10.55.0.4:0 0 0 Workstation - \
   nonVolatile 255.255.255.255:0
```

4 Save the *snmpd.cnf* file.

5 Stop and restart **snmpd** for the changes to take effect.

Once the SNMPv3 is configured, the user will be prompted for a name and password when doing a SNMPv3 query. The **getone** utility included with the SNMP distribution checks the *<VitalQIP_directory>/snmp/config/mgr.cnf* file for information about to the user. A user without an entry is prompted for authentication and privacy passwords for each query. By default, the SNMP master agent assumes that authentication is in the form of clear text or MD5.

Important! While the **getone** utility offers the password prompts for SHA authentication users, it does not authenticate them. The SHA password must be added to the *mgr.cnf* file as described below.

The need to type passwords can be eliminated by adding the user authentication protocol, privacy protocol, and passwords. This is required in order to use SHA authentication. To do so, add or modify the following sections in the *<VitalQIP_directory>/snmp/config/mgr.cnf* file:

```
usmUserEntry snmpEngineID User_from_previous_section \
    usmNoAuthProtocol|usmHMACMD5AuthProtocol|usmHMACSHAAuthProtocol \
    usmNoPrivProtocol|usmDESPrivProtocol  nonVolatile \
   <monitoring_station_location> \
    "Authentication_Password_in_Quotes" "Privacy_Password_in_Quotes"
```

The following example is of a modified **usmUserEntry** section using SHA authentication on a remote SNMP monitoring station:

```
usmUserEntry 00:00:00:63:00:00:00:a1:0a:33:00:05 \
   User usmHMACSHAAuthProtocol \
   usmDESPrivProtocol  nonVolatile - "AuthPass" "PrivPass"
```

Configure to receive and send SNMP traps

Traps can be sent from the server using the same privacy protocols that are used to get and set MIB variables. Be aware that using users added in "2. Set up the User" is discouraged as the user may be restricted to using a certain workstations. Instead, create separate users with trap privileges to secure SNMP against unauthorized MIB variable queries from a restricted IP address. To do so, follow these steps:

1 Using a text editor, open the *<VitalQIP_directory>/snmp/config/snmpd.cnf* file.

2 Add or modify the following section as follows:

```
usmUserEntry snmpEngineID User_Name \
   usmNoAuthProtocol|usmHMACMD5AuthProtocol|usmHMACSHAAuthProtocol \
   usmNoPrivProtocol|usmDESPrivProtocol nonVolatile \
   <monitoring_station_location> \
   <authentication_password_or_-_for no password> \
    <privacy_password_or_-_for_none>
vacmAccessEntry <name_of_user_group> <context_prefix_leave_as_a_- usm> \
   noAuthNoPriv|authNoPriv|authPriv exact <view_level_for_get> \
   <view_level_for_set> <view_level_for_traps> nonVolatile
vacmSecurityToGroupEntry usm <user_defined_in_usmUserEntry> \
   <group_defined_in_vacmAccessEntry> \
   nonVolatile
snmpNotifyEntry <monitor_station_ID_tag> <monitor_station_location> trap \
   nonVolatile
snmpTargetAddrEntry <monitoring_station_ID_tag> snmpUDPDomain \
   <monitoring_station_IP_address>:0 100 3 <monitoring_station_location> \
   <v3_user_parameter_tag> \
   nonVolatile 255.255.255.255:0
snmpTargetParmEntry <v3_user_parameters_tag> 3 usm <user_name> \
   NoAuthPriv|authNoPriv|authPriv nonVolatile
```

Important! The authentication and privacy passwords are encrypted when **snmpd** is started.

Once one group is set up, several users with the same authentication and privacy levels can be associated to it.

The following is an example of a modified *snmpd.cnf* file. The user is Trapper and can only receive traps using MD5 and DES privacy protocol. The monitoring station IP address is 10.55.0.5, its location is northCampus, and its tag is 99.

```
usmUserEntry localSnmpID Trapper usmHMACMD5AuthProtocol usmDESPrivProtocol \
   nonVolatile - \
   AuthPass PrivPass
vacmAccessEntry Group - usm AuthPriv exact - - All nonVolatile
vacmSecurityToGroupEntry usm Trapper Group nonVolatile
snmpNotifyEntry 99 northCampus trap nonVolatile
snmpTargetAddrEntry 99 snmpUDPDomain 10.55.0.5:0 100 3 northCampus \
   v3TrapperParams nonVolatile \
   255.255.255.255:0 \
snmpTargetParamsEntry v3TrapperParams usm Trapper AuthPriv nonVolatile
```

3 Save the *snmpd.cnf* file.

4 Stop and restart **snmpd** to begin broadcasting traps.

After the server is configured, all trap receiving utilities must be configured to receive traps. With the SNMP Module, the *traprcv* utility is distributed to test functionality of the trap process. It is not intended to be used as a monitoring and reporting tool. To configure this utility to obtain the user's information defined above, follow these steps:

5 With a text file editor, open the *<VitalQIP_directory>/snmp/config/mgr.cnf* file.

6 In the **usmUserEntry** section, add the user as shown in the following example:

```
UsmUserEntry 00:00:00:63:00:00:00a1:0a:33:00:05 Trapper \
   UsmHMACMD5AuthProtocol \
   usmDESPrivProtocol nonvolatile - AuthPass PrivPass
```

7 Save the *mgr.cnf* file.

□

Index