VitalQIP® DNS/DHCP & IP Management Software

SNMP MODULE | VERSION 2.3

USER'S GUIDE

190-409-038R2.3
ISSUE 1
APRIL 2009

Licenses

Refer to Appendix C, “Third party software license statements” in the *VitalQIP Release 7.2 Installation Guide (190-409-043R7.2)* for a complete description of all software licenses used to develop this product.

Contents

About this document

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.

Reason for revision

The following table shows the revision history of this document.

Issue	Feature	Description	Feature impact
1	New DNS MIB	New DNS MIBs were added and older variables were removed.	Table , "Lucent DNS MIB variables" (p. 17)
1	New SNMP Module installation	There is a new SNMP Module installation.	• Procedure , "Install the SNMP Module on Windows" (p. 2-3) • Procedure , "Install SNMP Module on a UNIX platform" (p. 2-13)
1	Console and template installation	This chapter provides information about installing SNMP from the console and the template	Chapter 4, "Console and template installations"
1	Trap object Ids (OIDs)	The "Trap object Ids" section has been revamped. Newer examples have been added.	"Trap object IDs (OIDs)" (p. 1-37)

Intended audience

This document 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:

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

How to use this document

The manual is organized as follows:

Chapter	Description
Chapter 1, “SNMP Support for the Lucent DHCP and DNS Servers”	Describes how 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”	Describes the tasks required to install and configure the Lucent SNMP DHCP and DNS Agents.
Chapter 3, “Start the SNMP Master Agent”	Describes how to start the SNMP Agent and how to verify SNMP operations.
Chapter 4, “Console and template installations”	Describes how to install SNMP from the console and the template.
Appendix A, “Increase SNMP Module security”	Describes how to increase security for the SNMP Module.

Conventions used

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

Convention	Meaning	Example
boldface	Names of items on screens. Names of commands and routines Names of buttons you should click. Uniform Resource Locators (URLs)	Select the **Client** check box. The **qip_getappllst** routine returns the entire list of existing applications. Click **OK**. The VitalQIP product site can be found at **http://www.alcatel-lucent.com/wps/portal/products**.
Helvetica bold	Names of keys on the keyboard to be pressed.	Press **Enter** to continue.
Letter Gothic	Output from commands, code listings, and log files	`# Name: Share shared-network _200_200_200_0`
Letter Gothic bold	Input that you should enter from your keyboard.	Run the following command: **`c:\setup.exe`**
<angle brackets>	Variables that you must substitute another value for.	*<debugfile>.bak.log*
italics	Manual and book titles. Directories, paths, file names, and e-mail addresses.	Refer to the *VitalQIP User's Guide* for more information. A symbolic link must be created from */etc/named.conf* that points to *named.conf*.
bold italic	Emphasis	***Read-only***. The name of the service element.
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.
double-click	Double-click the left button on your mouse.	Double-click the book icon.

Related information

Use the *VitalQIP Administrator Reference Manual* (part number: 190-409-042) with this product. This guide describes planning and configuring your network, information about the VitalQIP interface, advanced DNS and DHCP configurations, and troubleshooting.

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:

Region	Address	Contact information
North 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, and Africa	Alcatel-Lucent Voyager Place Shoppenhangers Road Maidenhead Berkshire SL6 2PJ 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 Brasil S/A Avenida Marginal Direita Anchieta, 400 - Km 11,5 CEP: 04182-901 - Jardim Santa Cruz - Sao Paulo - SP Brazil	Phone: 0800 89 19325 or +55 11 3205 7626 For other local CALA numbers, consult the web site **https://support.lucent.com** or contact your local sales representative.
Asia Pacific	Alcatel-Lucent Australia 280 Botany Road Alexandria NSW 2015 Australia	Phone: 1800-458-236 (toll free from within Australia) (IDD) 800-5823-6888 (toll free from Asia Pacific - China, 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 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 access additional VitalQIP manuals online at **https://support.lucent.com**. Select **Documentation** from the **Customer Support** menu and click the **Product index** link. Click the **V** link to access VitalQIP and VitalQIP add-on products.

How to comment

To comment on this , go to the Online Comment Form (**http://www.lucent-info.com/comments/**) 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

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.

Alcatel-Lucent 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 Alcatel-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”.
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.

Function	MIB Variable(s)	Description
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.
Number of Bootp Packets dropped with unknown clients	*dhcpServBootpCountDropped UnknownClients*	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	*dhcpServBootpCountDroppedNot ServingSubnet*	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.

Function	MIB Variable(s)	Description
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.
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	*dhcpServDhcpCountDroppedUnknown Client*	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	*dhcpServDhcpCountDroppedNot ServingSubnet*	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.
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.

Function	MIB Variable(s)	Description
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.
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.
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:		

Function	MIB Variable(s)	Description
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.
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.
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.
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.

Function	MIB Variable(s)	Description
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.
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 time-out (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:		
Note: 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.
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.

Function	MIB Variable(s)	Description
Indicates the status of the partner server	*dhcpServFailoverPartnerStatus*	This variable indicates 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 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 response from each primary server.
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.

Function	MIB Variable(s)	Description
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.
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.

Function	MIB Variable(s)	Description
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. 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”.
Counter information by OP Code/class/resource record type:		

Function	MIB Variable(s)	Description
	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.
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

Function	MIB Variable(s)	Description
The DNS Record Type of this table entry	*dnsServCounterQType*	

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)

Function	MIB Variable(s)	Description
		256 - DNSSEC Look aside validation (DLV)
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.
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. 1-17). *

Function	MIB Variable(s)	Description
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. 1-17). *
IPv4 requests received	*dnsServCounterRequestv4*	IPv4 requests received. This also counts non-query requests.
IPv6 requests received	*dnsServCounterRequestv6*	IPv6 requests received. This also counts non-query requests.
Requests with EDNS(0) received	*dnsServCounterReqEdns0*	Requests with EDNS(0) received.
Requests with unsupported EDNS version received	*dnsServCounterReqBadEDNSVer*	Requests with unsupported EDNS version received.
Requests with TSIG received	*dnsServCounterReqTSIG*	Requests with TSIG received.
Requests with SIG(0) received	*dnsServCounterReqSIG0*	Requests with SIG(0) received.
Requests with invalid TSIG or SIG(0) signature received	*dnsServCounterReqBadSIG*	Requests with invalid TSIG or SIG(0) signature received.
TCP requests received	*dnsServCounterReqTCP*	TCP requests received.
Authoritative (non-recursive) queries rejected.	*dnsServCounterAuthQryRej*	Authoritative (non-recursive) queries rejected.
Recursive queries rejected	dnsServCounterRecQryRej	Recursive queries rejected.

Function	MIB Variable(s)	Description
Zone transfer requests rejected	*dnsServCounterXfrRej*	Zone transfer requests rejected.
Dynamic update requests rejected	*dnsServCounterUpdateRej*	Dynamic update requests rejected.
Responses sent	*dnsServCounterResponse*	Responses sent.
Truncated responses sent	*dnsServCounterTruncatedResp*	Truncated responses sent
Responses with EDNS(0) sent	*dnsServCounterRespEDNS0*	Responses with EDNS(0) sent.
Responses with TSIG sent	*dnsServCounterRespTSIG*	Responses with TSIG sent.
Responses with SIG(0) sent	*dnsServCounterRespSIG0*	Responses with SIG(0) sent.
Queries resulted in successful answer.	*dnsServCounterQrySuccess*	Queries resulted in successful answer. This means the query which returns a NOERROR response with at least one answer resource record.
Queries resulted in authoritative answer	*dnsServCounterQryAuthAns*	Queries resulted in authoritative answer.
Queries resulted in non-authoritative answer	*dnsServCounterQryNoauthAns*	Queries resulted in non-authoritative answer.
Queries resulted in referral answer	*dnsServCounterQryReferral*	Queries resulted in referral answer.
Queries resulted in in NOERROR responses	*nsServCounterQryNxrrset*	Queries resulted in in NOERROR responses with no data
Queries resulted in SERVFAIL	*dnsServCounterQrySERVFAIL*	Queries resulted in SERVFAIL.
Queries resulted in FORMERR	*dnsServCounterQryFORMERR*	Queries resulted in FORMERR.
Queries resulted in NXDOMAIN	*dnsServCounterQryNXDOMAIN*	Queries resulted in NXDOMAIN.
Queries which caused the server to perform recursion	*dnsServCounterQryRecursion*	Queries which caused the server to perform recursion in order to find the final answer.

Function	MIB Variable(s)	Description
Queries which the server attempted to recurse	*dnsServCounterQryDuplicate*	Queries which the server attempted to recurse but discovered an existing query with the same IP address, port, query ID, name, type and class already being processed.
Queries for which the server discovered an excessive number of existing recursive queries	*dnsServCounterQryDropped*	Queries for which the server discovered an excessive number of existing recursive queries for the same name, type and class and were subsequently dropped.
Other query failures	*dnsServCounterQryFailure*	Other query failures.
Requested zone transfers completed	*dnsServCounterXfrReqDone*	Requested zone transfers completed.
Dynamic update requests forwarded	*dnsServCounterUpdateReqFwd*	Dynamic update requests forwarded.
Dynamic update responses forwarded	*dnsServCounterUpdateRespFwd*	Dynamic update responses forwarded.
Dynamic update forward failed	*dnsServCounterUpdateFwdFail*	Dynamic update forward failed.
Dynamic updates completed	*dnsServCounterUpdateDone*	Dynamic updates completed.
Dynamic updates failed	*dnsServCounterUpdateFail*	Dynamic updates failed.
Dynamic updates rejected	*dnsServCounterUpdateBadPrereq*	Dynamic updates rejected due to prerequisite failure.
IPv4 notifies sent	*dnsServCounterZoneMaintNotifyOutv4*	IPv4 notifies sent.
IPv6 notifies sent	*dnsServCounterZoneMaintNotifyOutv6*	IPv6 notifies sent.
IPv4 notifies received	*dnsServCounterZoneMaintNotifyInv4*	IPv4 notifies received.
IPv6 notifies received	*dnsServCounterZoneMaintNotifyInv6*	IPv6 notifies received.
Incoming notifies rejected	*dnsServCounterZoneMaintNotifyRej*	Incoming notifies rejected.
IPv4 SOA queries sent	*dnsServCounterZoneMaintSOAOutv4*	IPv4 SOA queries sent.
IPv6 SOA queries sent	*dnsServCounterZoneMaintSOAOutv6*	IPv6 SOA queries sent.
IPv4 AXFR requested	*dnsServCounterZoneMaintAXFRReqv4*	IPv4 AXFR requested.
IPv6 AXFR requested	*dnsServCounterZoneMaintAXFRReqv6*	IPv6 AXFR requested.

Function	MIB Variable(s)	Description
IPv4 IXFR requested	*dnsServCounterZoneMaintIXFRReqv4*	IPv4 IXFR requested.
IPv6 IXFR requested	*dnsServCounterZoneMaintIXFRReqv6*	IPv6 IXFR requested.
Zone transfer requests succeeded	*dnsServCounterZoneMaintXfrSuccess*	Zone transfer requests succeeded.
Zone transfer requests failed	*dnsServCounterZoneMaintXfrFail*	Zone transfer requests failed.
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.
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	*dnsServStatNonAuthMinResponseTime*	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.
Object defined configuration and settings:		
Seconds elapsed when the server is in a persistent state	*dnsServConfigUpTime*	If the server has a persistent state (for instance, a process), this value is the seconds elapsed since it started. For software without persistent state, this value is zero.

Function	MIB Variable(s)	Description
Seconds elapsed since the last time the name server was reset.	*dnsServConfigResetTime*	If the server has a persistent state (for instance, a process) and supports a reset operation (for instance, can be told to re-read configuration files), this value is the seconds elapsed since the last time the name server was reset. For software that does not have persistence state or does not support a reset operation, this value is zero.
Server statistics and zone maintenance statistics are not counted	*dnsServConfigQddnsSnmpStats*	If the policy **`snmp-stats`** is set to **`no`** in the **`qddns`** block, server statistics and zone maintenance statistics are not counted. All dnsServCounters values are always set to zero. The default value for this policy is **`yes`**. When this policy is on, the server and zone maintenance counter variables are incremented twice - once in ISC code and once in **`qddns`** code.

Function	MIB Variable(s)	Description
Finds the authoritative name server for that zone and sends a query in UDP	*dnsServConfigQddnsRetryTcpOnTruncate*	When a query is sent to a DNS server, it finds the authoritative name server for that zone and sends a query in UDP. If the reply from the recipient authoritative name server is more than 512 bytes, it sends a maximum of 512 bytes of data and also sets **truncation bit** in the message to the DNS server. When the DNS server receives the reply message from authoritive server, the DNS server sees that the truncation bit is ON. According to DNS protocol, the DNS server initiates a TCP connection with the authoritive server. If TCP is blocked in the firewall, TCP handshake cannot happen and many SYN_SENT half open connections can be seen by running **netstat**. If too many requests arrive too fast, **named** runs out of file descriptors and are not able to serve any requests. If the policy **retry-tcp-on-truncate** can be set to **no** in **qddns** block of the DNS server, it cannot connect to the authoritive server using TCP. The DNS server replies with the partially received message.

Function	MIB Variable(s)	Description
Turn off EDNS globally when the name server sends query to a remote name server as a client.	*dnsServConfigQddnsClientEdns*	A policy **client-ends no** can be specified in the **qddn**s block to turn off EDNS globally when the name server sends query to a remote name server as a client. If **client-edns** is set **no**, the server configuration with the directive is ignored. The default value for **client-edns** is **yes**. The server acts like an ISC server. The default value for this policy is **no**.
Turn off **named** respond while syncing the journal files to disk.	*dnsServConfigQddnsSyncJournalToDisk*	If named receives many dynamic updates in master zones or has many slave zones which are obtained via IXFR, **named** does not respond while syncing the journal files to disk. This is the default behavior of ISC code. This default behavior can be changed with a new policy **sync-journal-to-disk no;** in the **qddns** block, which stops syncing the journal file to disk. Instead, it flushes the buffer.
Send dynamic updates to slave zones	*dnsServConfigQddnsAllowSecondaryUpdate*	If the policy **allow-secondary-update** is set to **yes**, dynamic updates can be sent to slave zones. The default value is **no**.
Send notify messages to slaves zones	*dnsServConfigQddnsNotifyAfterLoad*	After start up, **named** sends notify messages to slaves zones. To avoid notify message overload, the default behavior is to not send notify messages at start up. The policy **notify-after-load** can be set to **yes** if the default behavior is not desired.

Function	MIB Variable(s)	Description
IP address of the name server to use in EDUP message	*dnsServConfigQddnsEdupMyIP*	IP address of the name server to use in EDUP message.
IP address of Message Service where the EDUP messages are sent	*dnsServConfigQddnsEdupMessageServiceIP*	IP address of the Message Service where the EDUP messages are sent.
Message service port where the EDUP messages are sent.	*dnsServConfigQddnsEdupMessageServicePort*	Message service port where the EDUP messages are sent.
Organization ID in the EDUP message	*dnsServConfigQddnsEdupOrgId*	Organization ID added to the EDUP message.
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.
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. This trap is also sent if the qddns policy **remove-cname-on-cname-and-other-data-error** is set to **Yes**. In that case, the trap message will be CNAME and other data error, removing CNAME: (xyz).

Function	MIB Variable(s)	Description
The DNS has dumped its database to disk	*dnsServerDumped*	Sent by the DNS server when the DNS database files have been dumped to disk.

Trap object IDs (OIDs)

Purpose

This section describes the components that comprise an SNMP trap. The Alcatel-Lucent enterprise-specific trap contents are defined by the following:

- DHCP: **`IpspgDhcpTrapEntry`**
- DNS: **`IpspgDnsTrapEntry`**

Similar trap variables are defined for both DNS and DHCP, as shown in the following table.

Table 1-3 Trap variables

IpspgDhcpTrapEntry in dhcp.mib	IpspgDnsTrapEntry in named.mib	Type	Description
ipspgDhcpTrIndex	none	Integer	Indicates which trap is received for *dhcp.mib*: `1` = dhcpServerStarted `2` = dhcpServerStopped `3` = dhcpServerReload `4` = dhcpServerSubnetDepleted `5` = dhcpServerBadPacket `6` = dhcpServerFailoverActive `7` = dhcpServerFailoverReturnedControl `8` = dhcpServerSubnetThresholdExceeded `9` = dhcpServerSubnetThresholdDescent `10` = dhcpServerDropUnknownClient `11` = dhcpServerPingResponseReceived For a description of the DHCP traps, refer to the **DHCP server SNMP traps** category at the end of Table 1-1.
none	ipspgDnsTrIndex	Integer	Indicates which trap is received for *named.mib*: `1` = dnsServerStarted `2` = dnsServerStopped `3` = dnsServerReload `4` = dnsServerConfigError `5` = dnsServerDumped For a description of the DNS traps, refer to the **DNS traps** category at the end of Table 1-2.

IpspgDhcpTrapEntry in dhcp.mib	IpspgDnsTrapEntry in named.mib	Type	Description
ipspgDhcpTrSequence	ipspgDnsTrSequence	Counter	Indicates how many times a specific trap is received. This number is a counter that will increment every time a specific trap is received. Such counters are only reset by DHCP and DNS Starts or by a DHCP Restart.
ipspgDhcpTrId	ipspgDnsTrId	Integer	Indicates the application that generated the alarm. Currently, all traps are generated by the Monitor. The value is always 1.

ipspgDhcpTrText	ipspgDnsTrText	80-char string	An ASCII string describing the alarm condition/cause, for example:
			`Lucent DHCP stopped`
			`Lucent DNS started`

ipspgDhcpTrPriority	ipspgDnsTrPriority	Integer	Indicates the priority level as set on the agent for this class and type of trap. Both the DHCP and DNS servers send traps with the following priorities: `1` (inform) for start, stop, reload, failover returned control, subnet threshold descended, and ping response received. `2` (warning) a bad packet is received. Used when there are errors in the config file and when *qddns* finds CNAME and other data error in a zone or a file. `3` (minor) when a subnet threshold is exceeded. `4` (major) when a subnet is depleted. `5` (critical) when a failover is activated.
ipspgDhcpTrClass	ipspgDnsTrClass	Integer	This is not used and is set to the value of **`ipspgDhcpTrIndex`** (DHCP) or **`ipspgDnsTrIndex`** (DNS).
ipspgDhcpTrType	ipspgDnsTrType	Integer	This is not used and is set to the value of **`ipspgDhcpTrIndex`** (DHCP) or **`ipspgDnsTrIndex`** (DNS).

IpspgDhcpTrapEntry in dhcp.mib	IpspgDnsTrapEntry in named.mib	Type	Description
ipspgDhcpTrTime	ipspgDnsTrTime	Counter	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, the local time of the sub-agent host is: `1238788125` (which translates to Fri Apr 3 15:48:45 2009)
ipspgDhcpTrSuspect	ipspgDnsTrSuspect	32-char string	The hostname where the sub-agent (*named* or *dhcpd*) is running.
ipspgDhcpTrDiagId	ipspgDnsTrDiagId	Integer	This is not used and is set to the value of **`ipspgDhcpTrIndex`** (DHCP) or **`ipspgDnsTrIndex`** (DNS).

Sample DHCP traps

The following is sample output from the **`traprcv`** utility. The definitions of the ipspgDhcpTr family of values are in the above table.

```
Received SNMPv2c Trap:
Community: public
From: 127.0.0.1
sysUpTime.0 = 2 days, 03:56:26.29
snmpTrapOID.0 = dhcpServerSubnetThresholdExceeded
ipspgDhcpTrIndex = 8
ipspgDhcpTrSequence = 1
ipspgDhcpTrId = monitor(1)
ipspgDhcpTrText = Lucent DHCP Subnet Threshold exceeded for subnet
   10.51.0.0
ipspgDhcpTrPriority = minor(3)
ipspgDhcpTrClass = 8
ipspgDhcpTrType = 8
ipspgDhcpTrTime = 1184876862
ipspgDhcpTrSuspect = qa2k342
ipspgDhcpTrDiagId = 8


Received SNMPv2c Trap:
Community: public
From: 127.0.0.1
sysUpTime.0 = 2 days, 04:01:15.06
snmpTrapOID.0 = dhcpServerSubnetDepleted ipspgDhcpTrIndex = 4
ipspgDhcpTrSequence = 1
ipspgDhcpTrId = monitor(1)
ipspgDhcpTrText = Lucent DHCP Subnet Depleted for subnet 10.51.0.0
```

```
ipspgDhcpTrPriority = major(4)
ipspgDhcpTrClass = 4
ipspgDhcpTrType = 4
ipspgDhcpTrTime = 1184877151
ipspgDhcpTrSuspect = qa2k342
ipspgDhcpTrDiagId = 4
```

Sample DNS traps

The following is sample output from the **traprcv** utility. The definitions of the ipspgDnsTr family of values are in the above table.

```
Received SNMPv2c Trap:
Community: public
From: 127.0.0.1
sysUpTime.0 = 00:10:09.16
snmpTrapOID.0 = dnsServerStarted
ipspgDnsTrIndex = 1
ipspgDnsTrSequence = 1
ipspgDnsTrId = monitor(1)
ipspgDnsTrText = Lucent DNS started
ipspgDnsTrPriority = inform(1)
ipspgDnsTrClass = 1
ipspgDnsTrType = 1
ipspgDnsTrTime = 1184960025
ipspgDnsTrSuspect = spirit
ipspgDnsTrDiagId = 1


Received SNMPv2c Trap:
Community: public
From: 127.0.0.1
sysUpTime.0 = 00:10:09.61
snmpTrapOID.0 = dnsServerStopped
ipspgDnsTrIndex = 2
ipspgDnsTrSequence = 1
ipspgDnsTrId = monitor(1)
ipspgDnsTrText = Lucent DNS stopped
ipspgDnsTrPriority = inform(1)
ipspgDnsTrClass = 2
ipspgDnsTrType = 2
ipspgDnsTrTime = 1184960025
ipspgDnsTrSuspect = spirit
ipspgDnsTrDiagId = 2
```

Querying ipspgDhcpTrapEntry and ipspgDnsTrapEntry tables and variables

An administrator can query the DHCP and DNS trap tables by using the **gettab** or **getmany** commands. An administrator can also use a **getone** command to query a specific component of a thrown trap. For example,

ipspgDhcpTrText.15

would return the text value of the 15th trap in the table. Likewise,

ipspgDnsTrTime.10

would return the time that the 10th trap in the ipspgDnsTrapEntry table was thrown.

Trap table rolling

The ipspgDhcpTrapTable and ipspgDnsTrapTable only hold up to 31 entries each. When the 32nd DHCP or DNS trap is thrown, the TrapTable clears and the 32nd trap becomes the sole entry in it respective table. This behavior will repeat for each trap that is a multiple of 32.

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.

Installation and configuration order

Purpose

Use these steps as a guide to install and configure the SNMP Module.

Procedure

To successfully install and configure the agents, follow these steps:

1. Obtain the Alcatel-Lucent SNMP Distribution media. Refer to the *SNMP Module Release Notes* for instructions.

2. Complete the pre-installation task. Refer to the *SNMP Module Release Notes* for more information.

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. Refer to Chapter 3, "Start the SNMP Master Agent" for instructions.

END OF STEPS

Install the SNMP Module on Windows

Purpose

Use this procedure to install the SNMP Module on Windows.

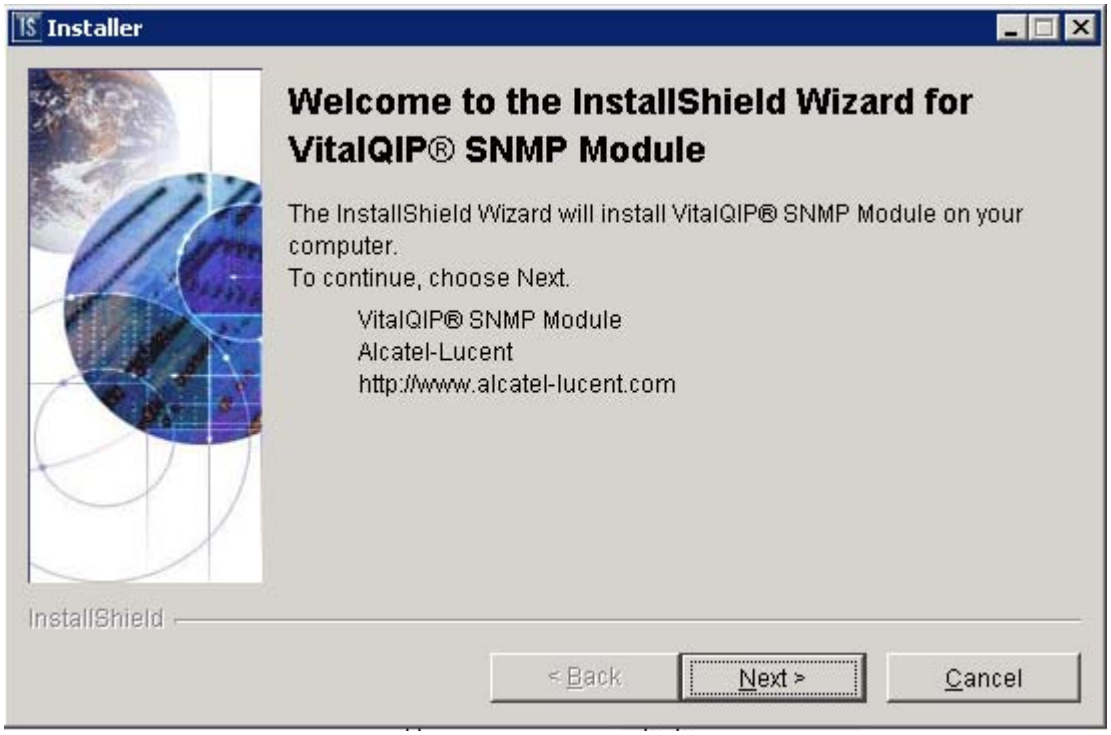
Procedure

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

1 Exit all Windows programs that are currently running.

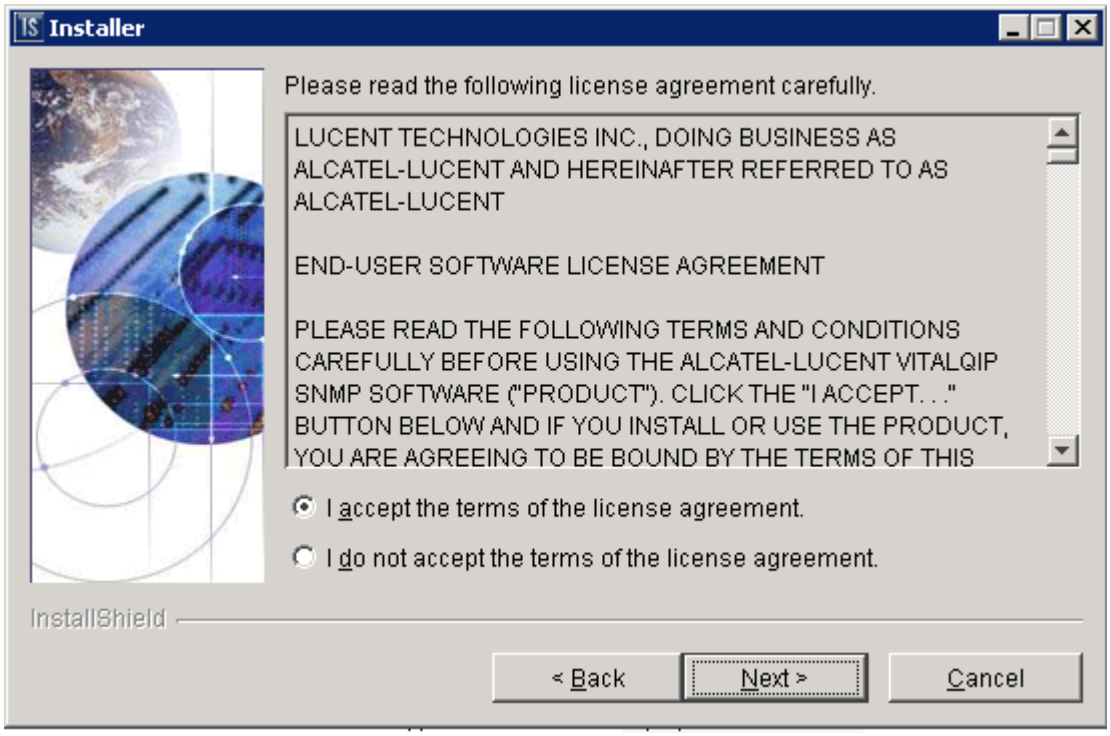
2 Click **Start**. Select the **Run** menu. Enter the drive letter and directory where the installation software has been loaded (for example, *x:\setup.exe*) and click **OK**.

Result: The InstallShield Wizard opens.



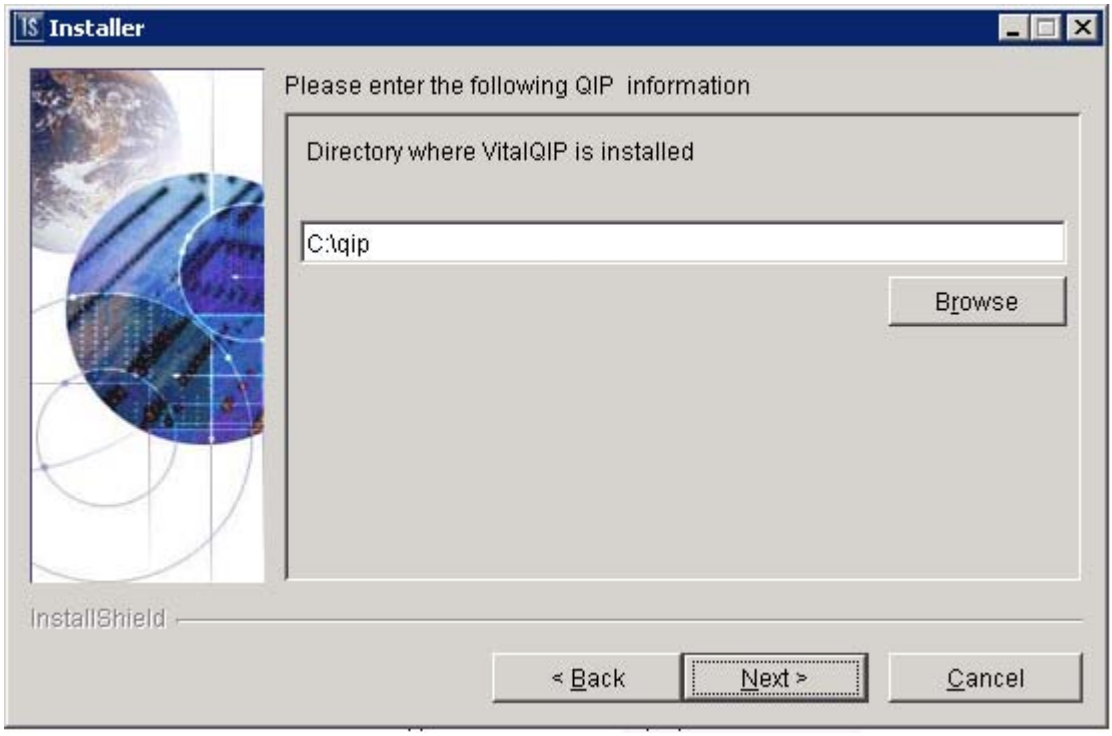
3 Click **Next**.

Result: The License Agreement window opens.



4 After reviewing the license agreement, click **I accept the terms of the license agreement** and click **Next**.

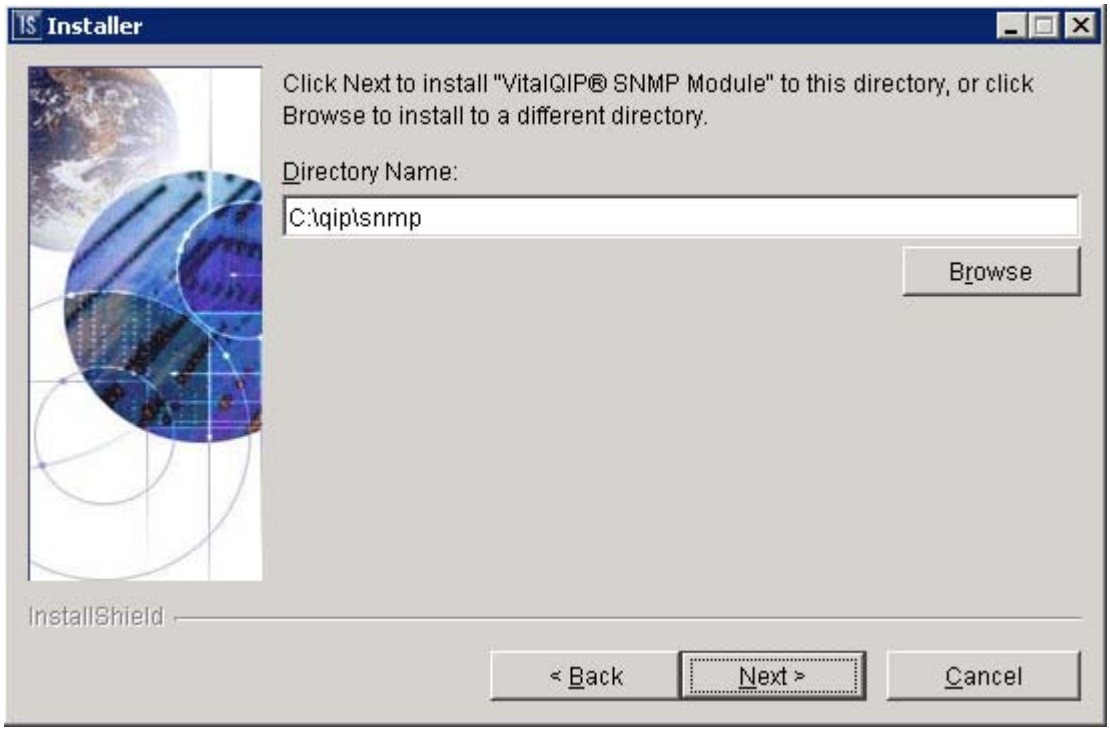
Result: The VitalQIP installation directory window opens.



5 This window displays the directory where the VitalQIP software is installed. Select the default directory or click **Browse** to select a different directory.

6 Click **Next**.

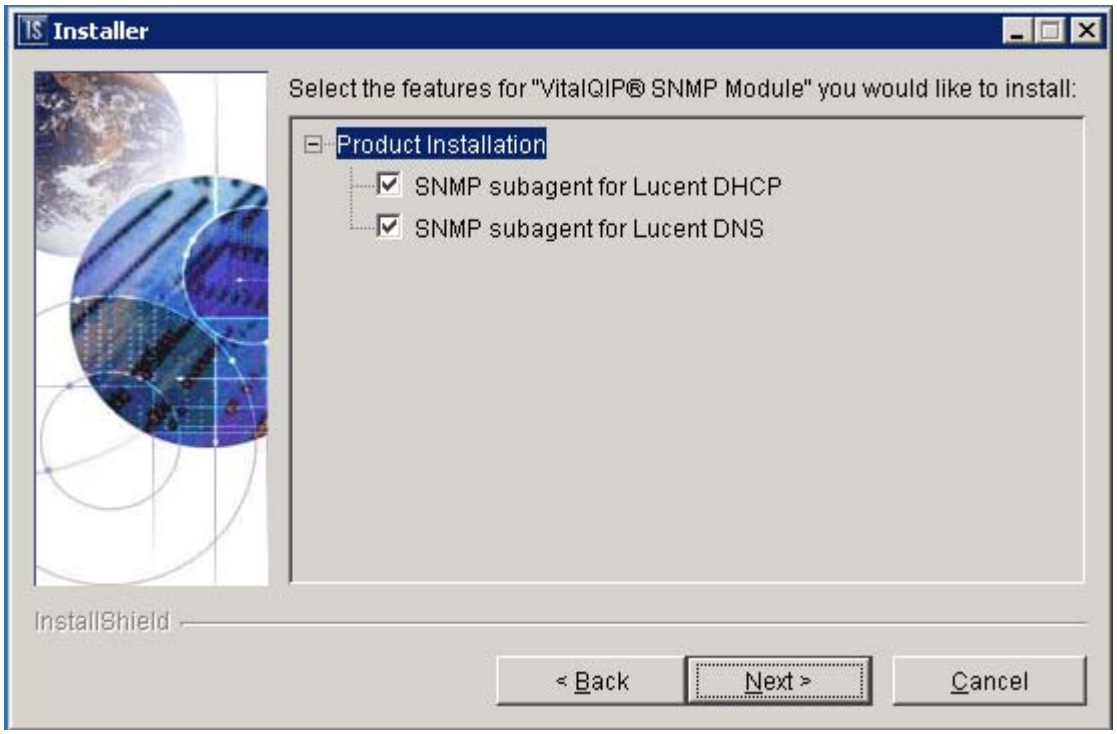
Result: The SNMP Module installation directory window opens.



7 This window displays the directory where the SNMP Module software is to be installed. Select the default directory or click **Browse** to select a different directory.

8 Click **Next**.

Result: The SNMP Module features window opens.

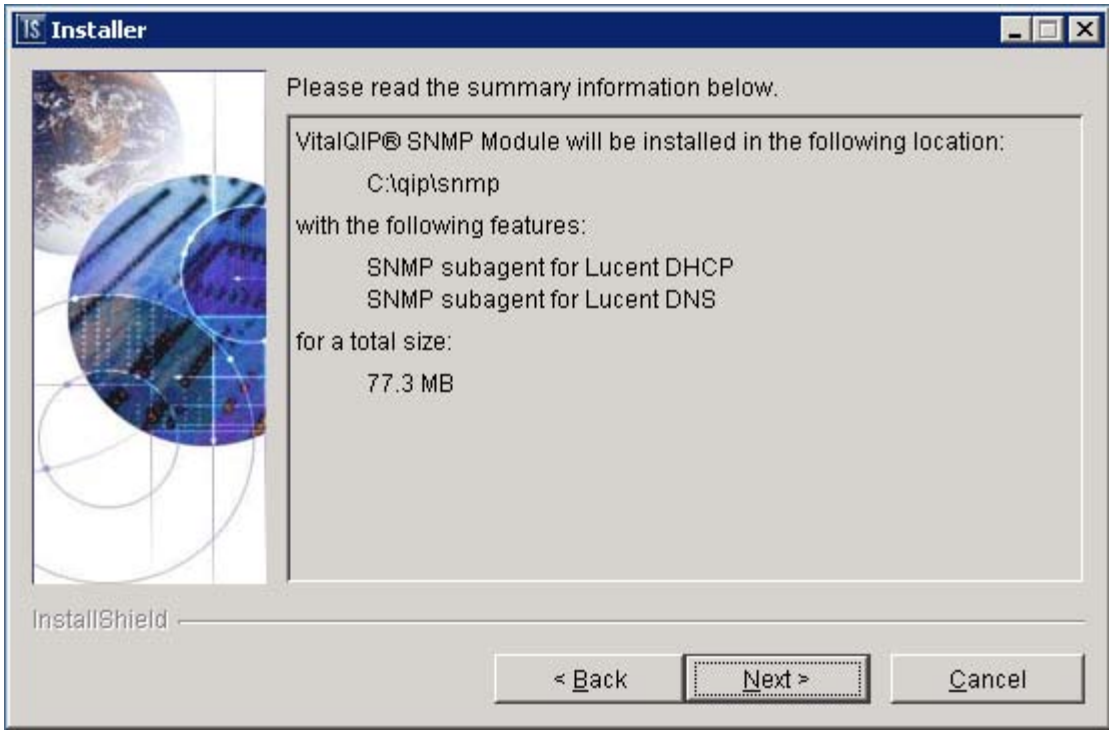


9 This window displays the options that are available for installation on this system. Select or clear the options to install the Lucent DHCP or Lucent DNS SNMP subagent components.

- If no DHCP Server is installed, the message **A Lucent DHCP Service is not installed in this system. Please unselect "DHCP SNMP Agent Server" and then proceed** appears in a dialog box.
- If no DNS Server is installed, the message **A Lucent DNS Service is not installed in this system. Please unselect "DNS SNMP Agent Server" and then proceed** appears in a dialog box.
- Click **Back**. Unselect the features that are not installed and then proceed with the installation.

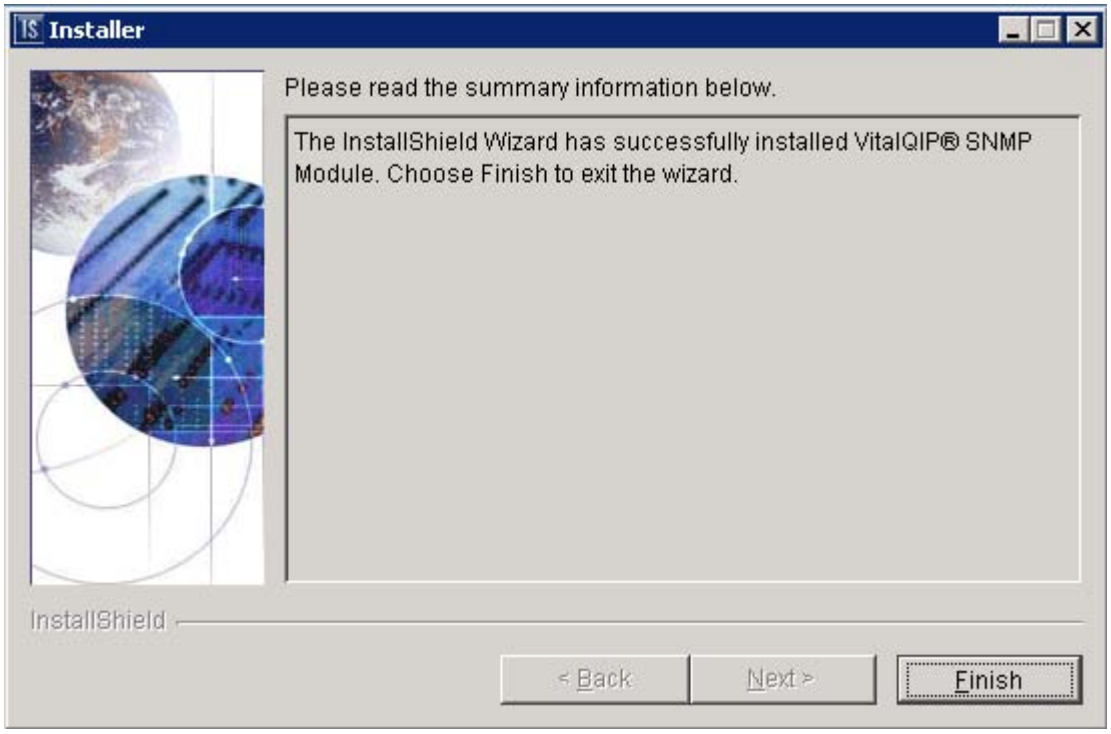
10 Click **Next**.

Result: The Summary window opens.



11 After verifying all settings, click **Next**.

Result: The Summary window opens.



12 Click **Finish**.

END OF STEPS

Uninstall the SNMP Module from Windows

Purpose

This section describes how to uninstall the SNMP Module.

Procedure

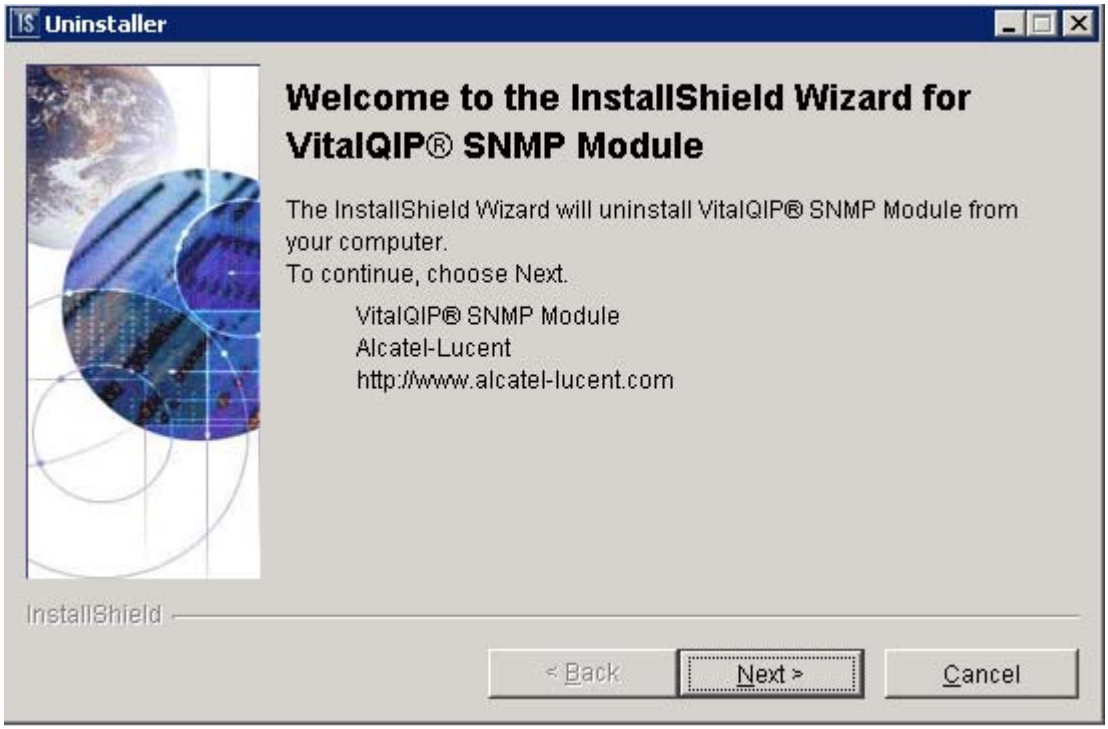
Uninstall the SNMP Module as follows:

1 Under the **Windows Control Panel**, click **Add/Remove Programs**.

Result: The Add/Remove Programs Properties window opens.

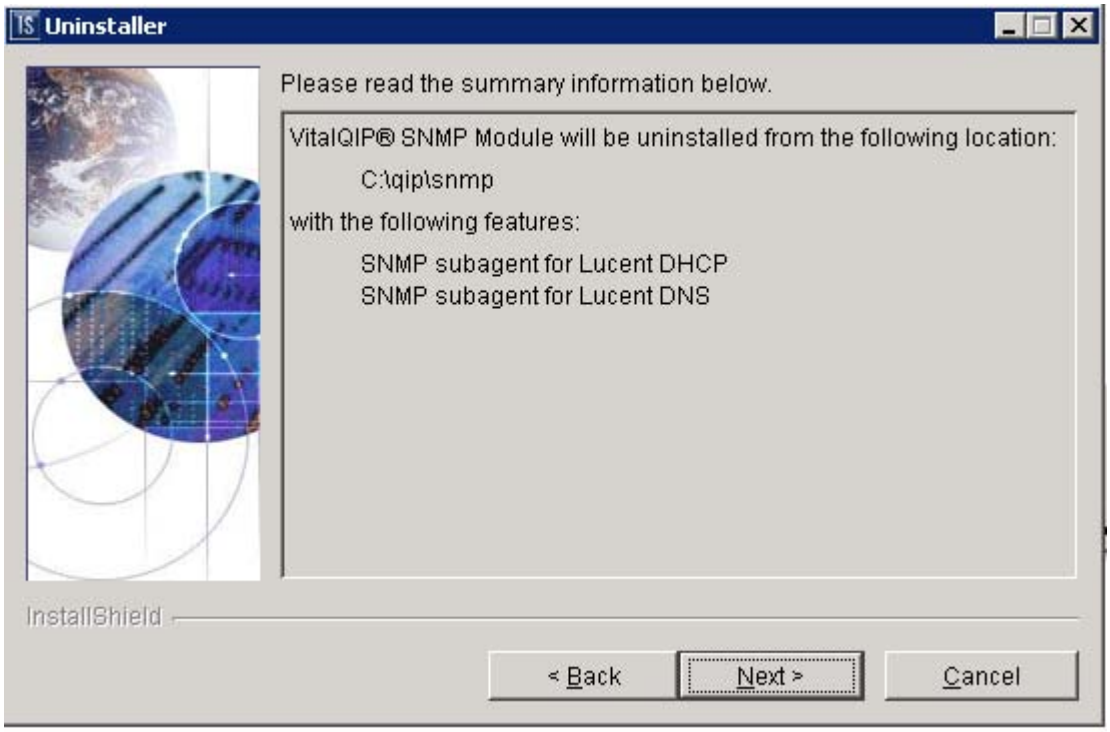
2 Select **VitalQIPVitalQIP®SNMP Module**, and click **Change/Remove**.

Result: The Install shield for uninstalling the SNMP Module window opens.



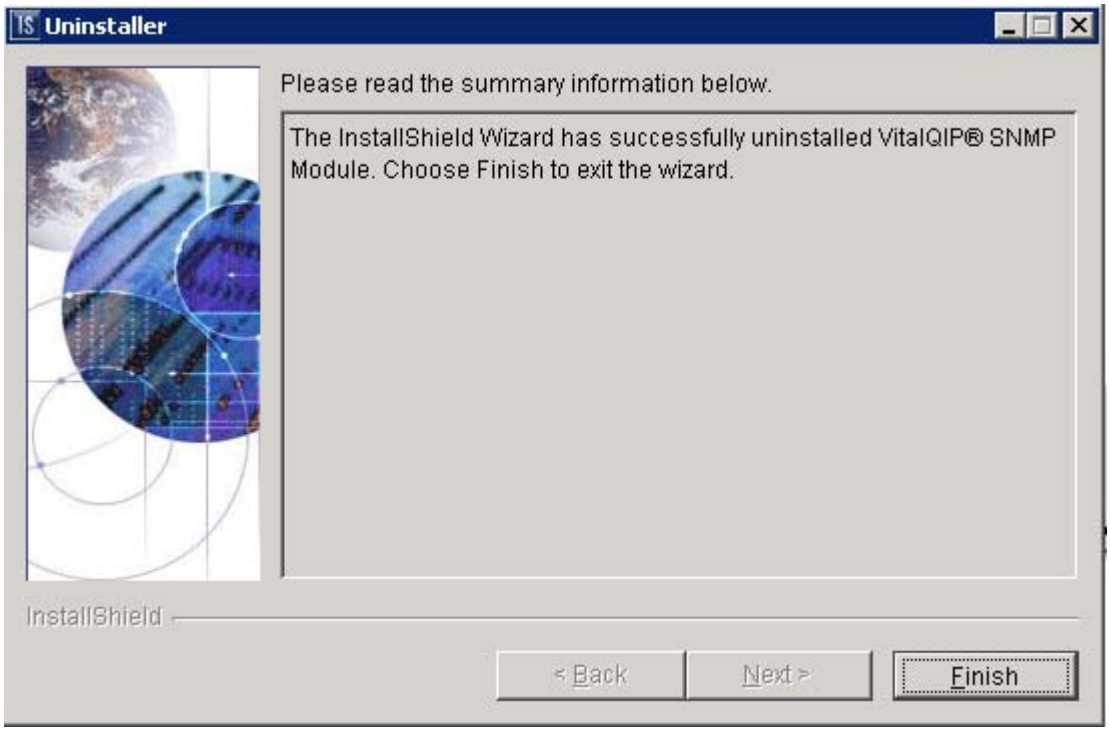
3 Click **Next**.

Result: The Summary window opens.



4 After verifying all settings, click **Next**.

Result: The Summary window opens.



5 Click **Finish**.

END OF STEPS

Install SNMP Module on a UNIX platform

Purpose

Use this procedure to install SNMP on UNIX.

Before you begin

- Before executing the load command, ensure the **`cshrc`** or **`shrc`** command is run to set the correct *$QIPHOME* environment.
- If DHCP is selected, the library *libqsidhcpsnmp.so* is copied to *$QIPHOME/usr/lib.* The earliest version of the Lucent DHCP Server (dhcpd) that must already exist is Version 5.5, Build 9.
- If DNS is selected, the library *libqsidnssnmp.so* is copied to *$QIPHOME/usr/lib.* The earliest version of Lucent DNS Server (*named*) that must already exist is Version 4.2, Build 9. The SNMP-enabling modules are copied to *$QIPHOME/snmp*.
- 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
```

- The log and template files created by this installation are located in *$QIPHOME/log*.
- The installation on UNIX creates the following directory structures:
  - *$QIPHOME/snmp/config* - The directory containing the SNMP Agent configuration files.
  - *$QIPHOME/snmp/bin* - The directory containing the SNMP Master Agent and utilities.

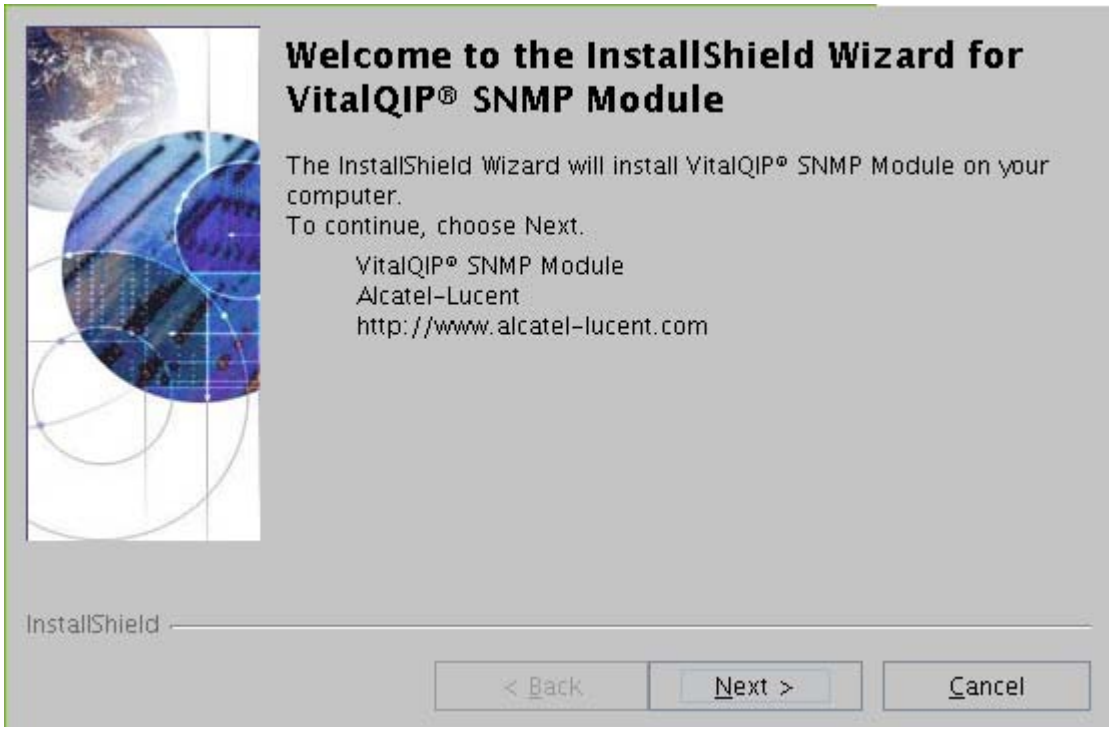
Procedure

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

1 Execute one of the following commands:

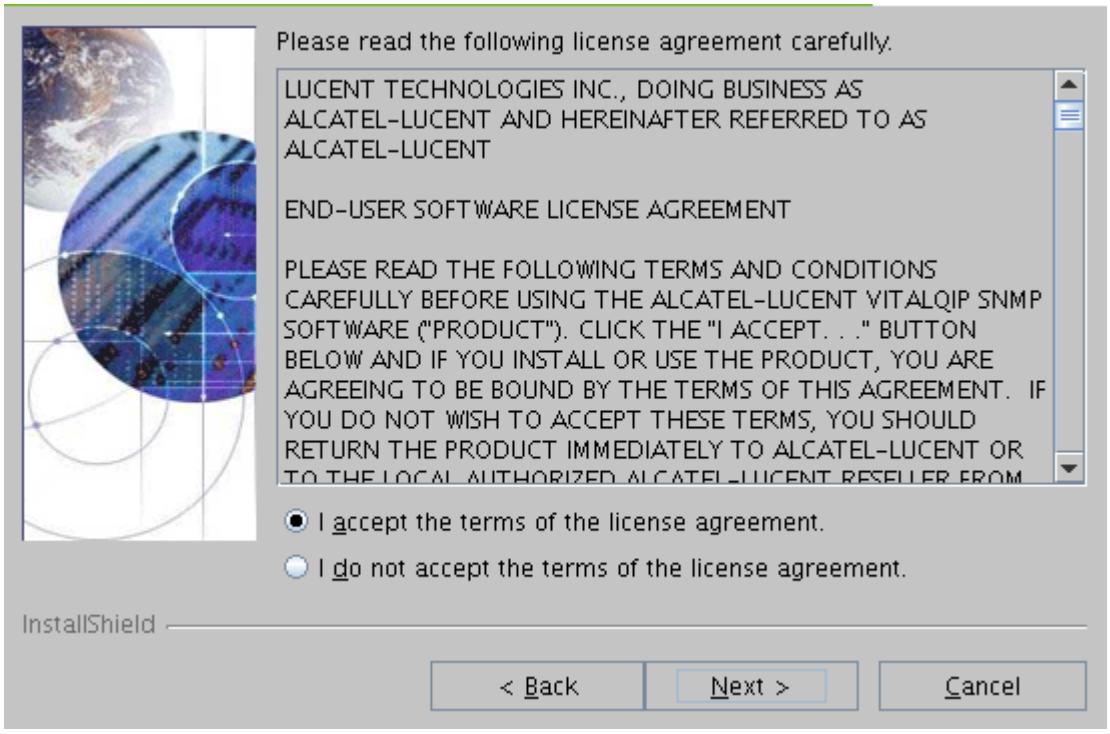
Platform	Executable
Linux	**`./snmp23SetupLinux.bin`**
Solaris	**`./snmp23SetupSolaris.bin`**

Result: The InstallShield Wizard opens.



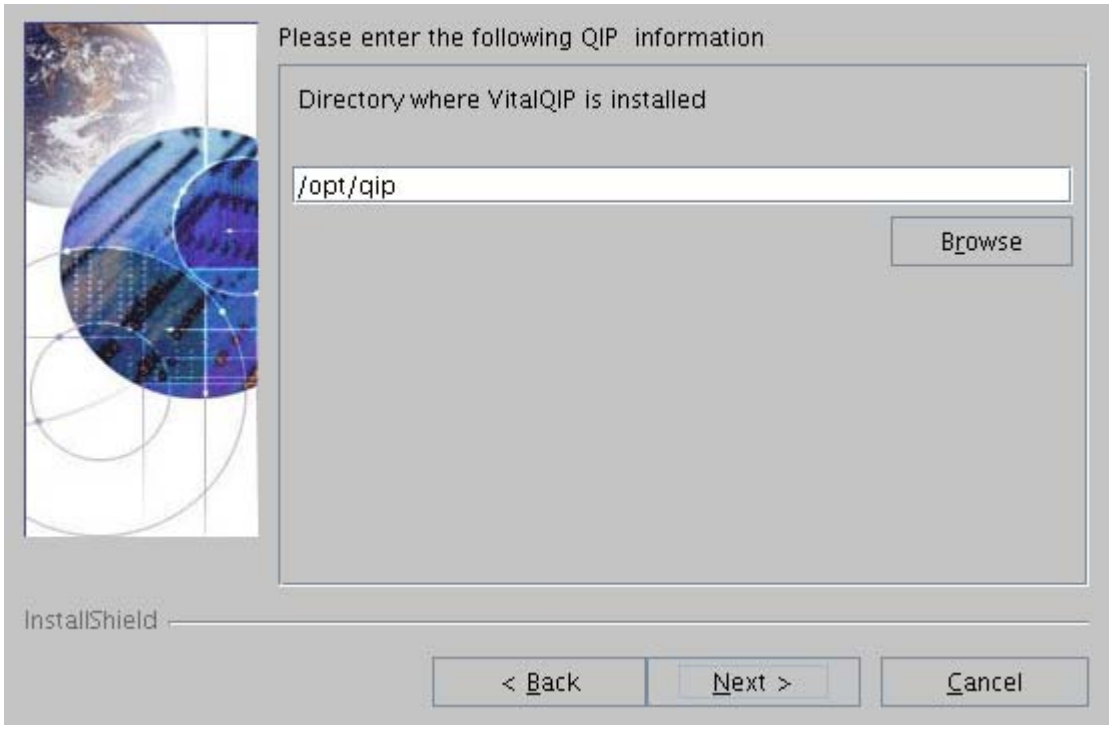
2 Click **Next**.

Result: The License agreement window opens.



3 After reviewing the license agreement, click **I accept the terms of the license agreement** and click **Next**.

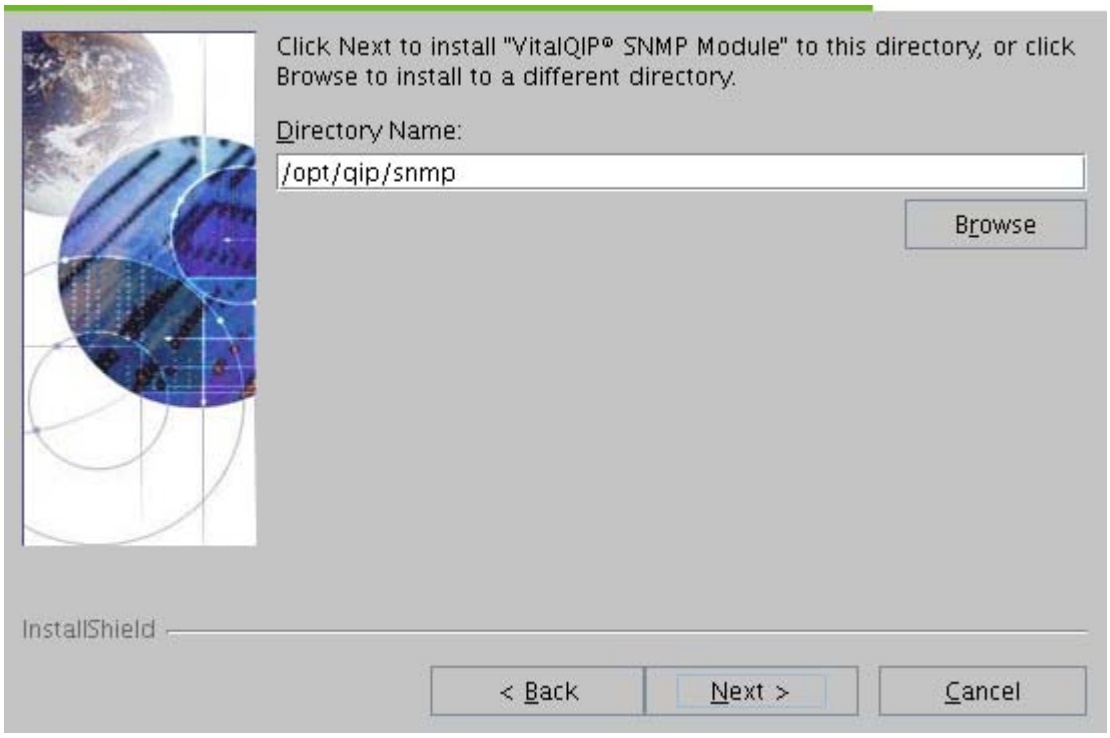
Result: The VitalQIP installation directory window opens.



4 This window displays the directory where the VitalQIP software is installed. Select the default directory or click **Browse** to select a different directory.

5 Click **Next**.

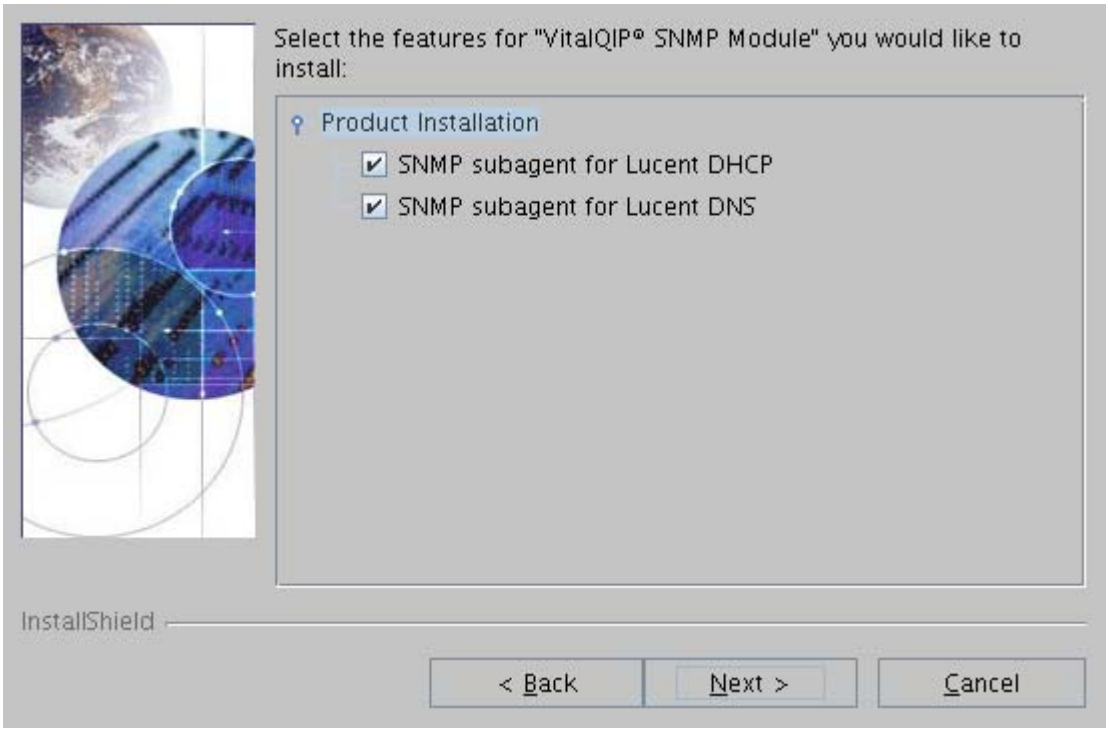
Result: The SNMP Module installation directory window opens.



6 This window displays the directory where the SNMP Module software is to be installed. Select the default directory or click **Browse** to select a different directory.

7 Click **Next**.

Result: The SNMP Module features window opens.

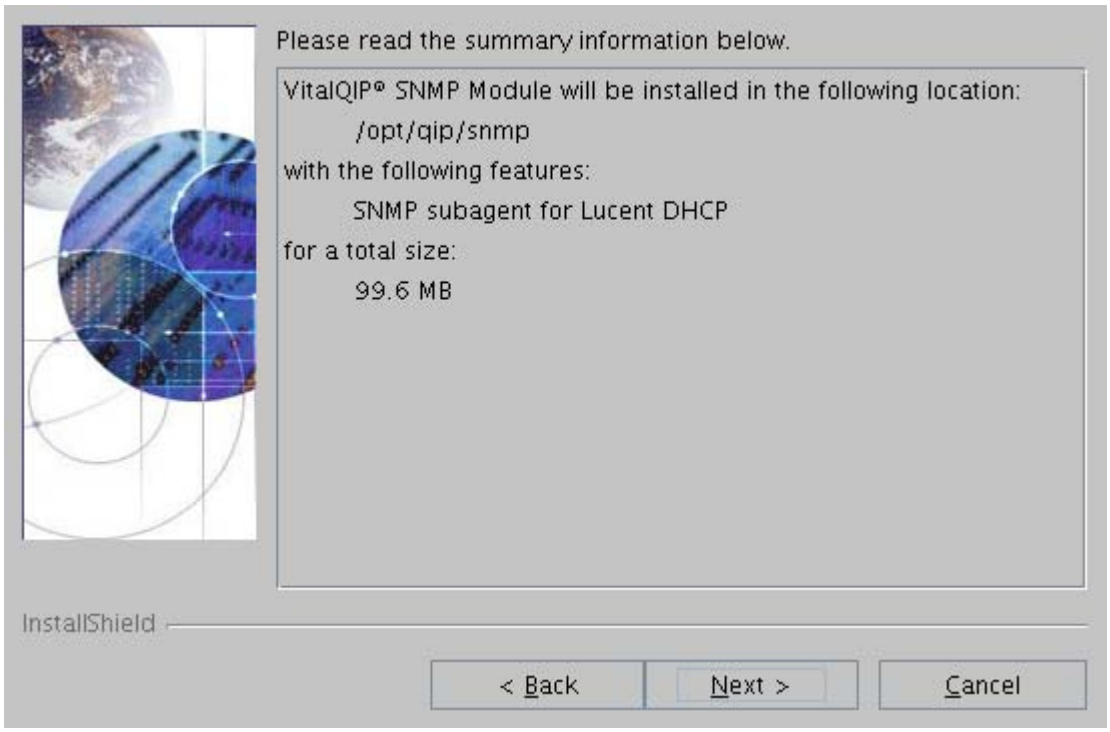


8 This window displays the options that are available for installation on this system. Select or clear the options to install the Lucent DHCP or Lucent DNS SNMP subagent components.

- If no DHCP Server is installed, the message **A Lucent DHCP Service is not installed in this system. Please unselect "DHCP SNMP Agent Server" and then proceed** appears in a dialog box.
- If no DNS Server is installed, the message **A Lucent DNS Service is not installed in this system. Please unselect "DNS SNMP Agent Server" and then proceed** appears in a dialog box.
- Click **Back**. Unselect the features that are not installed and then proceed with the installation.

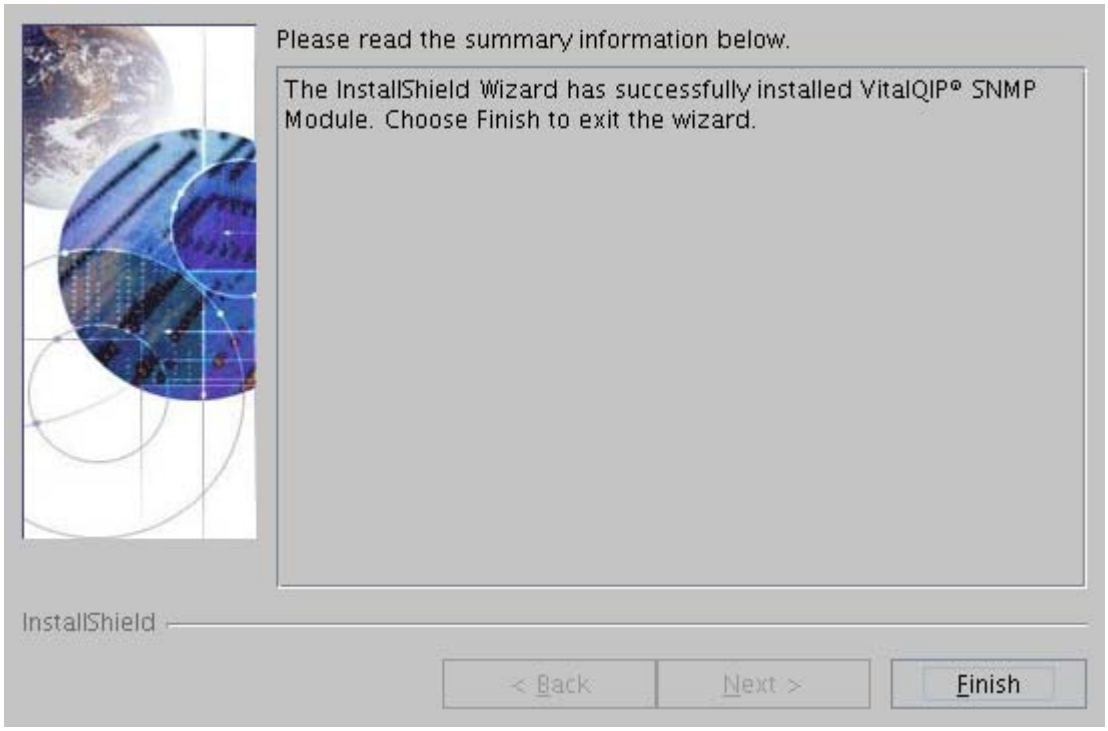
9 Click **Next**.

Result: The Summary window opens.



10 After verifying all settings, click **Next**.

Result: The Summary window opens.



11 Click **Finish**.

END OF STEPS

Configure the SNMP Master Agent

Before using the SNMP Module, you must configure the Master Agent. This involves the following:

Configuring 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 can define additional user names in the SNMP Master Agent configuration file, *snmpd.cnf*. Refer to "Configure additional user names" (p. 22) for instructions.

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.

Multiple Management Station destinations can be defined by adding additional lines in the *snmpd.cnf* file, as detailed in "Configure additional SNMPv1/v2c notification traps" (p. 24) and "Configure additional SNMPv3 notification traps" (p. 26).

Configure additional user names

Purpose

Use this procedure to specify a user name within an SNMPv3 Protocol Data Unit (PDU).

Procedure

Follow these steps:

1 To begin, choose one of the following actions.

If you are on...	Then...
UNIX	Enter the following commands: **`cd $QIPHOME/snmp/config`** **`vi  snmpd.cnf`**
Windows	Open a command window and enter the following command (or open the file in Notepad): **`edit %QIPHOME%\snmp\config\snmpd.cnf`**

2 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.

3 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
```

Refer to Appendix A, "Increase SNMP Module security" for additional information

> **Note:** 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.

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

Note: 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. Start the SNMP Master Agent and then the SNMP-enabled service(s).

END OF STEPS

Configure additional SNMPv1/v2c notification traps

Purpose

Use this procedure to send SNMPv1/v2 traps to a different Management Station destination.

Before you begin

- 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.
- 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.

Procedure

Follow these steps:

1 Open the *snmpd.cnf* file in an text editor.

2 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
```

Note: For SNMPv2, the line will contain `v2cExampleParams` instead.

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

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

Note: 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.

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

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

```
snmpNotifyEntry  31 Console trap nonVolatile
```

7 Copy and paste this line and then change the new **31** entry to the value that you assigned the *snmpTargetAddrEntry* line in Step 3 above.

8 Change the Description tag **Console** in the copied string to the unique value you selected in Step 5.

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

10 Stop the DHCP and/or DNS server.

11 Stop the SNMP Master Agent.

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

END OF STEPS

Configure additional SNMPv3 notification traps

Purpose

Use this procedure to send SNMPv3 traps to a different Management Station destination.

Before you begin

- 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.
- 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.

Procedure

Follow these steps:

1 Open the *snmpd.cnf* file in an text editor.

2 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
```

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

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

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

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

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

```
snmpNotifyEntry  32 TrapSink trap nonVolatile
```

7 Copy and paste this line and then change the second 32 to the value that you assigned the *snmpTargetAddrEntry* line in Step 3 above.

8 Change the second TrapSink to the unique tag value in Step 5.

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

10 Stop the DHCP and/or DNS server.

11 Stop the SNMP Master Agent.

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

END OF STEPS

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

Purpose

Use this procedure to start the SNMP Master Agent on UNIX.

Before you begin

- You must always start the SNMP Master Agent *before* starting either the DHCP or DNS server. If the SNMP Master agent is restarted, the DHCP and 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):.

Procedure

Follow these steps:

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.

END OF STEPS

Start the SNMP Master Agent on Windows

Purpose

Use this procedure to start the SNMP Research Master Agent on Windows.

Before you begin

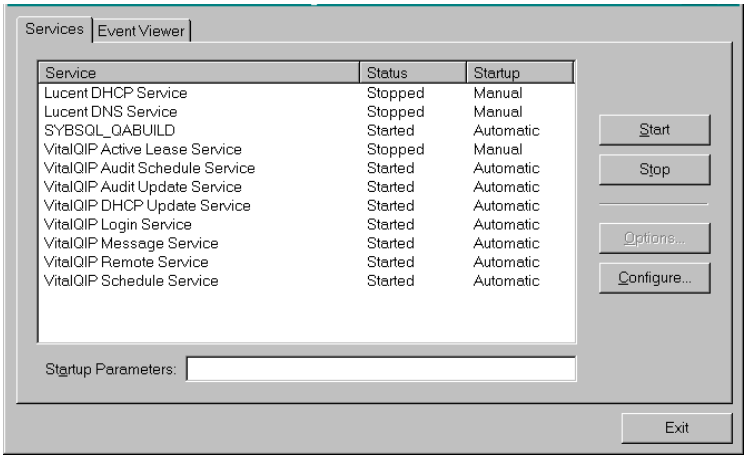
- This process assumes that the installation was successful and the Registry entries were created.
- The SNMP Master Agent must always be started before starting either the DHCP or DNS server. If the master agent is restarted, then the DHCP and/or DNS server must also be restarted. You may want to create a service dependency; see your system administrator or Windows documentation for the appropriate steps.

Procedure

Follow these steps:

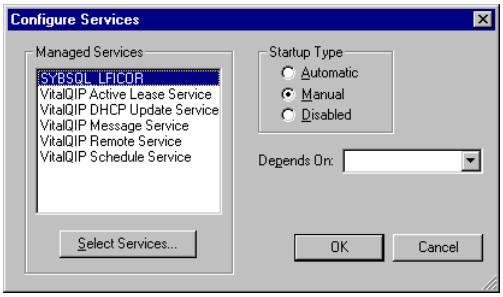
1 Start the VitalQIP Service Controller application. (This can be found in the **Start|Programs|Alcatel-Lucent VitalQIP** program group.)

Result: The Service Controller window opens.



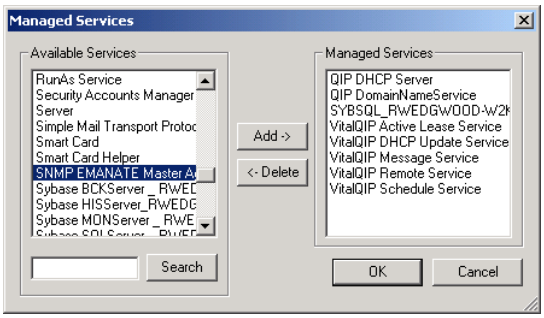
2 Click **Configure...**.

Result: The Configure Services window opens.



3 Click **Select Services**.

Result: The Managed Services window opens.



4 Click **Search**.

Result: This populates the **Available Services** list box.

5 **From the Available Services** list box, highlight the **SNMP EMANATE Master Agent** and click **Add**.

Result: SNMP EMANATE Master Agent appears in the **Managed Services** list box.

6 Click **OK** when you are done.

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

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

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

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

END OF STEPS

Verification of DHCP/DNS server SNMP MIB access

Purpose

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 verification procedure. Refer to "Verification on UNIX" (p. 6) for Windows.

Before you begin

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 UNIX

To verify DHCP/DNS SNMP MIB access on UNIX, 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. 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...	Then..

DNS	Run the test script file:
	$QIPHOME/snmp/bin/TestDNS
	Note: This test requires that *localhost* represent the current Hostname.
	Result: The Output indicates if the test was successful:
	• If verification passes, the DNS service description and version are returned. The following output displays for a successful test:
	`SUCCESS:`
	`Dns Server Subagent Test`
	`dnsServSystemDescr.0 = Lucent QDDNS <VERSION_INFO>`
	• If verification fails, the DNS service description and version are *not* returned. The following output displays for an unsuccessful test:
	`FAILURE:`
	`Error code set in packet - No such variable name. Index: 1.`

DHCP	Run the test script file:
	$QIPHOME/snmp/bin/TestDHCP
	Note: This test requires that *localhost* represents the current Hostname
	Result: The Output indicates if the test was successful:
	• If verification passes, the DHCP service description and version are returned. The following output displays for a successful test:
	`SUCCESS:`
	`DHCP Server Subagent Test`
	`dhcpServSystemDescr.0 = (Version: <VERSION_INFO> - Lucent DHCP Server)`
	• If verification fails, the DHCP service description and version are *not* returned. The following output displays for an unsuccessful test:
	`FAILURE:`
	`Error code set in packet - No such variable name. Index: 1.`

END OF STEPS

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*.

5 From a MS-DOS command window, perform the following actions.

For...	Then..

DNS	Run the test batch file from the home directory:
	%QIPHOME%\snmp\bin\TestDNS
	Note: This test requires that *localhost* represent the current Hostname.
	Result: The Output indicates if the test was successful:
	• If verification passes, the DNS service description and version are returned. The following output displays for a successful test:
	`SUCCESS:`
	`dnsServSystemDescr.0 = Lucent QDDNS <VERSION_INFO>`
	• If verification fails, the DNS service description and version are not returned. The following output displays for an unsuccessful test:
	`FAILURE:`
	`Error code set in packet - No such variable name. Index: 1.`

DHCP	Run the test batch file from the home directory:
	%QIPHOME%\snmp\bin\TestDHCP
	Note: This test requires that *localhost* represent the current Hostname.
	Result: Output is returned that indicates if verification passes:
	• If verification passes, the DHCP service description and version are returned. The following output displays for a successful test:
	`SUCCESS:`
	`dhcpServSystemDescr.0 = (Version: <VERSION_INFO> - Lucent DHCP Server)`
	• If verification fails, the DHCP service description and version are not returned. The following output displays for an unsuccessful test:
	`FAILURE:`
	`Error code set in packet - No such variable name. Index: 1.`

END OF STEPS

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, 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.

4 Console and template installations

Overview

Purpose

This chapter covers the SNMP console and template installation.

Contents

This chapter covers these topics.

Console installation

Overview

How a console installation differs from the standard installation

The console installation is a text-based installation. Since the console installation is not GUI-based, it is much faster. All the information that is collected in the regular SNMP installation appears as prompts in the console installation. The console installation is not dependent upon platforms.

Install SNMP with console installation

Purpose

You can also install SNMP by using InstallShield's console option. The console opens a separate screen, and displays the text mode of the install package, and thus is much faster.

The console installation can be used when you are unable to use the GUI installation. For instance, the console installation can be used when an X display cannot be exported or when you are installing over a slow WAN.

Start the installation

Follow these steps:

1 The table below shows the commands to start a console installation.

Table 4-1 Start the installation

If you are installing on...	Then...
Windows	1. Exit all Windows programs that you are currently running. 2. From your desktop, click **Start** and select **Run**. The Run screen opens. 3. Type: `cmd` 4. From the command prompt, change to the directory where the SNMP installation file is located: **`cd <drive>:\<path to directory>`** 5. Type: **`x:\snmp23SetupWin32.exe -console`** Where x: is the location of the copied installation binary.
Linux or Solaris	1. If SNMP daemons are running from an already installed version, stop all SNMP processes. 2. Change to the directory where the SNMP installation file is located: **`cd <path to directory>`** 3. Execute: – For Linux: **`./snmp23SetupLinux.bin -console`** – For Solaris: **`./snmp23SetupSolaris.bin -console`**

2 Follow the same steps and answer at the same prompts as you do for the standard installation of SNMP.

END OF STEPS

Template installation

Overview

How template installation differs from the standard installation

The template installation runs the standard installation and creates a template. The template can then be transported from one machine to another and used to install SNMP on multiple machines.

Template installation steps

The template installation is performed in two steps:

1. The template is recorded and created during an installation.
2. The template is used to install on additional servers.

Install SNMP using a template

Purpose

This section provides instructions on how to use a template installation.

Record and create the template

The table below shows the commands used to create a template file for an installation.

Note: If you do not give a full path to the template file, the template file will be created in the current directory.

Note: There is no space between **`-options`** and **`-record`**. Once an installation has been successfully completed using the install package, the template file will be created upon the reboot of the machine for Windows. No reboot is required for UNIX.

Table 4-2 Start the installation

If you are installing on...	Then...
Windows	1. Exit all Windows programs that you are currently running. 2. From your desktop, click **Start** and select **Run**. The Run screen opens. 3. Type: **`cmd`** 4. From the command prompt, change to the directory where the SNMP installation file is located: **`cd <drive>:\<path to directory>`** 5. Type: **`x:\snmp23SetupWin32.exe -options-record <template_file>`** Where x: is the location of the copied installation binary. **Note:** If you do not specify the path for *<template_file>*, the template file is created in the local directory.

If you are installing on…	Then…
Linux or Solaris	1. If SNMP daemons are running from an already installed version, stop them . 2. Change to the directory where the SNMP installation file is located: **cd *<path to directory>*** 3. Execute: – For Linux: **./snmp23SetupLinux.bin -options-record *<template_file>*** – For Solaris: **./snmp23SetupSolaris.bin -options-record *<template_file>***

Use the template to install SNMP

If you want to use this template to install SNMP on another machine, you will need to modify some data. Depending upon the type of installation, change the appropriate information for the computer using the template. Then type:

- Windows:

snmp23SetupWin32.exe -options *<template_file>*

- Linux:

./snmp23SetupLinux.bin -options *<template_file>*

- Solaris:

./snmp23SetupSolaris.bin -options *<template_file>*

This command will start the installer with the data already populated, according to the template file being used.

Silent mode

If you need to use the silent mode (no user input is required for the installer's prompts; installation is run based on the template entries), type:

- Windows:

snmp23SetupWin32.exe -options *<template_file>* -silent

- Linux:

./snmp23SetupLinux.bin -options *<template_file>* -silent

- Solaris:

./snmp23SetupSolaris.bin -options *<template_file>* -silent

Note: On Windows, you must reboot the machine after running an installation in Silent Mode. The installation program will not reboot the machine automatically.

A Increase SNMP Module security

Overview

Purpose

This appendix covers adding additional security to the SNMP Module. Security can be enhanced 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

Purpose

MIB variables can be queried by anyone who knows the default community. 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
```

Alcatel-Lucent does not recommend the "public" community to be deleted. Additional communities can be defined as described.

Procedure

To change the community name string, follow these steps:

1. Using 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 t0000000 public public localSnmpID - -
  nonVolatile
snmpCommunityEntry t0000001 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.

END OF STEPS

SNMPv3 security

Purpose

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.

Basic SNMPv3 security can be added by setting up user name and password protection. 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. 8). Otherwise, follow the steps in the section to establish basic user name and password protection.

Before you begin

Before setting the user name and password protection, obtain the following information:

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

Note: The engine ID is different for each host. The engine ID can be obtained by querying the server with the following command:

`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, Set Access privileges, and receive traps.

Set user privileges

To set user privileges, follow these steps:

1 Using 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
```

Note: 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.

END OF STEPS

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

END OF STEPS

Set authentication and privacy protocols for SNMPv3

Purpose

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.

Before you begin

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
  - Another view defined in **vacmViewTreeFamilyEntry** section
- Monitoring station location
- Monitoring station ID tag
- Monitoring station IP address
- SNMPv3 user parameter tag
- Engine ID

Note: 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 Using 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
```

Note: 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.

Result: 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 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.

Note: While the **getone** utility offers 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 is an example 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"
```

END OF STEPS

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 workstation. 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
```

Note: 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.

5 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:

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

b. 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
```

c. Save the *mgr.cnf* file.

END OF STEPS

Index