Release Notes

for

OmniVista 2500 NMS
Version 4.1.2.R02

May 2015

Revision A

Part Number 033030-10

READ THIS DOCUMENT

Includes OmniVista 2500 NMS for
Windows Server 2012 R2
Windows 7 Professional SP1
Windows 8.1
Red Hat Linux 6.5, 7.0
SUSE Linux 11, 12

Alcatel-Lucent Enterprise
26801 West Agoura Road
Calabasas, CA 91301
(818) 880-3500
(818) 880-3505 Fax

Table of Contents

Revision History

Release	Revision	Date	Description of Changes
4.1.2.R02	A	05/22/15	GA Release
4.1.2.R01	B	12/19/14	Maintenance Release
4.1.2.R01	A	10/24/14	GA Release
4.1.1	B	12/19/14	Maintenance Release
4.1.1	A	09/10/14	GA Release
3.5.7	B	04/21/14	Maintenance Release
3.5.7	A	01/27/14	GA Release

1.0 Introduction

These Release Notes cover the basic feature set supported by Alcatel-Lucent Enterprise OmniVista 2500 NMS 4.1.2.R02 for the following certified platforms (supported on 64-bit platforms only):

- Windows
  - Windows Server 2012, Release 2
  - Windows 7 Professional, Service Pack 1
  - Windows 8.1
- Linux
  - Redhat Linux ES Versions 6.5, 7.0
  - SUSE Professional v11 (Service Pack 3), v12
- Virtual Appliance
  - CentOS 6.4

  Note: The OmniVista 2500 NMS Server should only be installed on **Windows 7 or 8.1** platforms when managing **500 or fewer devices**. For managing more than 500 devices, install the OmniVista Server on one of the other certified Windows or Linux platforms shown above.

Known problems, limitations, and workarounds are included. Please read the applicable sections in their entirety as they contain important operational information that may impact successful use of the application.

1.1 Technical Support Contacts

For technical support, contact your sales representative or refer to one of the support resources below. Alcatel-Lucent Enterprise Service and Support can be reached as follows:

- North America, Latin America, Other International
  - Phone
    - North America: 1-800-995-2696
    - Latin America: 1-877-919-9526
    - Other International: 1-818-878-4507
  - World Wide Web: https://service.esd.alcatel-lucent.com
  - E-Mail for Non-Critical Technical Questions: esd.support@.alcatel-lucent.com
- Europe
  - Phone: +800 00 200 100
  - World Wide Web: https://businessportal.alcatel-lucent.com
  - E-Mail for Non-Critical Technical Questions: ebg_global_supportcenter@alcatel-lucent.com
- Asia Pacific
  - Phone: +800 00 200 100
  - World Wide Web: www.businesspartner.alcatel-lucent.com
  - E-Mail for Non-Critical Technical Questions: ebg_global_supportcenter@alcatel-lucent.com

1.2 Documentation

The user documentation is contained in the on-line Help installed with this product.

1.3 New in 4.1.2.R02

Applications

Analytics

A new Analytics Application is now available. The application enables the user to create and display reports that provide a comprehensive view of network resource utilization, including information on users, devices, and applications. Reports can also provide information on usage trends, including predictive analysis of future network resource utilization.

There are basically two report types: "Visibility" Reports (Top N Applications, Top N Clients, Top N Switches, and Top N Ports Utilization) can be configured to show network utilization over different time periods (e.g., daily, hourly, monthly), and show trends in network utilization over those time periods. Note that “Visibility” Reports are supported on the following AOS devices only: OS6250/6450 devices (6.6.1.R01 and later), OS6850/6855 devices (6.4.1.R01 and later, OS6860 (8.1.1.R01 and later), OS6900 (7.3.2.R01 and later), OS10K (7.3.1.R01 and later).

"Availability" Reports (Network Availability, Alarms) provide a "real-time" view of all discovered network switches. You can view the reports in different formats and customize how the data is displayed. The reports are described below:

- **Top N Applications -** Displays information about the top applications being accessed on the network, including which users are using an application, and which switches/clients have the most traffic for an application.
- **Top N Clients -** Displays information for the Top Network clients including the number of traffic flows for each user.
- **Top N Switches -** Displays information for the top devices on the network in terms of the device's resource usage. Devices are ranked based on the device's CPU usage, memory usage, and temperature.
- **Top N Ports Utilization -** Displays network ports by utilization over time. This report can also provide predictive analytics to show expected future usage. There is also a separate Anomalies Screen that displays any anomalies that are discovered in established port utilization trends.
- **Network Availability -** Displays the current operational state of all discovered network devices (Up/Warning/Down).
- **Alarms -** Displays network alarms by severity level for all discovered network devices.

You can also print a report or download a report in PDF or PNG format, and even include the data as part of a scheduled report that is automatically generated in the Report Application.

Report

A new Report Application is now available. The application enables you to create and schedule Analytics Reports that can be viewed and stored as PDF documents. This way, in addition to real-time viewing of Analytics Reports in the Analytics Application, you can automatically generate and store Analytics Reports that you can view at any time. You can configure a report to include information from specific Analytics Reports (e.g., Top N Users, Top N Apps) as well as specific views of that report (e.g., Summary View, Detailed View). You can then configure the

report to be generated at specific times/intervals (e.g., Daily, Weekly). When the report is generated, it takes a current snapshot of the Analytics information you specified. These generated reports are then displayed on the Report List Screen, where they can be downloaded and viewed as PDFs.

Unified Access

The Unified Access application now supports additional functionality for OmniAccess WLAN devices. The following configuration screens were added under Unified Profiles. They enable the user to configure authentication parameters for wireless devices and include those parameters in Access Authentication Profiles.

- **Global Configuration -** Used to configure global Unified Profile settings and create global AAA Profiles.
- **Access Policies -** Used to create Location and Time Period Access Policies. These policies enable you to configure location and time parameters that can be included in an Access Role Profile to limit the profile to specific devices and time periods.
- **Port Groups -** Used to create Edge Port Groups. If a Port Group is assigned to a port(s) that is not assigned to an Access Authentication Profile, that port(s) is automatically configured as an Edge Port.
- **Diagnostics -** Used to display Access Guardian Diagnostics information for an end station.
- **Wireless Profiles -** Used to configure authentication parameters for wireless Access Points (APs). These parameters are then included as part of an Access Authentication Profile that can be applied to AP Groups on wireless controllers. There is a new "Wireless Settings" section on the Access Authentication Profile Screen. This section enables you to configure a "Virtual" AP Profile (containing wireless 802.1X/MAC Authentication/SSID parameters). You then assign the Virtual AP Profile (as part of an Access Authentication Profile) to AP Groups. The following configuration screens are used to configure authentication parameters for Virtual AP Profiles as well as AP Groups:
  - **802.1X Authentication Profile -** Used to configure 802.1X Authentication Profiles for wireless devices.
  - **MAC Authentication Profile -** Used to configure MAC Authentication Profiles for wireless devices.
  - **SSID Profile -** An SSID Profile can be created and included in an Access Authentication Profile that can be assigned to wireless devices on the network
  - **AP Group -** Used to configure an AP Group. An AP Group is a logical set of physical APs. To apply an Access Authentication Profile (and Virtual AP Profile) to an AP, you apply the profile to a specific AP Group on a wireless controller.

VXLANs

A new VXLAN application is now available. A VXLAN uses a L2 over L3 encapsulation technique to overlay L2 network segments on L3 network infrastructure. Only Virtual Machines (VMs) residing in the servers attached to the same VXLAN can communicate with each other.

The Alcatel-Lucent Enterprise implementation of VXLAN enables you to create a VXLAN Service. A VLAN Service defines a Virtual Forwarding Instance (VFI) that is capable of learning device MAC addresses from the access side and from the network side and then switching the traffic based on this information. Each VXLAN Service is basically an VFI that is capable of

learning customer MAC addresses from the access side (Service Access Points - SAP) and from the network side (mesh Service Distribution Point - SDP) and then switching traffic based on this information. Creation of multicast VXLAN Services requires PIM configuration on devices. A new IP Multicast (PIM) Application is also now available in OmniVista.

VM Snooping

A new Virtual Machine (VM) Snooping Application is now available under the VXLANs Application. VM Snooping on the switch detects and identifies Virtual Extensible LAN (VXLAN) traffic by inspecting packets to determine if they are VXLAN encapsulated packets. Once VXLAN traffic is identified, the VM Snooping feature on OmniVista collects and stores information about the VM flows.

IP Multicast

A new IP Multicast Application is now available to configure Protocol-Independent Multicast (PIM) routing. PIM is an IP multicast routing protocol that uses routing information provided by unicast routing protocols. Creation of Multicast VXLAN Services requires PIM configuration on devices in the VXLAN. The IP Multicast application enables you to configure PIM Global Profiles and assign them to network devices to enable PIM on the switch, and configure basic PIM parameters. The application also enables you to configure PIM interfaces and PIM Candidate Profiles on a switch. A PIM Candidate Profile is the Candidate Rendezvous Point (RP) Router and Candidate Bootstrap Router (BSR) configured on the switch.

ProActive Lifecycle Management

The ProActive Lifecycle Management Feature periodically gathers detailed information for all discovered devices on your network and periodically uploads the information to the ProActive Lifecycle Management Web Portal. A summary of Lifecycle information is also available through a widget that can be displayed on the OmniVista 2500 NMS Dashboard for easy reference.

When you install OmniVista 2500 NMS, the ProActive Lifecycle Management option is selected by default on the License Agreement Screen. If you accept this default, the feature is enabled and inventory information is gathered and sent to the web portal.

The feature can also be enabled/disabled in the Preferences Application (Preferences – System Preferences - ProActive Lifecycle Management).

mDNS

The Multicast Domain Name System (mDNS) Application is now included in the Web UI. mDNS protocol is used by "Zero Configuration Networking" solutions such as Apple's Bonjour, Avahi LGPL, and Linux NSS-mDNS. mDNS is a resolution service that is used to discover services on a LAN. mDNS allows the resolution of host names to IP addresses within small networks without the need of a conventional DNS server. The mDNS protocol uses IP multicast User Datagram Protocol (UDP) packets and is implemented by Apple Bonjour, Avahi (LGPL), and Linux NSS-mDNS. In a BYOD network, mDNS is leveraged by providing wireless guests and visitors access to network devices, such as printers.

Premium Services (BYOD)

Enabling IP Helper to Forward Endpoint Information to ClearPass

You can now enable profiling of endpoints by enabling the IP Helper function when assigning switches to ClearPass. This enables DHCP request information to be forwarded to a ClearPass Server residing on a different VLAN. When the endpoint information is learned, it is displayed in the BYOD tab of the Locator Application.

BYOD Diagnostics

A new BYOD Authentication Records Screen displays ClearPass Authentication Records. A user can view records for a specific MAC Address by entering the address on the Authentication Records Screen; or the screen can be accessed from the Locator application by selecting a MAC Address in the Netforwarding Table to view authentication records for that MAC Address. The Authentication Screen displays Authentication information retrieved from ClearPass that informs the user of the type of authentication being done for the device, the switch acting as the Network Access Device (NAD), the protocol, the service, and the result of the authentication.

Assigning Unified Policy Lists and Access Role Profiles/UNPs to ClearPass

The Access Guardian application now supports assigning Unified Policy Lists and Access Role Profiles/UNPs not just to switches, but also to ClearPass. Also, an additional set of ClearPass attributes has been added to the Access Role Profile Screen enabling configuration of a ClearPass Redirect URL and RADIUS Enforcement Profiles.

Device/Release Support

AOS 7.3.4.R01

OmniVista 2500 NMS now supports AOS 7.3.4.R01. This includes a new port number scheme for OS6900-Q32 switches, and support for the Port Split feature for OS6900-Q32 Switches.

AOS 6.6.5.R02

OmniVista 2500 NMS now supports AOS 6.6.5.R02 running all previously supported OS 6250 and OS6450 Switches, plus the following new devices. It will also discover and display the transceivers for these devices in the Modules tab in Topology as shown below:

- OS6450-P10S
- OS6450-U24S

OAW-IAP Devices

OmniVista 2500 NMS now supports OmniAccess WAN Instant Access Point (IAP) Devices (Release 6.4.2.0-4.1.1.1 and later), including support for device authentication classification in the Unified Access application. The following devices are supported:

- OAW-IAP103
- OAW-IAP104/ 105 (Certified)
- OAW-IAP114/115
- OAW-IAP134/135
- IAP-175P/175AC
- OAW-RAP108/109

- OAW-RAP155/155P
- OAW-IAP204/205 (Certified)
- OAW-IAP224/225
- OAW-IAP274/275.

Note: As of Release 6.4.2.0-4.1.1.1, it is recommended that networks with more than 128 APs should be designed as multiple, smaller Virtual Controller networks with Layer-3 mobility enabled between them. OmniVista will only manage the Virtual Controller IP. Also note that OmniVista attributes will not be completely mapped to IAP attributes. Some attributes will not be sent to the IAP because they are not supported in the device or it cannot be mapped to OmniVista.

Third-Party Aruba Devices

OmniVista 2500 NMS now supports ArubaOS Controllers and IAP Devices (6.4.2.3-4.1.1.2 and later). OmniVista supports these devices at the same level of existing wireless devices except links will not be displayed in the Topology Application because there is no LLDP or AMAP supported through SNMP on these devices.

1.4 Feature Set Support

1.4.1 Element Manager Integration

To provide additional support for various devices with different architectures, OmniVista 2500 NMS can integrate with independent Element Managers to provide direct access to devices in each class. Element Managers enable you to access, configure, and gather statistics from individual devices. The Element Managers currently supported in OmniVista 2500 NMS are listed below.

Element Manager	Supported Devices	Description
WebView	• OmniSwitch 6250-8M, 6250-24M, 6250-24M Rev. B, 6250-24MD, 6250-24MDRev. B • OmniSwitch 6400-24, 6400-P24, 6400-U24, 6400-DU24, 6400-48, 6400-P48, 6400-BPS-PS • OmniSwitch 6850, 6850-24, 6850-48, 6850-24X, 6850-48X, 6850-P24, 6850-P48, 6850-P24X, 6850-P48X, 6850 Lite Series • OmniSwitch 6855-14, 6855-P14, 6855-U10, 6855-24, 6855-U24, 6855-U24X • OmniSwitch 6860, 6860E OmniSwitch 6850E-C24, 6850E-P24, 6850E-C24X, 6850E-P24X, 6850E-C48, 6850E-P48, 6850E-C48X, 6850E-P48X, 6850E-U24X • OmniSwitch 6900-X20, 6900-X40, 6900-T20, 6900-T40, 6900-Q32 • OmniSwitch 9600, 9700, 9800 OmniSwitch 9700E, 9800E • OmniSwitch 10K	WebView is platform independent and interfaces through a web browser. It can also be invoked in the **Topology** application's All Discovered Devices table using the **WebPage** right click menu item.

1.4.2 Device Feature Support

The following table details OmniVista 2500 NMS feature support by device.

Feature	OS10K/ 6900	OS6860	Other AOS	OA WLAN	OmniAccess ESR	3rd Party Switches
Application Visibility (1)	X (2)	X				
Access Guardian (Java UI)	X		X			
Analytics (17)	X	X	X			
Basic MIB-2 Polling and Status Display	X	X	X	X	X	X (3)
ClearPass (BYOD) (18)	X	X	X			
CLI Scripting	X	X	X	X	X	X
Discovery	X	X	X	X	X	X (3)
Ethernet OAM/SAA (4)	X					
Health Monitoring	X	X	X			
Locator	X	X	X	X		X (5)
mDNS		X	X (6)			
MIB Browsing (7)	X	X	X	X	X	X (8)
PolicyView-QoS	X	X	X			
Premium Service (BYOD)		X	X			
ProActive Lifecycle Mgmt (14)	X	X	X	X		
Quarantine Manager		X	X	X		
Report	X	X	X			
Resource Manager BU/Restore/Upgrade	X	X	X			
SecureView-SA	X	X	X			
SIP (9)		X	X			
Statistics	X	X	X	Port Util Only		Port Util Only
Telnet	X	X	X	X	X	X
Topology Links (AMAP)	X	X	X			
Topology Links (LLDP) (10)			X			
Trap Absorption	X	X	X (11)	X		X
Trap Display/Trap Responder	X	X	X	X	X	X
Trap Replay	X	X	X			
Unified Access		X		X		
UNP (12)	X	X	X			
VLAN Configuration	X	X	X			
VM Manager (13)	X		X			
VM Snooping	X (15)					
VXLANs	X (16)					

1. The Application Visibility Application is available for evaluation purposes only. Contact your Alcatel-Lucent Enterprise Representative for an Application Visibility Evaluation License.

2. Application Visibility is supported on OS6900 Switches only.

3. Cisco and Extreme are supported by default. Other devices can be added manually by providing OIDs.

4. The Ethernet OAM/SAA feature is only supported on OS6900 and OS10K switches (7.3.2.R01 and later).

5. Requires MIB-2 support for 3rd-party devices.

6. AOS 6.4.6.R01 Switches only.

7. MIB browser support is for monitoring purposes only, NOT for configuration.

8. Basic MIB-2 browsing supported for 3rd-party devices. Custom MIBs may be imported and associated with 3rd party devices.

9. The SIP feature is only supported on the following devices running 6.4.5.R02 and later: 6850E (C24/24x/48/48X, P24/24X/48/48X, U24X), 6855 (U24x), 9700E (C-24/48, P24, U2/6/12/24), 9800E (C24/48, P24, U2/6/12/24).

10. LLDP is supported on OS10K/OS6900 Devices (7.x.x and later), AOS Devices (6.3.1.R01 and later), and IPD SR7x50 devices (version 9.x and later). Also note that OmniVista 2500 NMS does not display LLDP links reported by a single device. For a link to be displayed, both devices must be supported devices and LLDP MIB interface from each must have the Link.

11. Trap absorption feature is already built into AOS devices.

12. The UNP feature within Access Guardian is supported on 6250, 6450, 6400, 6850, 6850E, 6855, 6900, and OS10K devices. OS9000 and OS9000E running 6.4.3.R01 and later support VLAN-only UNP.

13. The VM Manager application is supported on OS6400, 6850E, and 6855 Switches (6.4.5.R02 and later); OS6900 (AOS 7.2.1.R01 and later) and OS10K (AOS 7.2.1.R02 and later).

14. The ProActive Lifecycle Management feature is supported on the following devices: OS10K, OS9000E, OS6900, OS6860/E, OS6850/E, OS6855, OS6450, OS6250, OAW-4005, OAW-4010, OAW-4030, OAW-4504, OAW-4604, OAW-4704, OAW-4550,OAW-4650,and OAW-4750.

15. VM Snooping is supported on OS6900 and OS10K Switches 7.3.4.R01 and later). Note that on OS10K Switches, VM Snooping is only supported on OS10K-XNI-U16E NIs. VM Snooping is supported on a port/linkagg, fixed bridge port, UNP bridge port, service access port, and UNP Service Access Point. VM Snooping is not supported on eVB, SDP, or VXLAN service ports.

16. VXLANs are supported on OS6900-Q32 Switches (7.3.4.R01 and later).

17. The Analytics feature is supported on OS6250/6450 devices (6.6.1.R01 and later), OS6850/6855 devices (6.4.1.R01 and later, OS6860 (8.1.1.R01 and later), OS6900 (7.3.2.R01 and later), OS10K (7.3.1.R01 and later).

18. ClearPass (BYOD) is supported on OS6850E Switches (AOS 6.4.6.R01 and later), OS6850E, OS6855, OS9700E, OS6250, and OS6450 (6.6.5.R01 and later), and OS6860 (8.1.1.R01 and later).

1.4.3 SSHv2/Telnet Element Management

Many devices provide element management through a user interface accessible through SSHv2/telnet. For example, you can perform element management for most Alcatel-Lucent Enterprise devices via telnet using the device's CLI (Command Line Interface). You can use OmniVista 2500 NMS to access and configure telnet capable devices. This is generally not recommended if these tasks can also be performed using OmniVista 2500 NMS. If you change device configurations without using OmniVista 2500 NMS, configuration information stored by OmniVista 2500 NMS must then be refreshed to reflect the current device configuration, using manual or automatic polling.

To access the telnet feature, select the device in the Topology application's All Discovered Devices table, right click and select the CLI Scripting menu item. Refer to the switch's manual for information on how to use the CLI.

2.0 System Requirements

The following builds are certified for OmniVista 2500 NMS 4.1.2.R02:

AOS

- OS6250 - 6.6.3.R01, 6.6.4.R01, 6.6.5.R02
- OS6400 - 6.4.4.R01
- OS6450 - 6.6.3.R01, 6.6.4.R01, 6.6.5.R02
- OS6850 - 6.4.4.R01
- OS6850E - 6.4.5.R02, 6.4.6.R01
- OS6855 - 6.4.4.R01, 6.4.6.R01
- OS6855-P14 - 6.4.4.R01, 6.4.6.R01
- OS6860 – 8.1.1.R01
- OS6900 - 7.3.2.R01, 7.3.3.R01, 7.3.4.R01
- OS9600 - 6.4.3.R01
- OS9700 - 6.4.3.R01
- OS9800 - 6.4.3.R01
- OS9700E - 6.4.4.R01, 6.4.6.R01 6.4.5.R02
- OS9800E - 6.4.4.R01, 6.4.6.R01
- OS10K - 7.3.2.R01, 7.3.3.R01, 7.3.4.R01

OmniAccess WLAN

- 6.3, 6.3.1, 6.4

OmniAccess WLAN IAP

- 6.4.2.0-4.1.1.1, 6.4.2.3-4.1.1.2

OmniAccess WAN ESR

- 5710, 5720, 5725, 5840, 5850 - Release 11

OmniVista 2500 NMS 4.1.2.R02 Upgrade Paths Certified

- 4.1.2.R01 to 4.1.2.R02
- 4.1.1 to 4.1.2.R02
- 3.5.7 to 4.1.2.R02

2.1 Requirements for All Platforms

The following sections detail requirements for all supported platforms. Please note that OmniVista 2500 NMS has been certified on English-USA versions of Windows and Linux. There is no explicit certification for International Versions of OS and non-English locale.

2.1.1 Java Requirements

OmniVista 2500 NMS includes Java 2 Runtime Environment (JRE) Version 1.8 (Update 40), which is required for the OmniVista Server. It is bundled with the installers for all supported platforms, and is automatically installed with OmniVista 2500 NMS. Because the bundled JRE is installed in the OmniVista 2500 NMS installation directory, it should NOT affect or conflict with any other JRE or Java Virtual Machine previously installed on your machine. OmniVista Client machines can be run with Java 1.7 or 1.8. See the *OmniVista 2500 NMS Installation Guide* for specific java settings required to launch the java client.

2.1.2 Server Platform Requirements

The OmniVista 2500 NMS Server should be installed on a machine with a static IP address.

2.1.3 Specific Platform Requirements

The following sections describe requirements specific to certain platforms.

2.1.3. Firewall Requirements

You must configure the firewall appropriately for OmniVista 2500 NMS to run properly on Red Hat Enterprise Linux (RHEL), 64-bit, as well as Microsoft Server 2012 R2. By default, both operating systems block TCP (except SSH), which can impact functionality of certain OmniVista 2500 NMS applications that use FTP/RMI/Telnet services of the platform (e.g., PolicyView QoS, Resource Manager, Telnet).

Note: The client and Server initiate connections on a configurable range of dynamic ports. The client PC firewall, the network firewall, and the server firewall all must be configured to allow incoming and outgoing connections between the client and server hosts over a wide range of ports.

The Omni Vista Client interacts with Server over Remote Method Invocation (RMI) Ports. In addition, OmniVista Services also interact with each other over RMI Ports. The default RMI Registry Port is 1127. For communication with the Client, the Server opens up a server socket on port 1127 and listens for incoming requests. If a request comes in, another random port is used to handle the request and send back the response. This can overload ports that are being used for other purposes, and cause problems communicating through a firewall.

OmniVista enables you to configure a range of ports that will be used for RMI communication. When a range of ports is configured, OmniVista will only use these ports as

RMI Ports, enabling you to distribute the load over a range of specific ports; and ensure communication between these ports through a firewall.

Configuring a range of RMI Ports restricts OmniVista Client/Server communications to those ports. However, other services can still use those ports. RMI Port Range ports should not be blocked by a Firewall, otherwise behavior of OV may be unpredictable.

This range can be configured on the Preferences application in the java client (default range is 3,000 – 3,200). To open the Preferences application, launch any java-based application (e.g., Discovery, Topology) from the web GUI, click on the File Menu, then select Preferences.

Note: Installation of the OmniVista Server in a Network Address Translation (NAT) environment is not supported. However, as a workaround, you can configure a site-to-site IPSec Tunnel, where the Server is one site and the Client is another site.

2.1.4 OmniVista 2500 NMS Ports

The following table lists the default ports used to communicate between the OmniVista 2500 NMS Server and Client, and the OmniVista 2500 NMS Server and network devices.

Service	Port	Source/Destination
SFTP/SSHv2	22	OV Server/OV Client
Telnet	23	OV Client/Net Device
RMI	1127	OV Server/OV Client
SNMP Request	161	OV Server/Net Device
SNMP Trap	162	OV Server/Net Device
FTP	21	OV Server/Net Device
SFTP/SSHv2	22	OV Server/Net Device
TFTP	69	OV Server/Net Device
LDAP Server	5389	OV Server/Net Device
Syslog Listener	514	OV Server/Net Device
Web Server (HTTP)	8071	OV Server/OV Client
Web Server (HTTPS)	8072	OV Server/OV Client

2.2 Recommended System Configurations

The table below provides recommended system configurations based on the number of devices being managed by OmniVista 2500 NMS 4.1.2.R02 (500, 2,000 and 5,000 devices). These configurations should be used as a guide. Specific configurations may vary depending on the network, the number of clients, the number of VLANs, applications open, etc. For more information, contact Customer Support.

Number of Managed Devices	OV Server Processor	Overall Server RAM
500	2.4 GHz Quad Core	16 GB
2,000	2.4 GHz 6 Cores	32 GB
5,000	2.4 GHz 8 or 12 Cores	64 GB
Disk Space Requirements	**Server:** 500 Devices (256 GB), 2,000 Devices (512 GB) 5,000 Devices (2.0 TB) of free disk space on the drive on which you install the OmniVista Server.	

Note: The OmniVista Server should only be installed on a **Windows 7 or 8.1** platform when managing **500 or fewer devices**. For managing more than 500 devices, install the OmniVista Server on one of the other certified Windows or Linux platforms.

3.0 Installation

OmniVista 2500 NMS is installed from a download file available on the Customer Support website. Note that you can only upgrade to OmniVista 4.1.2.R02 from OmniVista 4.1.2.R01, 4.1.1, or 3.5.7. Installation (and uninstall) procedures are detailed in the *OmniVista 2500 NMS 4.1.2.R02 Installation Guide*.

3.1 Licensing

The OmniVista 2500 NMS Core Application has a single installer and single license, with a tier-based licensing system where the user's License determines the maximum number of devices that can be managed, depending on the user's purchase. The VM Manager application is purchased separately and also has a tier-based licensing system where the user's License determines the maximum number of VMs that can be managed. A separate license is also required for the Application Visibility application, which is provided in OmniVista 2500 NMS 4.1.2.R02 for evaluation purposes only.

"Starter Pack" Core and VMM Licenses provide full functionality, but for a limited number of devices. **A "Starter Pack" License for Application Visibility is not available. Only an Evaluation License is available, on request, for Application Visibility.** The following tables provide an overview of the different license types.

Core License Types (Node Management)

	Starter Pack	Evaluation	Production
Device Count	10 AOS, or 10 AOS and 10 Third Party)	50	Chosen at license generation (Full OV functionality)
Expires	No	60 Days	No

Note: OAW Devices are counted as AOS Devices. Third-Party Aruba devices are counted as Third-Party Devices.

VMM License Types

	Starter Pack	Evaluation	Production
VMM Count	10	200	Chosen at license generation (Full VMM functionality)
Expires	No	60 Days	No

Application Visibility License Types

	Starter Pack	Evaluation	Production
Device Count	Not Available	20	Not Available
Expires	Not Available	60 Days	Not Available

The maximum number of devices allowed and the current number being managed is displayed in License Application (Administrator – License). This enables the user to determine if more devices can be added for management. Trying to discover new devices after the allowed limit will result in an Audit log and Status message.

> **Note:** Licenses are imported/upgraded in the License Application. After installing OmniVista 2500 NMS 4.1.2.R02, go to Administrator – License, import the license, then select the license type you want to upgrade/relicense and enter the License Key.

3.2 Upgrading a Starter Pack or Evaluation OmniVista License to a Production License

A Starter Pack License of the OmniVista 2500 NMS Application allows you to manage up to 20 devices (10 AOS and 10 Third-Party) with no expiration date. An Evaluation license of OmniVista 2500 NMS is valid only for a limited period of time. To gain permanent use of the OmniVista 2500 NMS software, you must order Permanent Core License. The following procedure describes how to obtain an OmniVista license key. The following procedure describes how to obtain an OmniVista license key.

1. Purchase a permanent OmniVista 2500 NMS Core License. You will receive an e-mail that contains a Customer ID and Order Number.

2. Once you receive your e-mail, log onto the Customer Support website at http://service.esd.alcatel-lucent.com/portal/page/portal/EService/LicenseGeneration and select **OmniVista 2500 NMS 4.1.2.R02**.

3. Select the product from the drop-down menu for which you want a key.

4. Enter your Site Name, Company Name, Phone, E-Mail in the required fields. The e-mail address will be used to send a valid license key to you.

5. Click **Submit**. An e-mail will be sent to you with a valid license key. A text file with the license keys will also get downloaded to your browser's default Downloads directory.

6. Make a note of the License Key.

7. Go to the **License** Application and select **Core License**.

8. Click **Relicense**.

9. Enter the license key and click **OK**. The new license will take effect immediately.

If you have questions or encounter problems upgrading your OmniVista 2500 NMS License, please contact Alcatel-Lucent Enterprise Customer Support.

4.0 Launching OmniVista 2500 NMS

To launch OmniVista 2500 NMS on Windows or Linux platforms, enter the IP address of the OmniVista Server and applicable port number in a supported web browser, for example: https://IPAddress:8072/login.html or http://IPAddress:8071/login.html.

> **Note:** The Watchdog Application, which enables all of the necessary OmniVista Services must be started to launch OmniVista. By default, Watchdog should start automatically when OmniVista 2500 NMS is installed. However, if you are having trouble launching OmniVista, check to make sure that the Watchdog Service is enabled. If it is not, enable it. It will launch the remaining OmniVista Services.

- **Windows:** Go to **Start > Control Panel > Administrative Tools > Services > OmniVista Watchdog Services**.
- **Linux:** Linux: Go to **System > Administration > Services** and start the "ovwatchdog" service.

4.1 Logging Into OmniVista 2500 NMS

After launching OmniVista for the first time, log in using the Default Username and Password:

- **Username:** admin
- **Password:** switch

5.0 Known Problems

5.1 Known General Problems

5.1.1 Print Table Function Does Not Work in JNLP-Launched Applications

The “Print Table” function does not work in JNLP-Launched applications.

Workaround: No workaround at this time.

PR# 194187

5.2 Known Access Guardian Problems

5.2.1 OV Shows Failure in Log When Removing UNP With Policy List

OmniVista displays an error in the Log File when removing a UNP with a Policy list. However, when cross checked through CLI, the UNP is successfully removed from device.

Workaround: The action failed because an Inner Tag Policy was configured on the switch, so the switch would not allow a UNP to be assigned.

PR# 181823

5.2.2 UNP-Classification Rules: Customer ID and Tag Position Should be Grayed Out for 6.4.5 R02 Devices

When configuring Classification Rules (MAC Rule, MAC-Range Rule, IPNET rule, VLAN Tag Rule) , the Customer ID field is not grayed out. The Tag position in VLAN Tag rule is also not grayed out. this is not correct for 6.4.5 R02 devices.

Workaround: When UNP-Classification Rules are assigned to 6.4.5.R02 devices, the Customer ID and Tag position fields are ignored and the columns will be empty in the View - UNP - Classification Rules View Tab in Access Guardian.

PR# 176625

5.2.3 Backup Profile Assigned for VNP

Backup UNP Profiles cannot be assigned to a 6.4.5.R02 Device as a VNP. The DA MIB does not support a Backup UNP Profile.

Workaround: Assigning UNP Profile with a Backup Profile for VNP is not supported. If a UNP Profile with a backup UNP Profile is assigned for VNP, the profiles will be assigned as two separate UNPs on 6.4.5 R02 devices.

PR# 176576

5.3 Known Analytics Problems

5.3.1 OmniVista Always Registers as sFlow Receiver "1" on Switches

If the sFlow Receiver is configured on a switch in the CLI as Receiver “1” and a user applies an Analytics Profile to the switch OmniVista overwrites the CLI-configured sFlow receiver with its own IP address as Receiver “1” (even though it leaves the configured sfFow port samplers intact.

Workaround: Using the CLI, configure other sFlow collectors to have other Receiver numbers (2 and up), leaving Receiver 1 for OmniVista to use.

PR# 205843

5.4 Known CLI Scripting Problems

5.4.1 CLI Scripting Built-In Variable Value Contains Extra Space

OmniVista CLI Scripting built-in variables replaced by their value with extra " " (space)

Workaround: These spaces help many JavaScripts work in field. If you do not want the spaces around built-in variable values, you can use JavaScript to strip them off as follows:

```
<js>
var ipAddr = "$IP_ADDRESS";
ipAddr = ipAddr.substring(1);
cli.sendCmd("show running-config tftp://10.10.110.251/"+ipAddr);
</js>
```

PR# 163776

5.4.2 CLI Scripting Guidelines

The following guidelines should be kept in mind when creating scripts for the CLI Scripting application:

- You must always use semicolons to mark the end of a line/statement.
- Multi-line comments are supported. Single-line comments ('//') are not.
- The dollar sign being used to identify user-defined variables, if you need to use it in another context, you need to go through a variable. For instance, to use it in a JavaScript variable called 'dollar': var dollar = String.fromCharCode(36)
- The <tapps>...</tapps> tags are not meant to be used for proper scripting; they are only commodity methods, allowing you to execute one command at a time. In other words, each tapps command must to have its own <tapps> tags.

For example:

<tapps>import file1</tapps>

<tapps>import file2</tapps>

Rather than:

<tapps>

import file1

import file2

</tapps>

PR# N/A

5.5 Known PolicyView Problems

5.5.1 Problems Re-Caching When Port Policy Applied to Both OS6900-X32 Switches and Non-OS6900-X32 Switches

If you mix OS6900-Q32 and other switches in a policy that contains an action on a physical port, the configuration can be applied on the wrong port on some switches. You can mix switches in a policy only if the policy does not contain any physical port in the policy action.

Workaround: If you want to create a policy with a Policy Action on a physical slot/port of OS6900-Q32 switches, do not include any switch that is not an OS6900-Q32 switch in the same policy. Create separate policies.

PR# 202737

5.6 Known Resource Manager Problems

5.6.1 OV Shows Failure When Upgrading FPGA on OS9000, But Upgrade Is Successful When Monitored Through Console

When FPGA kit is upgraded on OS9000/9000E through Resource Manager, OmniVista shows failure due to timeout, whereas FPGA kit is successfully upgraded on when the monitored through the device console.

Workaround: This is a switch issue. No workaround at this time.

PR# 186803

5.7 Known Topology Problems

5.7.1 Incorrect Transceiver Information Displayed on OS6900-X20

The Modules tab in the Topology application is displaying incorrect information for transceivers connected to OS-XNI-U12E daughter cards on OS6900-X20 devices.

Workaround: Switch issue. No workaround at this time.

PR# 187119

5.7.2 AMAP Entries for ERP-RPL Links Are Not Always Displayed

AMAP is a proprietary protocol and has been deprecated, so AMAP Entries for ERP-RPL Links are not always displayed.

Workaround: AMAP Adjacency Protocol functionality on the switch does not work properly with ERPv2 in case of ERP-RPL link, which may affect ERPv2 functionality. Use LLDP as the adjacency protocol when working with ERPv2.

PR# 177202

5.8 Known VM Manager Problems

5.8.1 SAAs Created in OmniVista Are Not Saved in boot.cfg File on OS6900/10K

The SAA configurations created in OmniVista are not written to the boot.cfg file even after write memory on OS6900/OS10 running AOS 7.3.2R01/7.3.3R01. On device reboot, OmniVista SAA configurations are lost. Only SAA configurations created with CLI/Webview (OWNER=USER) are saved.

Workaround: This is a switch issue. No workaround at this time.

PR# 189082

5.8.2 VLAN Notification Does Not Generate a Notification When Default UNP of LAG Port Is Deleted

VLAN notifications doesn't come up when the default UNP of a Link Agg Port is deleted

Workaround: This is a switch issue. When the default UNP is taken away from the LAG, the switch takes longer than usual to populate the MAC Learning Table. For a period of time, the MAC Address belong to the VM disappears and hence cannot even be located. Both commands 'show unp user' and 'show mac-learning' have no entry of the VM's MAC address. This behavior is not observed on the standard port. Notification eventually gets raised as the switch populates its table.

PR# 174181

5.8.3 VLAN Notification Does Not Generate a Notification for Missing Configuration for XenServer

VLAN Notification does not generate a notification when there is missing configuration on the switch for XenServer.

Workaround: VM Manager requires that the link discovery protocol be turned off on the port connecting the Hypervisor (ESXi Server/XenServer), or on the Hypervisor itself. Some Hypervisors may introduce LLDP packets which make it seem to have another physical bridging device, rather than an end station. In addition, if LLDP is enabled on a port, the port disposition will not be known.

PR# 173890

5.8.4 VMM Locator VM Count Can Be Greater Than VMM License VM Count or Reported by vCenter

If VMs are using multiple Physical NIC Interfaces, the same VM will be bound to different MAC Addresses and OmniVista will display multiple rows for the VM in VMM Locator search and browse applications. However, this will not affect VM Manager Licensing. The VMM License Manager will count multiple references as single Virtual Machine its UUID and the count will match the number of Virtual Machines reflected in vCenter.

Workaround: N/A

PR# 163885

5.8.5 OmniVista Treats a VM Template as a Virtual Appliance

This is working as designed. vCenter treats Virtual Machine Templates and Virtual Machines in a similar manner. A MAC address is assigned to templates and they can be converted to a Virtual Machine in a single click. vCenter returns VM Template in the list of Virtual Machines like any other VM, and OmniVista treats VM Templates like any other Virtual Machine.

Workaround: N/A

PR# 163314

5.9 Known Unified Access Problems

5.9.1 UA Policy Re-Caches Incorrectly with Policies on AOS Switch

If you notify an AOS switch on the Unified Policy List page, the switch will retrieve certain Policy Lists even if they do not contain a policy assigned to that switch, pushing empty Policy Lists to the switch. It does not affect any functionality on the AOS switch, but a stale configuration remains on the switch.

Workaround: No workaround at this time. Issue will be fixed in next release.

PR# 205481

5.9.2 Cannot Find End Station Using Upper Case MAC Address in Diagnostics

Cannot find end station using upper case MAC address when trying to locate a device on the Diagnostics Screen.

Workaround: Specify MAC address in lower case only.

PR# 205365

5.10 Known Other Problems

5.10.1 VC Takeover Affects Inventory Reporting if Switch is Added in Topology with IP Address of the EMP Port

Change of a device's Management IP address due to VC takeover causes problems with inventory reporting in the ProActive Lifecycle Management application. This happens when a VC is added in Topology with a Management IP Address that is assigned to the EMP interface on the slave chassis. This typically happens after a VC takeover scenario. The problem does not occur on a VC of 1.

Workaround: Always configure an EMP-VC IP address on the VCs (*ip interface master emp address*) or configure an "ip interface" on the device. Then, make sure that the VC is displayed with one of these IP addresses in Topology. If the VC is displayed with any physical chassis IP address (EMP-CMMA-CHAS1 or EMP-CMMA-CHAS2), change the IP Address by right clicking the device and selecting "Edit".

PR# 205556

5.10.2 No Traps Generated on 7.x/8.x when Trap Port Set to Number Other Than 162

OmniVista does not receive traps from 7.x and 8.x devices when the SNMP trap port on the switch is configured to use a port number other than 162. This is due to a switch issue.

Workaround: No workaround at this time. This is a switch issue.

PR# 198919

5.10.3 OmniVista Does Not Display Application Visibility DPI Statistics on Switches Running AOS 8.1.1

Application Visibility DPI Statistics are generated with incorrect format after upgrade from 811GA build to 811postGA build and OmniVista does not display DPI statistics.

Workaround: Login to the switch CLI and delete the files "/flash/switch/afn/dpi/dpi_flow_records.csv" and "/flash/switch/afn/dpi/dpi_flow_records.csv.old." The files will get created again with the correct format after the deletion.

PR# 197850

5.10.4 Health Application is not Showing Temperature Statistics for OS6860 and OS6900 Devices

For OSS6860 and OS6900 Devices, Health application is only showing Temperature statistics under Health->View Devices->Current average. It is not showing CPU, Memory, Rx, TxRx.

Workaround: Switch issue. No workaround at this time.

PR# 195951

5.10.5 Apostrophe Is an Invalid Character in SNMP Community String

Workaround: Remove Apostrophe from the SNMP community string.

PR# 195715

5.10.6 Unable to Access Web UI Using IP Address on I/E

Unable to access Web UI using IP address on Internet Explorer browser, locally on a Windows 2012 R2 system.

Workaround: Have the correct mapping for 'localhost' in the hosts file and use 'localhost' instead of IP address to access the Web UI locally.

PR# 194913

5.10.7 SIP Does Not Display Active Call Records on Devices Running AOS 6.4.6.R01

SIP does not display Active Call Records on devices running AOS 6.4.6.R01 even when SIP call is running successfully on device.

Workaround: No workaround at this time.

PR# 189041

5.10.8 U-Boot Version for OS6450 Devices Shows as "NA" in Inventory Report

U-Boot Version for OS6450 Devices Shows as "NA" in OmniVista Inventory Report.

Workaround: This is a hardware issue with the OS6450. No workaround at this time.

PR# 181085

6.0 Problems Fixed

6.1 Problems Fixed Since 4.1.2.R01 Maintenance Release

- When linkagg removed via CLI, UNP linkagg is deleted on switch, but not in OmniVista (PR 195702)
- Installation of OmniVista Fails with “Error: Mongo couldn’t be started” and the installation rolls back (PR 197900)

6.2 Problems Fixed Since 4.1.2.R01

- VA Upgrade - Error "The SNMP trap listener could not be created on port 162" when notification app opened (PR 201406)
- OmniVista Discovery issue for Juniper switches in VC configuration (PR 190524)
- Chassis ID option to define a port in VC missing under QoS Policy Action in OV (PR 194979)
- Clarification in color status change for Link Aggregate link status (PR 196909)
- Issue with the SPB One Touch Feature (PR 197937)
- "Max Timeout" script error seen when sending SPB Configuration Telnet Script through OmniVista (PR 199393)
- Unable to assign ClearPass Server for AOS device (6.4.4.R01) (PR 199978)
- OmniVista Tomcat Service does not start if database backup is imported from 3.5.7 through a RADIUS Server (PR 200009)
- SSLv3 vulnerability issue (PR 200391)
- Access Guardian - Create UNP Profile missing Policy List (PR 201248)
- OmniVista Server in a VA installation should be able to bind to a port lower than 1024 (e.g., 162, 514) (PR 201007)
- OmniVista should not show stack split warning icon when the stack does not support SSP and is not in loop (PR 201483)

6.3 Problems Fixed Since Release 4.1.1

- Even if GetBulk is disabled in the SNMP Settings of the Java UI, OmniVista 411 services such as Access Guardian 2.0, Application Visibility, and BYOD ignore this setting and still use GetBulk (PR 196768)

6.4 Problems Fixed Since 3.5.7 Maintenance Build

- Live Search for IP Phones Issue (PR 187956)

6.5 Problems Fixed Since Release 3.5.7 GA

- "Set Row" displays error when user logs into OV Server with multiple browser windows (PR 188220)
- Status "In Active" of Statistics profile is not correct; and the Calendar does not work when scheduling a Statistics profile (PR 188827)
- There is no log in vmm.log when manually polling the Xen Citrix Server (PR 188874)
- Error in vmm.log if the VM name contains "/" character and the VM name in VM Manager is not correct (PR 188876)
- Ethoam Preferences: Apply, Apply All, Revert buttons are not getting enabled when configuring only one Ethoam attribute (PR 189813)
- Issue when exporting information in Statistics view from VC OS6900 (PR 190523)
- Unable to start OV Server if LDAP server is not running (PR 191084)
- 64-bit OmniVista2500 3.5.7 does not detect the previously installed version during upgrade (PR 192354)

Appendix A - Sample Telnet Scripting Program

```
package com.alcatel.ov1.ws1.client;
import java.net.InetAddress;
import java.net.UnknownHostException;
import java.rmi.RemoteException;
import java.text.SimpleDateFormat;
import java.util.StringTokenizer;
import javax.net.ssl.HostnameVerifier;
import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLSession;
import javax.xml.rpc.ServiceException;
import javax.xml.rpc.Stub;
/**
* Sample Standalone client for testing Telnet Scripting Web Services
*
* @version 1.0
*/
public class TelnetScriptingClient
{

public static final String FILENAME = "TelnetScriptingData.fileName";
public static final String TIMESTAMP = "TelnetScriptingData.timeStamp";
public static final String LOG_FILENAME = "TelnetScriptingLogData.fileName";
public static final String LOG_DATE = "TelnetScriptingLogData.date";

/*
* Copied from TelnetScriptingSendResultData server object
*/
public static int NO_ERROR = 0;
public static int PARAMETER_ERROR_MISSING_VARIABLES = 1;
public static int PARAMETER_ERROR_MISSING_LOGINS = 2;
public static int RUN_ERROR = 4;

public static final int EQUALS = 0;
public static final int NOT_EQUALS = 1;
public static final int LESS_THAN = 2;
public static final int LESS_THAN_EQUAL = 3;
public static final int GREATER_THAN = 4;
public static final int GREATER_THAN_EQUAL = 5;
public static final int STARTS_WITH = 6;
public static final int ENDS_WITH = 7;
public static final int CONTAINS = 8;
public static final int OPS_SIZE = 9; // Number of operations

private boolean _ssl = false;
public TelnetScriptingClient(String[] args)
{

String testScriptName = "MyScript";

WebService1 ovWeb = null;

try {
String newScriptContent = "no more\n" +
"show system\n" +
```

```
"show chassis\n" +
"show hardware info\n";

String endPoint = "http://yourOmniVistaServerIP:8080/axis/services/OVWeb1";
String deleteScript = "Y";
String deleteLogs = "N";
String switchIp = "10.255.11.161";
String username = "admin";
String password = "yourPassword";
String secondaryPw = "yourSecondPassword";

if (_ssl == true)
{
//
// Bypass security check for self-signed peer certificate
//
HostnameVerifier hv = new HostnameVerifier() {
public boolean verify(String urlHostName, SSLSession session) {
System.out.println("Warning: URL Host: "+urlHostName+" vs.
"+session.getPeerHost());
return true;
}
};
HttpsURLConnection.setDefaultHostnameVerifier(hv);
}

int MAX_RESULTS = 500;
WebService1ServiceLocator ovWebService = new WebService1ServiceLocator();
if (endPoint != null)
{
System.out.println("Setting end point to: " + endPoint);
ovWebService.setOVWeb1EndpointAddress(endPoint);
}
ovWeb = ovWebService.getOVWeb1();
Stub stub = (Stub)ovWeb;
stub._setProperty(Stub.SESSION_MAINTAIN_PROPERTY, Boolean.TRUE);
System.out.println("Login");
ovWeb.login(username.getBytes(), password.getBytes());
System.out.println("Login succeeded");

/* We can construct sorter or filter and pass it to querying method
SortObj[] mySorters = new SortObj[1];
mySorters[0] = new SortObj(false, FILENAME); // descending sort order
FilterObj[] myFilters = new FilterObj[1];
myFilters[0] = new FilterObj(FILENAME, STARTS_WITH, "test".getBytes(), true);
*/
TelnetScriptingResultSet results = ovWeb.queryScriptFiles(null, null,
/*myFilters, mySorters,*/ MAX_RESULTS);
long cnt = results.getNumResults();
System.out.println("Result contains " + cnt + " rows.\n\n");

System.out.println("Query available scripts on the system\n");
TelnetScriptingData[] tnsData =
ovWeb.getScriptFilesData(results.getUniqueId(), 0, MAX_RESULTS);
System.out.printf("%-50s %-20s\n\n", "File Name", "Create Timestamp");
```

```
SimpleDateFormat fmt = new SimpleDateFormat("MMM dd yyyy hh:mm a");
for (int i = 0; i < tnsData.length; i++) {
TelnetScriptingData tns = tnsData[i];
long createTimeMillisec = tns.getTimeStamp();
System.out.printf("%-50s %-20s\n", tns.getFilename(),
fmt.format(createTimeMillisec) );
}

SortObj[] sorters = new SortObj[1];
sorters[0] = new SortObj(false /*descending*/, FILENAME);

ResultSet sortedResults = ovWeb.sortScriptFilesResults(results.getUniqueId(),
sorters);

System.out.println("\n============================================================
============================================================\n\n");
System.out.println("Sort objects on file names descending\n\n");
System.out.printf("%-50s %-20s\n\n", "File Name", "Create Timestamp");
tnsData = ovWeb.getScriptFilesData(sortedResults.getUniqueId(), 0,
MAX_RESULTS);
for (int i = 0; i < tnsData.length; i++) {
TelnetScriptingData tns = tnsData[i];
long createTimeMillisec = tns.getTimeStamp();
System.out.printf("%-50s %-20s\n", tns.getFilename(),
fmt.format(createTimeMillisec) );
}
System.out.println("\n============================================================
============================================================\n\n");
System.out.println("Filter objects on file names starting with 'sha' from
previous sorted result\n\n");
System.out.printf("%-50s %-20s\n\n", "File Name", "Create Timestamp");
FilterObj[] addOnFilters = new FilterObj[1];
addOnFilters[0] = new FilterObj(FILENAME, STARTS_WITH, "sha".getBytes(),
true);
ResultSet filteredResults =
ovWeb.refineScriptFilesResults(sortedResults.getUniqueId(), addOnFilters);
tnsData = ovWeb.getScriptFilesData(filteredResults.getUniqueId(), 0,
MAX_RESULTS);
for (int i = 0; i < tnsData.length; i++) {
TelnetScriptingData tns = tnsData[i];
long createTimeMillisec = tns.getTimeStamp();
System.out.printf("%-50s %-20s\n", tns.getFilename(),
fmt.format(createTimeMillisec) );
}


// Dispose of the result sets when we no longer need to fetch or do
sorting/filtering from it.
ovWeb.disposeScriptFilesResults(results.getUniqueId());
ovWeb.disposeScriptFilesResults(sortedResults.getUniqueId());
ovWeb.disposeScriptFilesResults(filteredResults.getUniqueId());
System.out.println("\n============================================================
============================================================\n\n");

System.out.println("Create a script file " + testScriptName + "\n");

// Create a test file what requires 2 user-defined variables - $parm1 and
```

```
$parm2 when it gets executed
ovWeb.createScriptFile(testScriptName, newScriptContent.getBytes());

results = ovWeb.queryScriptFiles(null, null, MAX_RESULTS);
cnt = results.getNumResults();

System.out.println("Query available scripts on the system\n");
System.out.println("Result contains " + cnt + " rows.\n\n");

tnsData = ovWeb.getScriptFilesData(results.getUniqueId(), 0, MAX_RESULTS);
System.out.printf("%-50s %-20s\n\n", "File Name", "Create Timestamp");

for (int i = 0; i < tnsData.length; i++) {
TelnetScriptingData tns = tnsData[i];
long createTimeMillisec = tns.getTimeStamp();
System.out.printf("%-50s %-20s\n", tns.getFilename(),
fmt.format(createTimeMillisec) );
}
ovWeb.disposeScriptFilesResults(results.getUniqueId());

System.out.println("\n============================================================
============================================================\n\n");
System.out.println("Get file content of " + testScriptName + "\n");

byte[] contentBin = ovWeb.getScriptFileContent(testScriptName);
String fileContent = new String(contentBin);
System.out.println("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++");
StringTokenizer strTok = new StringTokenizer(fileContent, "\r\n");
while (strTok.hasMoreTokens())
{
String line = strTok.nextToken();
System.out.print("+");
System.out.printf("%-100s", line);
System.out.println("+");
}
System.out.println("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++");

System.out.println("\n============================================================
============================================================\n\n");


TelnetSwitchInfo swInfo = new TelnetSwitchInfo();
swInfo.setSwitchIp(switchIp);
swInfo.setUsername(username);
swInfo.setPassword(password);
swInfo.setSecondaryPassword(secondaryPw);

TelnetScriptingSendRequest tnRequest = new TelnetScriptingSendRequest();
tnRequest.setSwitchInfos(new TelnetSwitchInfo[] {swInfo});

String clientIp = "";
try {
clientIp = InetAddress.getLocalHost().getHostAddress();
} catch (UnknownHostException ex) {
ex.printStackTrace();
```

```
}

tnRequest.setClientIp(clientIp);
tnRequest.setParamNames(new String[]{"$showParm1", "$showParm2",
"$tcpParm"});
tnRequest.setParamValues(new String[] {"configuration", "snapshot",
"ports"});

System.out.println("Sending Script " + testScriptName);
tnRequest.setScriptName(testScriptName);

TelnetScriptingSendResultData result = ovWeb.sendScriptFile(tnRequest);

try {
if (result == null)
{
System.out.println("Result object is null.");

}
else if (result.getErrorCode() == PARAMETER_ERROR_MISSING_LOGINS)
{
String[] missingLogins = result.getMissingLoginSwitchIps();
System.out.println("Missing Logins: ");
for (int i = 0; missingLogins != null && i < missingLogins.length; i++)
{
System.out.println(missingLogins[i]);
}

}
else if (result.getErrorCode() == PARAMETER_ERROR_MISSING_VARIABLES)
{
String[] missingParams = result.getMissingParamsNames();
System.out.println("Missing Params: ");
for (int i = 0; missingParams != null && i < missingParams.length; i++)
{
System.out.println(missingParams[i]);
}
}
else
{
long startTime = System.currentTimeMillis();
boolean isCancelled = false;

while (result.getErrorCode() == NO_ERROR && result.getUniqueId() != null &&
result.getProgress().intValue() < 100)
{
// Conditional to abort abort the operation if it takes too long
if (System.currentTimeMillis() - startTime > 120 * 1000 )
{
ovWeb.cancelSendScriptTask(result.getUniqueId());
isCancelled = true;
break;
}

result = ovWeb.getSendScriptProgress(result.getUniqueId());

// In the mean time check for progress every second
```

```
System.out.println("Progress = " + result.getProgress() + "%");
try {
Thread.sleep(1000);
} catch (InterruptedException ex) {
ex.printStackTrace();
}
}

if (result.getErrorCode() != NO_ERROR) // This could be RUN_ERROR
{
System.out.println("Error encountered: " + result.getErrorMessage());
}
else {
if (isCancelled)
{
System.out.println("Cancelled.\n\n");
}
else
{
System.out.println("Done.\n\n");
}
}


System.out.println("Telnet Scripting Send messages:");

String[] messages = ovWeb.getSendScriptEventMessages(result.getUniqueId());
System.out.println("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++");
for (String aMesg : messages)
{
System.out.print("+");
System.out.printf("%-100s", aMesg);
System.out.println("+");

}
System.out.println("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++\n\n");
}


}
finally
{
if (result != null)
ovWeb.disposeSendScriptResults(result.getUniqueId());
}

results = ovWeb.queryScriptFiles(null, null, MAX_RESULTS);
cnt = results.getNumResults();

System.out.println("Query available scripts on the system\n");
System.out.println("Result contains " + cnt + " rows.\n\n");
tnsData = ovWeb.getScriptFilesData(results.getUniqueId(), 0, MAX_RESULTS);
System.out.printf("%-50s %-20s\n\n", "File Name", "Create Timestamp");

for (int i = 0; i < tnsData.length; i++) {
```

```
TelnetScriptingData tns = tnsData[i];
long createTimeMillisec = tns.getTimeStamp();
System.out.printf("%-50s %-20s\n", tns.getFilename(),
fmt.format(createTimeMillisec) );
}
ovWeb.disposeScriptFilesResults(results.getUniqueId());

System.out.println("\n============================================================
============================================================\n\n");


System.out.println("\n============================================================
============================================================\n\n");


if ("Y".equals(deleteScript.toUpperCase())) {
System.out.println("Delete " + testScriptName + "\n\n");
int numFilesDeleted = ovWeb.deleteScriptFiles(new String[] { testScriptName
});
}

TelnetScriptingLogResultSet resSet = ovWeb.queryScriptLogFiles(null, null,
MAX_RESULTS);
cnt = resSet.getNumResults();

System.out.println("Query available telnet scripting log files\n");
System.out.println("Result contains " + cnt + " rows.\n\n");

TelnetScriptingLogData[] logData =
ovWeb.getScriptLogFilesData(resSet.getUniqueId(), 0, MAX_RESULTS);
System.out.printf("%-20s %-20s %-40s %-30s\n\n", "Ip Address", "Name",
"Filename", "Date");

for (int i = 0; i < logData.length; i++) {
TelnetScriptingLogData tsl = logData[i];
System.out.printf("%-20s %-20s %-40s %-30s\n", tsl.getIpAddress(),
tsl.getSysName(), tsl.getFileName(), fmt.format(tsl.getDate()));
}

sorters = new SortObj[1];
sorters[0] = new SortObj(true /*ascending*/, LOG_FILENAME);

sortedResults = ovWeb.sortScriptLogFilesResults(resSet.getUniqueId(),
sorters);
logData = ovWeb.getScriptLogFilesData(sortedResults.getUniqueId(), 0,
MAX_RESULTS);

System.out.println("\n\nSort log files on file names ascending\n");
System.out.println("Result contains " + cnt + " rows.\n\n");

System.out.printf("%-20s %-20s %-40s %-30s\n\n", "Ip Address", "Name",
"Filename", "Date");

for (int i = 0; i < logData.length; i++) {
TelnetScriptingLogData tsl = logData[i];
System.out.printf("%-20s %-20s %-40s %-30s\n", tsl.getIpAddress(),
tsl.getSysName(), tsl.getFileName(), fmt.format(tsl.getDate()));
```

```
}

String filterString = "MyScript";
addOnFilters[0] = new FilterObj(LOG_FILENAME, STARTS_WITH,
filterString.getBytes(), true);
TelnetScriptingLogResultSet filteredRes = (TelnetScriptingLogResultSet)
ovWeb.refineScriptLogFilesResults(resSet.getUniqueId(), addOnFilters);
cnt = filteredRes.getNumResults();
logData = ovWeb.getScriptLogFilesData(filteredRes.getUniqueId(), 0,
MAX_RESULTS);

System.out.println("\n\nFilter log files on file names starting with '" +
filterString + "' \n");
System.out.println("Result contains " + cnt + " rows.\n\n");

System.out.printf("%-20s %-20s %-40s %-30s\n\n", "Ip Address", "Name",
"Filename", "Date");

for (int i = 0; i < logData.length; i++) {
TelnetScriptingLogData tsl = logData[i];
System.out.printf("%-20s %-20s %-40s %-30s\n", tsl.getIpAddress(),
tsl.getSysName(), tsl.getFileName(), fmt.format(tsl.getDate()));
}


ovWeb.disposeScriptLogFilesResults(resSet.getUniqueId());
ovWeb.disposeScriptLogFilesResults(sortedResults.getUniqueId());
ovWeb.disposeScriptLogFilesResults(filteredRes.getUniqueId());

if (logData.length > 0)
{
String testLogFileName = logData[0].getFileName();

System.out.println("\n==========================================================
============================================================\n\n");
System.out.println("Get log file content of " + testLogFileName + "\n");

String switchIP = logData[0].getIpAddress();
String filename = logData[0].getFileName();
contentBin = ovWeb.getScriptLogFileContent(switchIP, filename);
String content = new String(contentBin);
System.out.println("\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++");
strTok = new StringTokenizer(content, "\r\n");
while (strTok.hasMoreTokens())
{
String line = strTok.nextToken();
System.out.print("+");
System.out.printf("%-100s", line);
System.out.println("+");
}

System.out.println("++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++\n\n");

if (logData != null && logData.length > 0)
{
```

```
String[] switchIPs = new String[logData.length];
String[] filenames = new String[logData.length];

for (int i = 0; i < logData.length; i++)
{
switchIPs[i] = logData[i].getIpAddress();
filenames[i] = logData[i].getFileName();
}


if ("Y".equals(deleteLogs.toUpperCase()))
{
System.out.println("Sending request to delete log files \n");
ovWeb.deleteScriptLogFiles(switchIPs, filenames);
System.out.println("Done deleting log files.");
}
}

}

System.out.println("DONE.");


}
catch (RemoteException ex) {
System.out.println("RemoteException " + ex);
}
catch (ServiceException ex) {
System.err.println("ServiceException " + ex);
}
catch (Exception ex) {
System.out.println("Exception " + ex);
}
finally
{
try
{
if (ovWeb != null)
ovWeb.logout();



}catch(Exception ex1)
{
System.out.println("Error logging out of Web Services");
}
}
}

public static void main(String[] args)
{
if (args.length < 2)
{
System.out.println("usage: TelnetScriptingClient <configFile> <scriptFile>");
return;
}
TelnetScriptingClient tnsc = new TelnetScriptingClient(args);
```

```
}
}
```