



Enterprise Management Systems Part II: Enterprise Quality of Service (QoS) Provisioning and Integration

*Deepak Kakadia, Sun Microsystems, Inc.,
Dr. Tony G. Thomas, AdventNet, Inc.,
Dr. Sridhar Vembu, AdventNet, Inc.,
Jay Ramasamy, AdventNet, Inc.
Sun BluePrints Online Article—May 2002*



<http://www.sun.com/blueprints>

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95045 U.S.A.
650-960-1300

Part No. 816-4819-10
Revision 1.0, 05/22/02
Edition: May 2002

Copyright 2002 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Sun BluePrints, Java, JavaBeans, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2002 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Sun BluePrints, et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Please
Recycle



Adobe PostScript

Enterprise Management Systems

Part II: Enterprise QoS Provisioning and Integration

Part I of this article, “Enterprise Management Systems: Architectures and Standards,” in the April issue, described how to manage services in Service Driven Networks (SDNs), provided an introduction to Enterprise Management Systems (EM Systems), and provided a good understanding of the fundamental architectures of both Sun™ Management Center 3.0 (Sun MC) and AdventNet WebNMS 2.3 software products.

This article, Part II, continues to build on this knowledge, describing how these products are integrated to provide a complete solution that can effectively manage a multivendor environment, and describes how to provision end-to-end services. This article, written for network administrators and network architects, offers a solution on how to provision complex services that may span many heterogeneous devices from a single administrative console. It also details how many of the labor intensive tasks involved in provisioning end-to-end emerging services can be automated, thus improving productivity. As an example of an end-to-end service, a Quality of Service (QoS) provisioning solution is detailed, creating a solution that is much more manageable.

This article details the following:

- Integrating Sun Management Center 3.0 software
- Integration results
- QoS provisioning
- Basic QoS provisioning procedure
- Advanced QoS provisioning procedure

Integrating Sun Management Center 3.0 Software

Integrating Sun MC 3.0 software into AdventNet WebNMS 2.3 software can be thought of as a glue that connects many vendor-specific management tools, and calls on these tools when needed. The Sun MC 3.0 software is a vendor-specific management software tool that manages Sun equipment better than any other tool. *Any* Simple Network Management Protocol (SNMP)-capable management tool can import and compile any vendor's management information bases (MIBs), then poll the device; but the Sun MC 3.0 software also provides client and server components that implement features specific to extract high resolution, detailed information providing highly specialized information about the status of a Sun device. The integration of Sun MC 3.0 software and WebNMS 2.3 software leave the Sun MC 3.0 software installation intact. Sun MC 3.0 software exposes a client API, that WebNMS 2.3 software uses as the integration point. The Sun MC 3.0 software agent and server components and console are not modified.

This section details the steps and files used to integrate the Sun MC 3.0 and AdventNet WebNMS 2.3 software products, including the extensible markup language (XML) and Java™ technology source files used to accomplish the integration efforts.

Note – This example is only for proof of concept, showing the capabilities and actual implementation in a lab environment. Please consult AdventNet, Inc. for complete detailed implementation configurations.

FIGURE 1 provides a high-level view of the key components of the integration and customization. At the top of FIGURE 1, the main AdventNet WebNMS services and XML configuration files are detailed. The various services read the configuration files in order to customize the specific service upon startup. The following steps describe the sequence of events that occur upon startup.

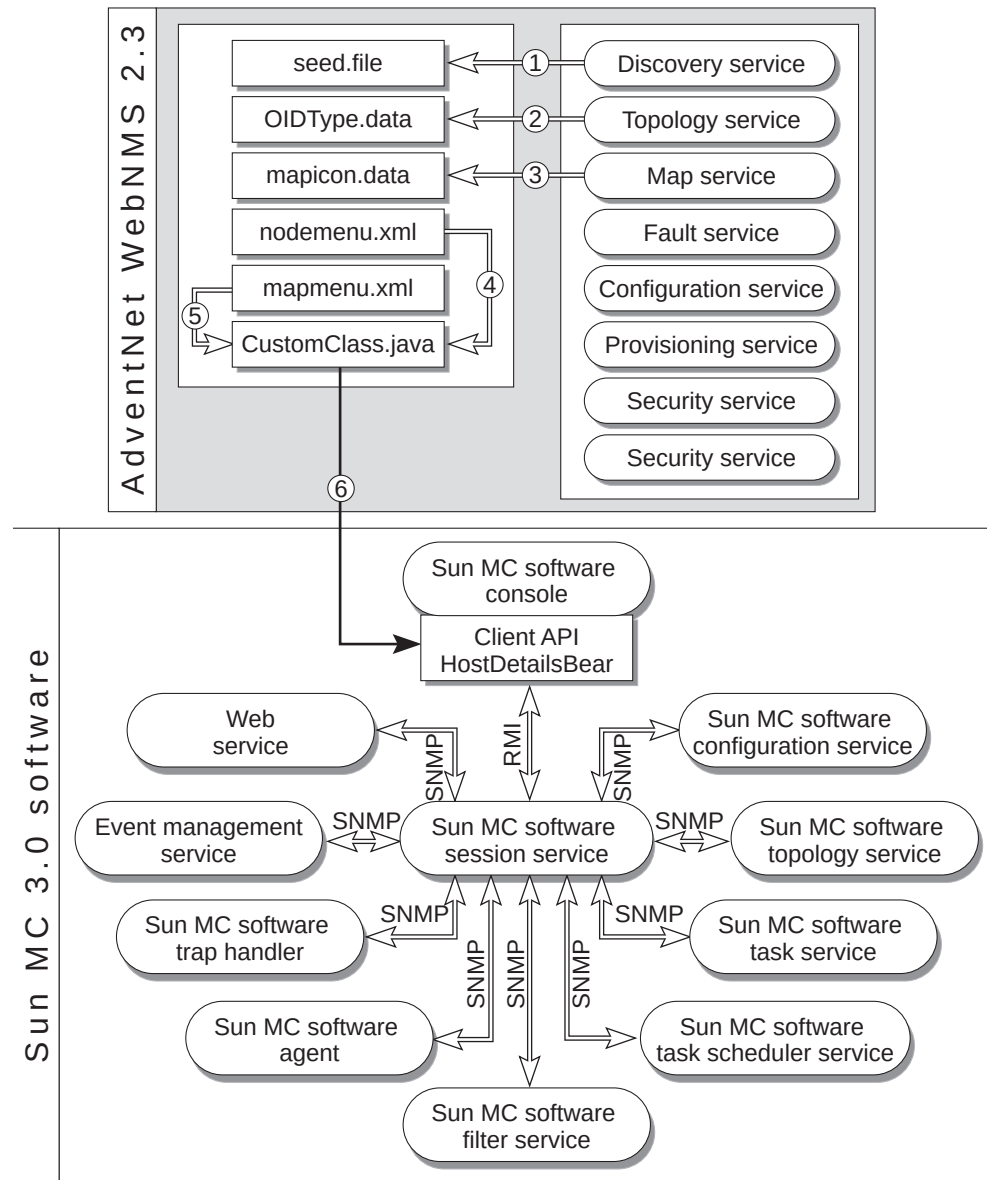


FIGURE 1 Sun MC 3.0 Software and AdventNet WebNMS 2.3 Software Integration

1. The Discovery Service starts up to discover the various devices on the network. The configuration XML file, named `seed.file`, contains the devices, times, and network ranges to control the device discovery service. The following code example illustrates an excerpt of the `seed.file` XML file.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE SEED SYSTEM 'seed.file.dtd'>

<SEED>
  **      <TO_DISCOVER>
  **      <net
  **          NETWORK_ID="192.168.1.0"
  **          NETMASK="255.255.255.0" />
  **      </TO_DISCOVER>
  **
  ** To configure multiple networks for discovery, a syntax as below should be used.
  ** Example 2:-
  **      <TO_DISCOVER>
  **      <net
  **          NETWORK_ID="192.168.1.0"
  **          NETMASK="255.255.255.0" />
  **      <net
  **          NETWORK_ID="192.168.2.0"
  **          NETMASK="255.255.255.0" />
  **      </TO_DISCOVER>
  **
  ** To discover only a range of ipaddresses in a network, using the following
  syntax.
  ** <TO_DISCOVER>
  **      <net NETWORK_ID="X.X.X.X" NETMASK="Y.Y.Y.Y" START_IP="Z.Z.Z.Z"
  END_IP="W.W.W.W" />
  ** </TO_DISCOVER>
  ** This will result in network X.X.X.X with netmask Y.Y.Y.Y being discovered.
  ** START_IP and END_IP entries allows discovery for a Range from START_IP to END_IP
  ** for the network X.X.X.X with netmask Y.Y.Y.Y
  **
  ** Example 1 :-
  **      <TO_DISCOVER>
  **      <net
  **          NETWORK_ID="192.168.1.0"
  **          NETMASK="255.255.255.0"
  **          START_IP="192.168.1.10"
  **          END_IP="192.168.1.20" />
  **      </TO_DISCOVER>
  ** means discovers from 192.168.1.10 till 192.168.1.20
  **
  ** Example 2 :-
  **      <TO_DISCOVER>
  **      <net
  **          NETWORK_ID="192.168.1.0"
  **          NETMASK="255.255.255.0"
  **          START_IP="192.168.1.10"
  **          END_IP="192.168.1.20" />
  **      <net
  **          NETWORK_ID="192.168.1.0"
  **          NETMASK="255.255.255.0"
  **          START_IP="192.168.1.30"
  **          END_IP="192.168.1.50" />
  **      </TO_DISCOVER>
  ** discovers only the nodes in the range from 192.168.1.10 to 192.168.1.20
  ** and 192.168.1.30 till 192.168.1.50 in the network 192.168.1.0
  **
```

2. The topology service consults the `OIDType.data` configuration file to determine various AdventNet WebNMS parameters of the discovered devices.

The object identifiers (OIDs) are unique codes that are retrieved from the SNMP devices, and this OID is used to associate various properties and labels in the AdventNet WebNMS software. The most common properties are the vendor and polling interval. The following code box is an excerpt of the `OIDType.data` file.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<OID_TYPE_DATA>
  <!--
    File for OID to Type conversion for discovered objects
    Also allows specifying custom classes for status testing interfaces/nodes

    Syntax:

    <DATA
      OID="1.3.6.1.2.1.1"
      TYPE="Cisco"
      POLL_INTERVAL="3600"

      USER_TESTER="userTesterClassName"
      DISC_FILTER="discFilterClassName" />

    The last two fields are optional.

  -->
  <DATA OID="default" TYPE="snmp-node" POLL_INTERVAL="3600" />
  <DATA OID="NonSnmp" TYPE="Node" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.99.1.1.1" TYPE="SNMPResearch" POLL_INTERVAL="3600" />
    <DATA OID=".1.3.6.1.4.1.9.5.18" TYPE="Cisco Catalyst" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.5.12" TYPE="Cisco WS-C2900" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.5.7" TYPE="Cisco WS-C5000" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.5.2" TYPE="Solaris" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.1.9" TYPE="CISCO" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.1.8" TYPE="CISCO" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.1.74" TYPE="CISCO" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.1.30" TYPE="CISCO" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.1.3" TYPE="CISCO" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.1.5" TYPE="CISCO" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.1.17" TYPE="CISCO" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9." TYPE="CISCO" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.9.5.18" TYPE="Cisco Catalyst" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.505." TYPE="Solaris" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.504." TYPE="Solaris" POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.11.2.3.10.1.2" TYPE="SunOS-HP Agent"
POLL_INTERVAL="3600" />
  <DATA OID=".1.3.6.1.4.1.11.2.3.10.1.1" TYPE="SunOS-HP Agent" POLL_INTERVAL="3600"
/>
  <DATA OID=".1.3.6.1.4.1.11.2.3.10.1" TYPE="SunOS-HP Agent" POLL_INTERVAL="3600" /
>
</OID_TYPE_DATA>
<?xml version="1.0" encoding="ISO-8859-1"?>

<OID_TYPE_DATA>
  <!--
    File for OID to Type conversion for discovered objects
    Also allows specifying custom classes for status testing interfaces/nodes
```

3. The map service takes the discovered devices and draws icons on the drawing canvas. The `mapIcon.data` configuration XML file associates icon images defined in the `OIDType.data` file.

For example, when a Sun device is found, it is identified by the OID, and an appropriate display icon is associated to it. The following is an example code listing.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MAP_ICON_DATA>

<!--
  This file allows specifying menus and the icons for Map objects
  based on the type of topo object.

  Syntax: The field MAP_FILTER is optional

  <DATA
    TYPE="typeName"
    MAP_FILTER="yourMapFilterClassName"
    iconName="imageFileName"
    menuName="menuFileName" />
-->

<OBJTYPES
  background="0"
  node="1"
  network="2"
  gateway="3"
  sub-symbol="4"
  site="5" />
<DATA
  TYPE="CISCO"
  menuName="routermenu"
  iconName="router.png" />
<DATA
  TYPE="Cisco Catalyst"
  menuName="hubmenu"
  iconName="hub.png" />
<DATA
  TYPE="CISCO Router"
  menuName="routermenu"
  iconName="router.png" />
<DATA
  TYPE="Cisco_WS-C2900"
  menuName="snmpmenu"
  iconName="switch.png" />
<DATA
  TYPE="Solaris"
  menuName="snmpmenu"
  iconName="sun-workstation.png" />
<DATA
  TYPE="SolCom_Ethernet_RMON_Probe"
  menuName="snmpmenu"
  iconName="probe.png" />

</MAP_ICON_DATA>
<DATA
  TYPE="SUN"
  menuName="snmpmenu"
  iconName="sun-workstation.png" />
<DATA
  (continued on next page)
```



```
(continued from previous page)
TYPE="SUN_PC"
  menuName="snmpmenu"
  iconName="sun-pc.png" />
<DATA
  TYPE="SUN_Device"
  menuName="snmpmenu"
  iconName="sun-workstation.png" />
<DATA
  TYPE="SunOS"
  menuName="snmpmenu"
  iconName="sun-workstation.png" />
```

4. The `nodemenu.xml` configuration file specifies the parameters to create dynamic menus on the console graphical user interface (GUI) and associated actions.

You can enter various properties in the `nodemenu.xml` file to define the menu item and associated action, which is a menu-driven GUI tool. In this case, the Java application, `CustomClass.java`, which was just created, is invoked. This Java application is a simple wrapper used to invoke the Sun MC 3.0 software console by using a JavaBeans™ architecture application programming interface (API).

```

<!DOCTYPE PANELMENUBAR SYSTEM "mapmenu.dtd">

<!--
    This file controls the menus.

    DESC: Below, we added a menu item for Sun MC Console, which
    when invoked, invokes the java wrapper - CustomClass.class.
    This wrapper is fed the hostname as a parameter, which
    then is fed to the sunmc client api, hostDetailsBean, which
    then invokes the Sun MC console, with a particular
    host to be managed.
-->

<PANELMENUBAR name="MIBMENU">
    <MENU name = "Custom Views"
        shortcut_key = "C">

        <MENU-ITEM name = "Add New Map"
            ACTION-ON-NO-PRIVILEGE="DISABLE"
            action_command = "Add New
Map"
            shortcut_key = "A"
            accelerator_modifier="CNTRL"
            accelerator_key="N">
        <OPERATION name="Map Editing Operations"/>
        </MENU-ITEM>

        <MENU-ITEM name = "Delete Map"
            ACTION-ON-NO-PRIVILEGE="DISABLE"
            action_command = "Delete
Map"
            shortcut_key
= "D"
            accelerator_modifier="CNTRL"
            accelerator_key="D">
        <OPERATION name="Map Editing Operations"/>
        </MENU-ITEM>

        <MENU-ITEM name = "SEPARATOR">
        </MENU-ITEM>
    ..
    ..
        <MENU-ITEM name="SEPARATOR">
        </MENU-ITEM>

        <MENU-ITEM name = "Events"
            action_command = "List
Events"
            shortcut_key = "E"
            accelerator_modifier="CNTRL"
            accelerator_key="E">
        </MENU-ITEM>

    (continued on next page)

```

```

(continued from previous page)
    <MENU-ITEM name = "Alarms"
Alerts"
    action_command = "List
</PANELMENUBAR>

    shortcut_key = "A"
    accelerator_modifier="CNTRL"
    accelerator_key="L">
    </MENU-ITEM>
    <MENU-ITEM name = "Statistics"
    action_command =
"INVOKE_CLASS:com.adventnet.nms.mapui.ListPoll?name=${name}&type=${type}"
    shortcut_key = "P"
    accelerator_modifier="CNTRL"
    accelerator_key="0">
    </MENU-ITEM>

.

    <MENU-ITEM
    name="Sun MC Console">
    <JAVA-UI
    action_command
="INVOKE_CLASS:com.adventnet.nms.examples.CustomClass?HOST=${objName}" >
    </MENU-ITEM>

```

5. Use the `mapmenu.xml` file to control another menu that invokes the wrapper `CustomClass.java`.

This configuration XML file specifies the parameters to create dynamic menus on the console GUI and associated actions. This action simply demonstrates that the user can invoke the Sun MC 3.0 software at various points. This wrapper class calls the Sun MC 3.0 software client API, `HostDetails`, `JavaBeans` specification, and is fed the `hostname` as a parameter.

```
package com.adventnet.nms.example;

//util imports
import java.util.Enumeration;
import java.util.Properties;

//WebNMS imports.
import com.adventnet.nms.util.*;

/**
 * CustomClass.java
 *
 * DESC: This program is a wrapper to invoke the SunMC console. The
 * Web NMS automagically gets the selected host from the map display,
 * retrieves the hostname and sends to this program as a parameter.
 */

public class CustomClass implements Runnable, java.beans.PropertyChangeListener,
com.adventnet.nms.util.CustomClassInterface
{
    /**
     * Implementation of CustomClassInterface.
     */

    private String host="";

    public void propertyChange(java.beans.PropertyChangeEvent p) {

        System.out.println("Prop Change = " + p);
        return;
    }

    public void setProperties( Properties p[]){

        for(int count=0;count <p.length; count++){
            for (Enumeration enum=p[count].keys();enum.hasMoreElements() ;) {

                String propertyName=(String) enum.nextElement();
                String propertyValue = (String) p[count].get(propertyName);
                (continued on next page)
            }
        }
    }
}
```

```

(continued from previous page)
System.out.println(propertyName + ":" + propertyValue);
}

host =(String)p[0].get("HOST");
Thread sunmc= new Thread(this);
sunmc.start();

}

}

/*
run thread: this thread will invoke the sunmc console,
the hostdetails bean takes the ipaddress of the sunmc
server=173.0.0.2, the port where the server is running,
which is 2099, the managed host, which is set to the
host variable, and the user and passwd of the server.

*/

public void run() {

try{

HostdetailsBean theBean = new HostdetailsBean();
theBean.init("173.0.0.2", 2099, host, "root", "gems123", null);
theBean.addPropertyChangeListener(this);
theBean.doLaunchHostdetails();

} catch (Exception e) {

e.printStackTrace();
}

}

```

6. The CustomClass.java file calls the client API of the Sun MC 3.0 software to invoke the console.

The Sun MC 3.0 software requires that the server is already running and is prepared to accept Java™ Remote Method Invocation (Java RMI) API calls that are related to invoking the HostDetails JavaBeans specification, which is the API to invoke the console.

Integration Results

FIGURE 2 illustrates the results of the discovery service detailed in the Section , “Integrating Sun Management Center 3.0 Software,” on page 2. All discovered devices, and the associated icons, are displayed. Notice the different icons for hosts and networks (see arrows 1 and 2).

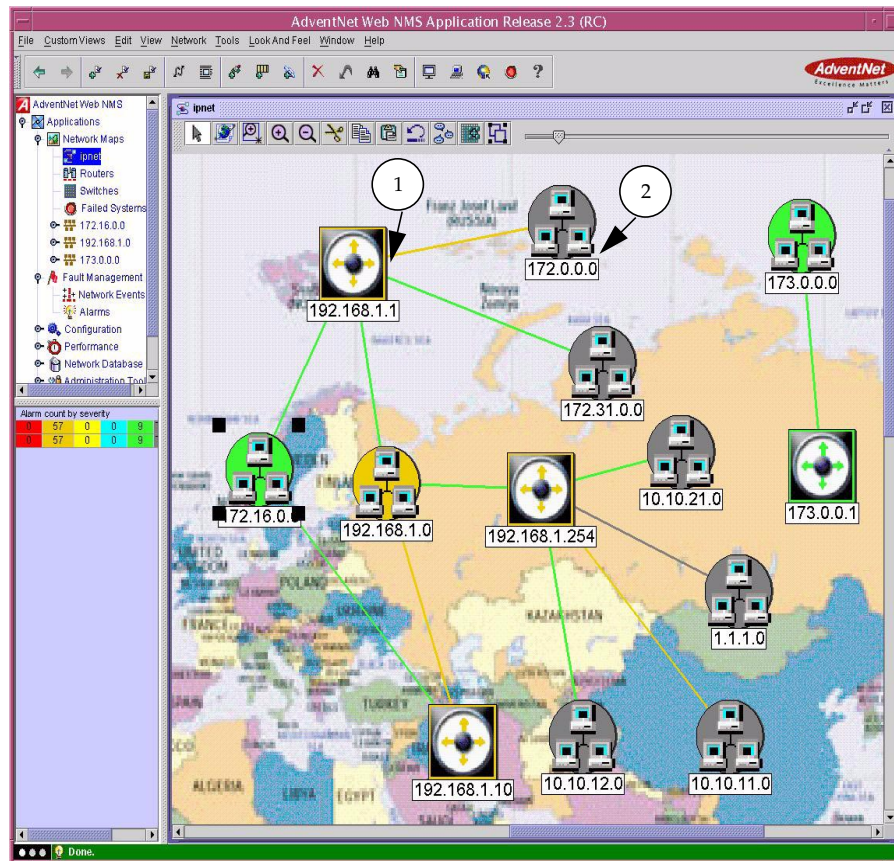


FIGURE 2 Discovery and Mapping Process

FIGURE 3 shows the results of clicking the network 173.0.0.0 network icon in FIGURE 2, and drilling down to focus on one host (173.0.0.2). The drop-down menu customization includes the Sun MC 3.0 software console. It is properly customized by using XML file modifications. This feature is a major step forward because only a few XML files had to be modified to fully integrate the third-party

NMS product (in this case, AdventNet WebNMS 2.3). The integration of the Sun MC 3.0 software into the AdventNet WebNMS 2.3 software bypassed the need to use significant coding efforts using proprietary APIs. Traditionally, this effort resulted in major costs of maintenance and support.

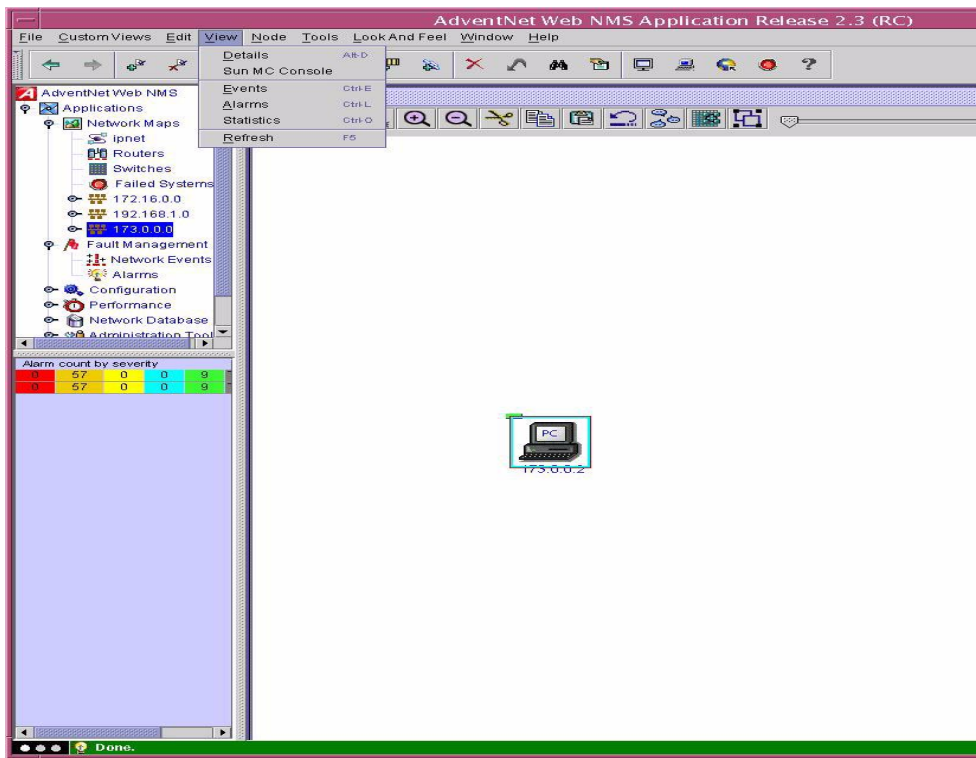


FIGURE 3 Results of Selecting the 173.0.0.0 Icon in FIGURE 2

FIGURE 4 shows the successful integration of the Sun MC 3.0 software server contacting the Sun MC 3.0 software agent on that host. Custom applications pop up, allowing powerful management of Sun devices.

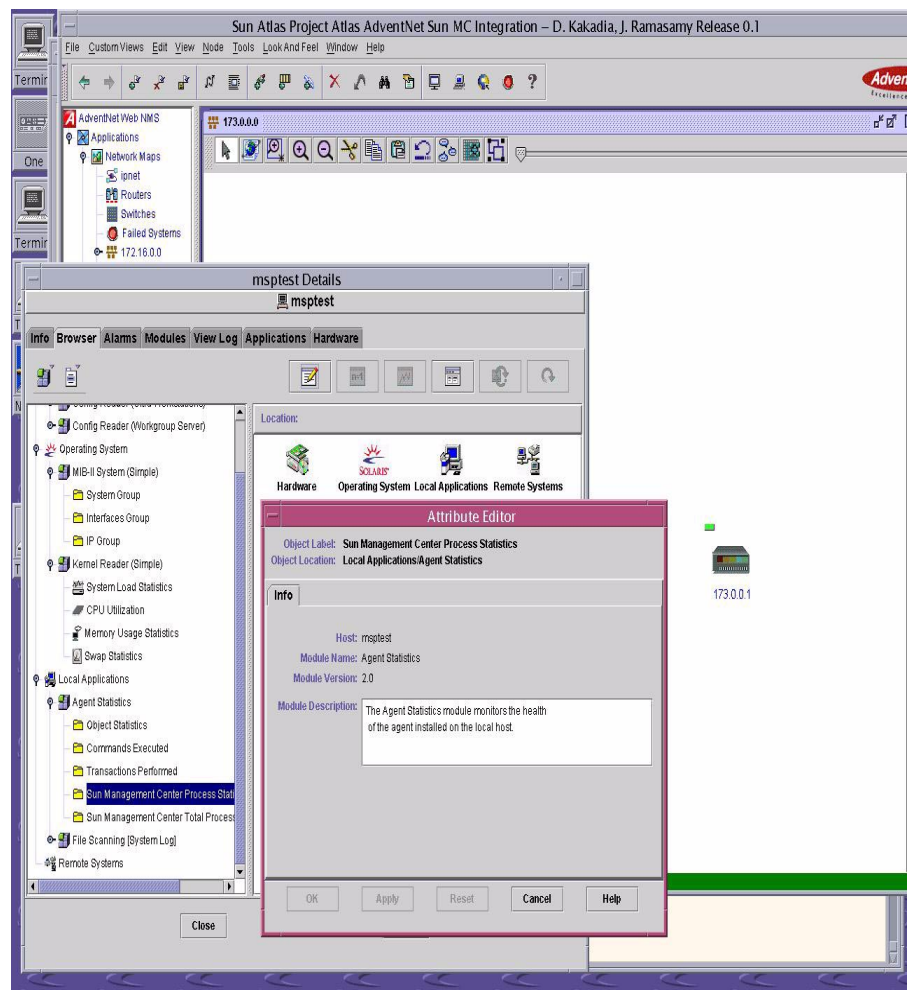


FIGURE 4 Successful Sun MC 3.0 Software Integration

QoS Provisioning

Provisioning an end-to-end service, such as QoS or a virtual private network (VPN), often requires the configuration of every device in the path between the client and server. This problem becomes much more complex when there are different devices from different vendors—each with proprietary, vendor-specific, service provisioning commands—and interfaces.

FIGURE 5 illustrates the problem and the solution from a high-level logical perspective. The diagram details the variety of networking devices that connect the client to a server. When the client is provisioned to a specific QoS, from end-to-end, each switch in the path, from a client to a server, must have the capability to identify a particular client's traffic, and prioritize that traffic in accordance with the QoS service level.

Each switch has different terminology and different commands to configure QoS. For example, Extreme Networks products uses QoS Profiles, qp1, . . . qp4, then it assigns to a port. With Cisco and Foundry products, each vendor has its own set of commands and terminology. In order for a network administrator to provision QoS manually, the command-line interface (CLI) or SNMP set commands must be used. In either case, this method requires a significant amount of labor and cost.

The proposed solution is to abstract the notion of QoS to a higher layer, such as Platinum, Gold, Silver, or Bronze, then map these high-level abstract notions to device-specific commands. When the network administrator selects the set of devices, then selects the QoS level (such as Silver), the AdventNet WebNMS 2.3 software automatically maps the selected QoS level to the particular device commands needed to provision that specific level of service.

The following sections detail the use of XML files to create these mappings. The supporting example details how elegant and powerful this solution is in this prototype.

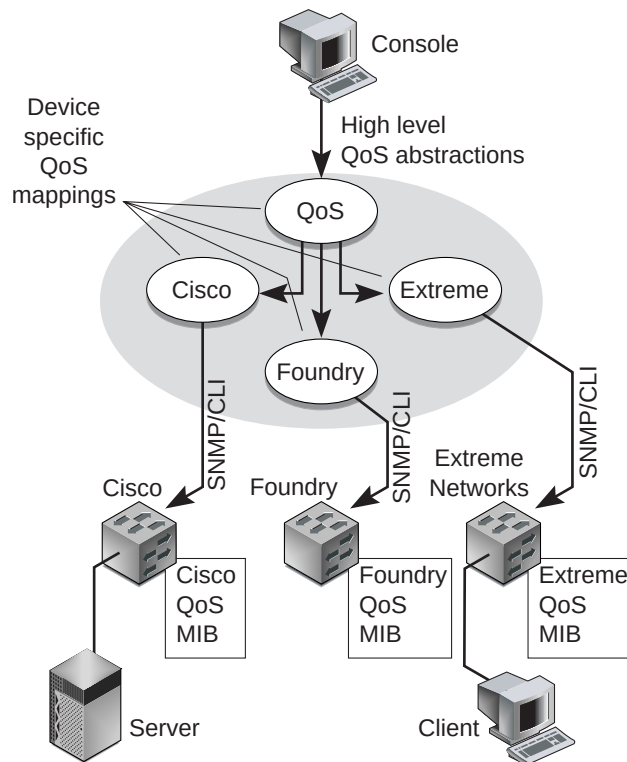


FIGURE 5 QoS Provisioning Solution in a Heterogeneous Environment

Basic QoS Provisioning Procedure

This section details the steps used to implement basic QoS provisioning, showing the relevant excerpts of the XML and Java technology source files used to create a prototype QoS provisioning solution.

1. Start the AdvenNet WebNMS 2.3 software.

The `routermenu.xml` file is read at startup time. In the code example details that follows, the menu falls under Router and after integration and has been selected for WebStack provisioning and QoS provisioning.

```

<?xml version="1.0" ?>
<!DOCTYPE MENU SYSTEM "listmenu.dtd">

<MENU
  name="Router"
  shortcut_key = "R"
  owner="root"
  group="users">

  <MENU-ITEM
    name="Router Details">
    <JAVA-UI
      action_type="openframe"

      action_value="com.adventnet.nms.examples.routermap.RouterMain?name
        =${name}">

    </JAVA-UI>
  </MENU-ITEM>

  <MENU-ITEM name = "Managed Object Properties"
    action_type="function"
    action_value="Managed Object Properties">
  </MENU-ITEM>

  <MENU-ITEM name = "SEPARATOR" />

  </MENU-ITEM>
  ..
  ..
  <MENU-ITEM
    name = "WebStack Provisioning">
    <JAVA-UI
      action_type="openframe"

      action_value="test.provisioning.TemplateNmsFrame?HOST=${name}&
        TemplateName=TelnetConfigTemplate&COMMUNITY=${community}" >
    </JAVA-UI>
  </MENU-ITEM>
  <MENU-ITEM
    name="Qos Provisioning">
    <JAVA-UI
      action_type="openframe"

      action_value="test.provisioning.TemplateNmsFrame?HOST=${name}&
        TemplateName=Extreme_Qos_Config&COMMUNITY=${community}" >
    </JAVA-UI>
  </MENU-ITEM>
  :
  ..

```

The `routermenu.xml` file invokes the Java technology program, `TemplateNMSFrame`, which takes the selected router as a parameter, specified as `HOST` and other parameter names.

As the following code example detail, this capability lets the Java technology program know which device to provision. There are two templates; the first one selects WebStack provisioning to provision the Extreme Networks switch using the CLI, and the second one selects QoS provisioning.

The WebStack provisioning allows the AdventNet WebNMS 2.3 process to Telnet and send specific CLI commands to provision the switch, while the QoS provisioning provisions the switch using the SNMP interface.

```

package test.provisioning;

import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.applet.*;

import com.adventnet.nms.util.*;
import com.adventnet.nms.startclient.NmsFrame;
import com.adventnet.nms.provisioning.xml.*;
import com.adventnet.nms.provisioning.ui.*;
import com.adventnet.nms.provisioning.*;

        public class TemplateNmsFrame extends JFrame implements NmsFrame,
        ActionListener

    {
        private boolean initialized = false;
        private java.applet.Applet applet = null;
        com.adventnet.nms.provisioning.ui.TemplateUIPanel
        TemplateUIPanel1 = null;

        public void init(JApplet applet)
        {
            try
            {
                this.applet = applet;

                String icon = applet.getParameter("FRAME_ICON");//No
                Internationalisation
                setIconImage(NmsClientUtil.getImage(applet.getDocumentBase() +
                icon));

                if (initialized == true) return;
                Container container = getContentPane();
                container.setLayout(new BorderLayout());

                TemplateUIPanel1 = new
                com.adventnet.nms.provisioning.ui.TemplateUIPanel(applet);
                container.add(TemplateUIPanel1, BorderLayout.CENTER);
                TemplateUIPanel1.clearStatus();

                TemplateUIPanel1.getProvisionPanel(applet.getParameter("TemplateN
                ame"));setTitle("Provisioning
                "+applet.getParameter("TemplateName"));

                initialized = true;
                pack();
            }
            catch(Exception exc)
            {
                System.out.println("Error inside nms frame");
                exc.printStackTrace();
            }
        }

        public void actionPerformed(ActionEvent e)
        {
            setVisible(false);
        }

        /**
        *This is called immediately after the init method. When an user selects the menu
        *item then a new instance of this class is instantiated the init method called and
        *then this method is called with the boolean true
        */

        (continued on next page)
    
```

```

        (continued from previous page)
    public void setVisible(boolean flag)
    {
        if(initialized)
        {
            NmsClientUtil.centerWindow(this);
            super.setVisible(flag);
        }
    }

    class MyWindowCloser extends WindowAdapter
    {
        public void windowClosing(WindowEvent e)
        {
            Window win = e.getWindow();
            win.setVisible(false);
            win.dispose();
        }
    }
}

```

2. Select the CLI option on the menu.

The `TelnetConfigTemplate.xml` file specifies the CLI QoS provisioning for Extreme Networks switches. This file is read by the `test.provisioning.TemplateNmsFrame` Java technology object. The following code example details this selection.

```

<?xml version="1.0"?>
<Template name="TelnetConfigTemplate" owner="root"
description="This template is an example to show how to provision QoS for
extremenetworks switch using the Command Line Interface
<Form title="Configuration via Telnet: Connection Parameters"
description="An example template to perform configuration using telnet " >
<UserInput id="1" name="TaskName" label="Task Identifier " >
default="ShowConfigTask1" />
<UserInput id="2" name="remoteHost" label="Remote Host"
default="$TemplateParam$HOST" editable="false" />
<UserInput id="3" name="telnetPort" label="Telnet Port"
default="23" >
<Qualifier type="numerictextfield" range="1-65535" />
</UserInput>

<UserInput id="4" name="loginName" label="Login Name"
default="jay" />
<UserInput id="5" name="password" label="Password"
default="" >
<Qualifier type="passwordfield" />
</UserInput>

<UserInput id="6" name="QosProfile" label="QoS Profile"
default="qp3" />
<UserInput id="7" name="Min BandWidth" label="Minimum BandWidth" default="5" />
<UserInput id="8" name="shellPrompt" label="Shell Prompt"
default="#" />
</Form>
<ConfigTask taskName="$UserInput$1" isNewTask="true"
isOverwrite="true" isSequential="false" >
<ProtocolMap name="telnet" >
<Device host="$UserInput$2" port="$UserInput$3"
loginName="$UserInput$4" password="$UserInput$5"
loginPrompt="." passwordPrompt="."
shellPrompt="$UserInput$8" retries="3"
(continued on next page)

```

```

                                (continued from previous page)
timeout="500000" />
</ProtocolMap>
<!-- The actual CLI command to provision Qos on Extreme
Networks, this is obtained
directly from the user guide manual for extremeware 6.1
the identifier is the command that is run on the telnet
session and value is prompt -->

<Attribute identifier="configure qosprofile
$UserInput$6 minbw $UserInput$7 % maxbw 99 % priority low "
value="$UserInput$8" />

</ConfigTask>
</Template>

```

3. Select the SNMP option on the menu.

The `Extreme_Qos_Config.xml` file specifies the SNMP QoS provisioning for Extreme Networks switches. The following code example details this selection.

```

<?xml version="1.0"?>
<!-- this template populates the Extreme Qos profiles -->
<!-- Deepak Kakadia, Jay Ramasamy - xml file for
extreme networks SNMP qos provisioning example -->

<Template      name="Extreme_Qos_Config"
               owner="root">
  <Form title="Extreme Qos Parameters" description="Form for
getting the Extreme Qos parameters ">
    <UserInput id="HOST" name="hostname" label="RemoteHost name"
default="$TemplateParam$HOST" />
    <UserInput id="PORT" name="port" label="SNMP Port"
default="161" />
    <UserInput id="qosmode" name="qosmode" label=
"Extreme Qos Mode" default="ingress">
    <Qualifier type="choice">
    <Enum name="INGRESS" value="1" />
    <Enum name="EGRESS" value="2" />
    </Qualifier>
    </UserInput>
    <UserInput id="pacemode" name="pacemode" label=
"Extreme Unit Pace Mode" default="NormalEthernet">
    <Qualifier type="choice">
    <Enum name="NormalEthernet" value="2" />
    <Enum name="LowLatency" value="3" />
    </Qualifier>
    </UserInput>
    <UserInput id="qosProfileIndex" name="index" label=
"Qos Profile Index" required="true">
    <Qualifier type="numerictextfield" />
    </UserInput>
    <UserInput id="qosProfileName" name="name" label=
"Qos Profile Name" required="true" />
    <UserInput id="qosMinBw" name="minBandwidth" label=
"Minimum Bandwidth" required="true">
    <Qualifier type="numerictextfield" range="0-100" />
    </UserInput>
    <UserInput id="qosMaxBw" name="maxBandwidth" label=
"Maximum Bandwidth" required="true">
    <Qualifier type="numerictextfield" range="0-100" />
    </UserInput>
    <UserInput id="qosProfilePriority" name="priority"
label="Qos Profile Priority" editable="false">
    <Qualifier type="choice">
    <Enum name="low" value="1" />
    <Enum name="lowNormal" value="2" />
    <Enum name="normal" value="3" />
    <Enum name="normalMedium"
    (continued on the next page)

```

```

                                (continued from previous page)
value="4" />
<Enum name="medium" value="5"
/>
<Enum name="mediumHi" value="6" />
<Enum name="high" value="7" />
<Enum name="highHi" value="8" />
</Qualifier>
</UserInput>
</Form>

<ConfigTask isNewTask="true" isOverwrite="true"
isSequential="false" taskName="ExtremeQosConfig" >
<ProtocolMap name="snmp" mibsToBeLoaded=
"mibs/EXTREME-QOS-MIB" >
<Device community="private"
host="$UserInput$HOST" port="$UserInput$PORT" retries="1"
timeout="500000" version="v2c"/>
</ProtocolMap>
value="$UserInput$qosmode" />
-->
<Attribute

<!-- <Attribute
identifier="extremeUnitPaceMode.0"
type="INTEGER"
value="$UserInput$pacemode" />

<Attribute
identifier="extremeQosMode.0"
type="INTEGER"

identifier="extremeQosProfileTable"
type="table"
index="$UserInput$qosProfileIndex" >
<ColumnAttribute
identifier="extremeQosProfileRowStatus"
type="INTEGER"
value="1" >
</ColumnAttribute>
<ColumnAttribute
identifier="extremeQosProfilePriority"
type="INTEGER"
value="$UserInput$qosProfilePriority" >
</ColumnAttribute>
<ColumnAttribute
identifier="extremeQosProfileMaxBw"
type="INTEGER"
value="$UserInput$qosMaxBw" >
</ColumnAttribute>
<ColumnAttribute
identifier="extremeQosProfileMinBw"
type="INTEGER"
value="$UserInput$qosMinBw" >
</ColumnAttribute>
<ColumnAttribute
identifier="extremeQosProfileName"
type="OCTET STRING"
value="$UserInput$qosProfileName" >
</ColumnAttribute>
</Attribute>
</ConfigTask>
</Template>

```

Advanced QoS Provisioning Procedure

The Section , “Basic QoS Provisioning Procedure,” on page 16 described how to provision QoS for a particular switch. This section builds on those steps and outlines how to configure QoS for a *set* of selected switches. This action is not a simple matter because each device has different QoS configuration parameters that need to be mapped from a higher layer abstraction of QoS.

Note – This example provides one possible approach, not an optimal approach, because the intent of this article is to provide an overall level of understanding.

1. Select a set of switches (from any vendor) and select QoS provisioning from the menu.

The QoS_Service_Provisioning.xml file abstracts the notion of QoS levels—Platinum, Gold, and Silver—and defines the QoS provisioning rules for Cisco and Extreme Networks switches.

An array of selected switches are saved in an object and that are accessible by the TemplateFilterExample Java technology object which is detailed in the following code example.

```
<?xml version="1.0"?>
<!-- This template populates the Extreme Qos profiles -->

<Template name="QoS Service Provisioning"
owner="root" >

  <Form title="QoS Service Provisioning"
description="Form for getting the Extreme Qos parameters " >
    <UserInput id="gosmode" name="gosmode" label="Qos Mode"
default="Silver" >
      <Qualifier type="choice" >
        <Enum name="Silver" value="1" />
        <Enum name="Gold" value="2" />
        <Enum name="Platinum" value="3" />
      </Qualifier>
    </UserInput>
    <UserInput id="pacemode" name="pacemode" label=
"Unit Pace Mode" default="NormalEthernet" >
      <Qualifier type="choice" >
        <Enum name="NormalEthernet" value="2" />
        <Enum name="LowLatency" value="3" />
      </Qualifier>
    </UserInput>
    <UserInput id="qosProfileIndex" name="index" label=
"Qos Profile Index" required="true" >
      <Qualifier type="numerictextfield" />
    </UserInput>
    <UserInput id="qosProfileName" name="name" label="Qos Profile
Name" required="true" />
    <UserInput id="qosMinBw" name="minBandwidth" label="Minimum
Bandwidth" required="true" >
      <Qualifier type="numerictextfield" range="0-100" />
    </UserInput>

  (continued on the next page)
```



```

        (continued from the previous page)
<UserInput id="qosMaxBw" name="maxBandwidth" label="Maximum
Bandwidth" required="true" >
<Qualifier type="numerictextfield" range="0-100" />
</UserInput>
<UserInput id="qosProfilePriority" name="priority"
label="Qos Profile Priority" editable="false">
<Qualifier type="choice" >
<Enum name="low" value="1" />
<Enum name="lowNormal" value="2" />
<Enum name="normal" value="3" />
<Enum name="normalMedium" value="4" />
<Enum name="medium" value="5" />
<Enum name="mediumHi" value="6" />
<Enum name="high" value="7" />
<Enum name="highHi" value="8" />
</Qualifier>
</UserInput>
</Form>
<!-- Configuration of Cisco Switches-->
<ConfigTask name="Cisco" is NewTask="true"
isOverwrite="true"
isSequential="false" taskName="CiscoQosConfig" >
<ProtocolMap name="snmp" mibsToBeLoaded=
"mibs/CISCO-QOS-PIB-MIB" >
<Device community="private"
host="$Templateparam$HOST$2" port=
"$UserInput$PORT" retries="1">
timeout="500000" version="v2c"/>
</ProtocolMap>
<Attribute
identifier="QosPolicerEntry"
type="table"
index="$UserInput$qosProfileIndex" >
<ColumnAttribute
identifier="qosPolicerId"
type="INTEGER"
value="1" >
</ColumnAttribute>
<ColumnAttribute identifier="qosPolicerRate"
type="Unsigned64"
value="" ><!--This default value can be set by
the pre-provisioning/Template filter -->
</ColumnAttribute>
<ColumnAttribute
identifier="qosPolicerNormalBurst"
type="Unsigned32"
value="$UserInput$qosMinBw" >
</ColumnAttribute>
<ColumnAttribute identifier="qosPolicerExcessBurst"
type="Unsigned32"
value="$UserInput$qosMaxBw" >
</ColumnAttribute>
<ColumnAttribute identifier="qosPolicerAction"
type="INTEGER"
value="" ><!--This default value can be set
by the pre-provisioning/Template filter -->
</ColumnAttribute>
</Attribute>
</ConfigTask>

<!-- Configuration of ExtremeSwitch -->
<ConfigTask name="Extreme" isNewTask="true"
is Overwrite="true"
isSequential="false" taskName="ExtremeQosConfig" >
<ProtocolMap name="snmp" mibsToBeLoaded=
"mibs/EXTREME-QOS-MIB" >
<Device community="private"
host="$Templateparam$HOST$1" port=
"$UserInput$PORT" retries="1">
timeout="500000" version="v2c"/>
</ProtocolMap>
<Attribute identifier="extremeQosProfileTable"
type="table"
index="$UserInput$qosProfileIndex" >
<ColumnAttribute identifier="extremeQosProfileRowStatus"
type="INTEGER"
value="1" >

```

(continued on the next page)

```

        (continued from previous page)
    </ColumnAttribute>
    <ColumnAttribute
    identifier="extremeQosProfilePriority"
    type="INTEGER"
    value="$UserInput$qosProfilePriority" >
    </ColumnAttribute>
    <ColumnAttribute identifier="extremeQosProfileMaxBw"
    type="INTEGER"
    value="" > <!-- This default value can be set
    by the pre-provisioning/Template filter -->
    </ColumnAttribute>
    <ColumnAttribute identifier="extremeQosProfileMinBw"
    type="INTEGER"
    value="$UserInput$qosMinBw" >
    </ColumnAttribute>
    <ColumnAttribute identifier="extremeQosProfileName"
    type="OCTET STRING"
    value="$UserInput$qosProfileName" >
    </ColumnAttribute>
    </Attribute>
    </ConfigTask>
    </Template>

```

2. Use the appropriate QoS provisioning tasks to provision the switch you selected in Step 1.

The `TemplateFilterExample.java` file is the main file that inputs the selected list of switches, XML definitions, and the selected level of QoS to be provisioned. The following code example details this solution.

```

package test;

import com.adventnet.nms.provisioning.server.TemplateFilter;
import com.adventnet.nms.provisioning.xml.Template;

import com.adventnet.management.config.xml.
InvalidTemplateException;

/**
 * This template will remove the config task for Cisco
 * device if the Cisco
 * Router is not selected.
 */
public class TemplateFilterExample implements TemplateFilter
{
    /** The no-argument constructor */
    public TemplateFilter()
    {
        System.out.println("Instantiating TemplateFilterExample.");
    }

    /**
     * Process the template and return the processed template
     * to be sent to the user.
     * @throws InvalidTemplateException On error or if template
     * should not be sent to the client
     */
    public Template filterTemplate(Template template) throws
    InvalidTemplateException
    {
        System.out.println("Filtering Template: \n"+template);
        Vector hostList = template.getTemplateParam("TYPE");
        boolean deleteCisco, deleteExtreme = false;
        for (int i = 0; i < hostList.size(); i++) {
            if ((String)(hostList.get(i)).equals("Extreme"))
                (continued on next page)

```

```

                (continued from previous page)
SdeleteExtreme = false;
else if ((String)(hostList.get(i)).equals("Cisco"))
deleteCisco = false;
}

XMLNode configNodeList = template.getConfigTask();
//Tasks defined in template
XMLNode userInputList = template.getUserInput();
//User Qos Selection

UserNode userInput = userInputList.get(0);
//There is only one tab in the Form
XMLAttribute usernode = userInput.getAttribute("qosmode");
//Get the access to user's choice.
String service = usernode.getValue();

if (deleteCisco)
configNodeList.deleteNode("Cisco");
else {
ConfigNode task = configNodeList.getConfigTask("Cisco");
XMLAttribute configAttribute =
task.getAttribute("qosPolicerRate");
if (service.equals("1")configAttribute.setValue("60");
else if (service.equals("2")configAttribute.setValue("85");
else if (service.equals("3")configAttribute.setValue("99");
}

if (deleteExtreme)
configNodeList.deleteNode("Extreme");
else {
ConfigNode task = configNodeList.getConfigTask("Extreme");
XMLAttribute configAttribute =
task.getAttribute("extremeQosProfileMaxBw");
if (service.equals("1") configAttribute.setValue("60");
else if (service.equals("2")
configAttribute.setValue("85");
else if (service.equals("3")
configAttribute.setValue("99");
}
}
}

```

FIGURE 6 illustrates the menu that is invoked to provision the selected device, 192.168.1.254. This basic idea of device-specific mappings provides a solution that is remarkably elegant, simple, and powerful.

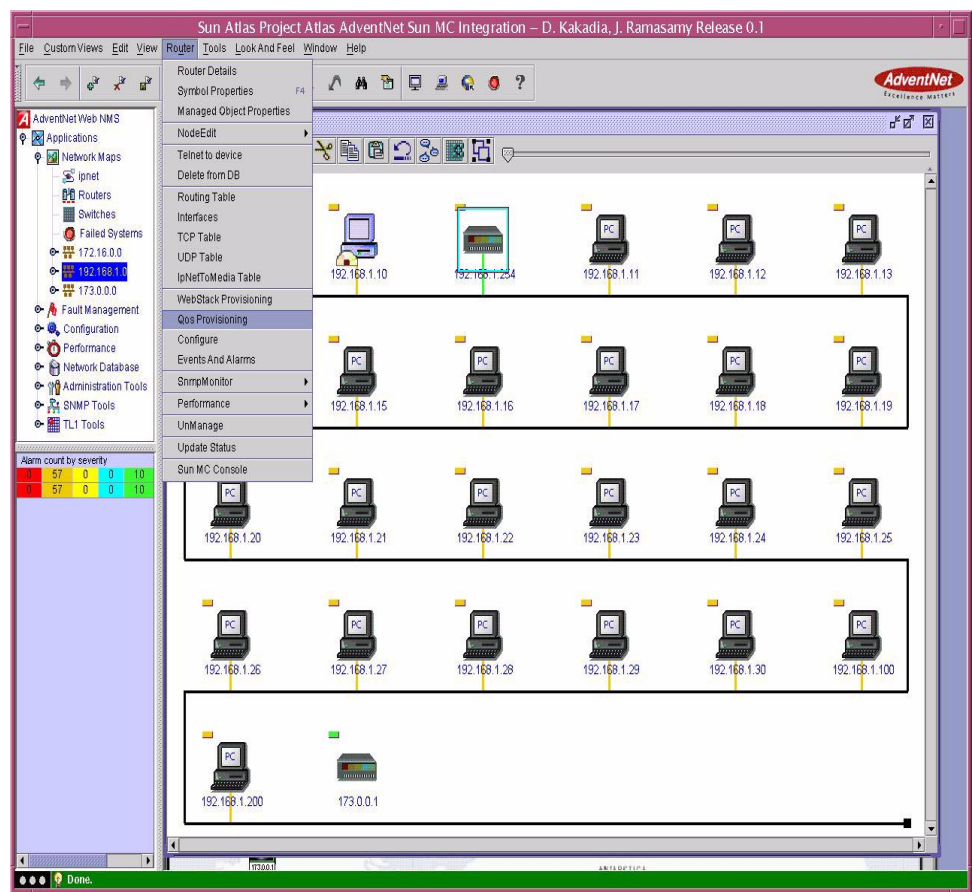


FIGURE 6 QoS Menu Configured by the routermenu.xml file

FIGURE 7 illustrates the QoS parameters inputted to provision an Extreme Networks switch defined in the `Extreme_QoS_Config.xml` file.

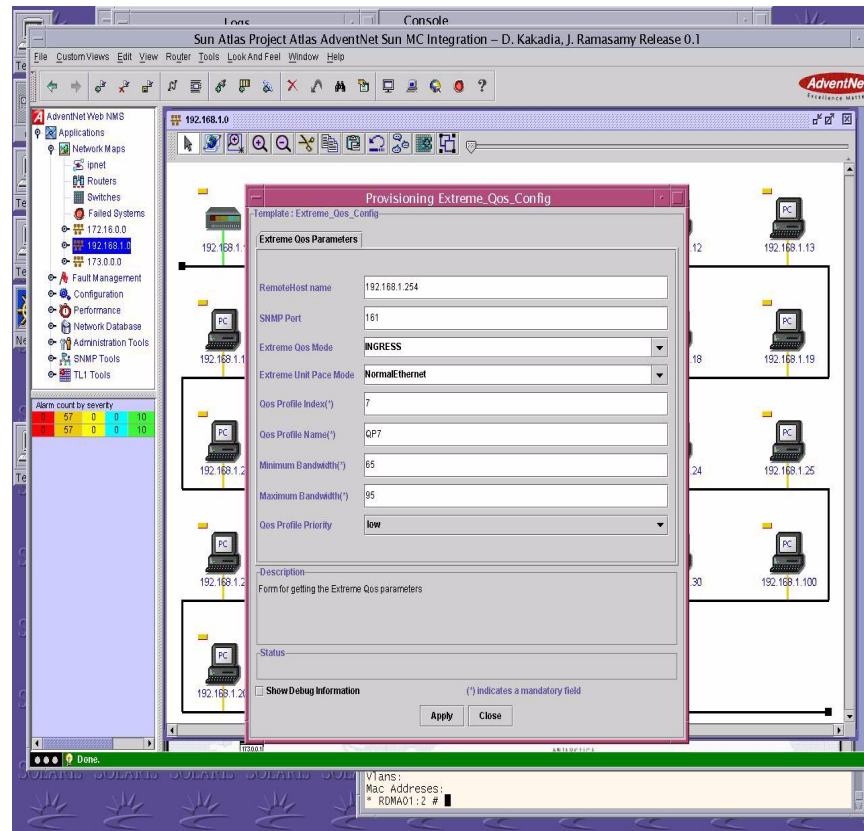


FIGURE 7 Extreme Networks QoS Provisioning Menu

The rapid advances in Service Driven Networks has introduced management challenges that have exposed the limitations of traditional network management products. There are solutions. One is creating an integrated management solution that leverages vendor-specific management products that can be controlled from one centralized console. A second solution is to provision an end-to-end service.

About the Authors

Deepak Kakadia is a staff engineer, network architect in the Enterprise Engineering Group for Sun Microsystems, Inc., located in Menlo Park, California. Deepak has been with Sun for seven years. He previously worked for various companies including Corona Networks as a principal engineer; Network Management Systems as a team leader for the QoS Policy-based NMS subsystem; Digital Equipment Corp, where he worked on DEC OSF/1; and with Nortel Networks (Bell Northern Research), in Ottawa, Canada, as member of the technical staff. Deepak received his B. Eng. in Computer Systems, M.S. in Computer Science, and completed his Ph.D. qualifying exams and course work. Deepak has filed two patents: Event Correlation and QoS in the area of Network Management.

Dr. Tony G. Thomas is the Chairman of the Board and chief architect at AdventNet, Inc. Prior to founding AdventNet, Inc., he was a member of the technical staff at AT&T Bell Laboratories in Holmdel, New Jersey. He was instrumental in developing advanced Frame Relay network management solutions at Bell Labs. He obtained his Ph.D. in Electrical Engineering from Johns Hopkins University, Baltimore, Maryland.

Dr. Sridhar Vembu is the Chief Executive Officer of AdventNet, Inc. He started his career at Qualcomm, Inc., in San Diego, where he was a wireless systems designer working on satellite-based personal communications systems. He has applied for three patents in this area. He obtained a Ph.D. in Electrical Engineering from Princeton University, Princeton, New Jersey.

Jay Ramasamy is a Senior Software Engineer with AdventNet, Inc., and has several years of Enterprise JavaBeans software development experience. He obtained his M.S. from Case Western Reserve University, Cleveland, Ohio.