HP OpenView Network Node Manager Integration with Service Information Portal

Windows® 2000, HP-UX, and Solaris

Manufacturing Part Number: J4800-90003

January 2002

Legal Notices

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty. A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend. All rights are reserved. No part of this document may be copied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DOD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

HEWLETT-PACKARD COMPANY

3404 E. Harmony Road

Fort Collins, CO 80528 U.S.A.

Use of this manual and flexible disk(s), tape cartridge(s), or CD-ROM(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

Trademark Notices.

Adobe® is a trademark of Adobe Systems Incorporated.

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Intel486 is a U.S. trademark of Intel Corporation.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Microsoft® is a U.S. registered trademark of Microsoft Corporation.

Netscape™ and Netscape Navigator™ are U.S. trademarks of Netscape Communications Corporation.

OpenView® is a registered U.S. trademark of Hewlett-Packard Company.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

Oracle Reports™, Oracle7™, and Oracle7 Server™ are trademarks of Oracle Corporation, Redwood City, California.

OSF/Motif® and Open Software Foundation® are trademarks of Open Software Foundation in the U.S. and other countries.

Pentium® is a U.S. registered trademark of Intel Corporation.

SQL*Net® and SQL*Plus® are registered U.S. trademarks of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of the Open Group.

Windows NT® is a U.S. registered trademark of Microsoft Corporation.

Windows® 2000 is a U.S. registered trademark of Microsoft Corporation.

Windows® and MS Windows® are U.S. registered trademarks of Microsoft Corporation.

All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

Contents

3. **NNM Alarms Module**

4. **Network Device Health Module**

Conventions

The following typographical conventions are used in this manual.

Font	What the Font Represents	Example
Italic	For book or manual titles, and for manpage names.	Refer to the *OVW Developer's Guide*.
	To provide emphasis.	You *must* follow these steps.
	To specify a variable that you must supply when entering a command.	At the prompt type: **rlogin** *your_name* where you supply your login name.
Bold	For glossary terms.	The **distinguishing attribute** of this class...
Computer	Text and items on the computer screen.	The Root map window ... The system replies: Press Enter
	Command names	Use the grep command ...
	File and directory names.	/usr/bin/X11
	Process names.	Check to see if pmd is running.
	Window/dialog box names	In the IP Internet map window...
Computer Bold	Text that you must enter.	At the prompt, type: **ovstatus**.
Keycap	Keyboard keys.	Press **Return**.
[Button]	Buttons on the user interface.	Click [NET]. Click on the [Apply] button.
Menu Items	A menu name followed by a colon (:) means that you select the menu, then the item. When the item is followed by an arrow (->), a cascading menu follows.	Select Edit:Find->Objects by Comment

Contact Information

Technical Support and Training

Technical support and training information can be found on the HP OpenView World Wide Web site at:

`http://openview.hp.com/`

Documentation Feedback

Your comments on and suggestions for the documentation help us understand your needs and better meet them.

You can provide feedback about documentation:

- via e-mail to: `ovdoc@fc.hp.com`, or
- via the HP documentation site at: `http://www.docs.hp.com`

If you encounter *serious errors* in the documentation that impair your ability to use the product, please contact the HP Response Center or your support representative so that your feedback can be entered into CHARTS (the HP Change Request Tracking System).

1 How NNM Works with SIP

HP OpenView Network Node Manager and SIP

HP OpenView Network Node Manager (NNM) provides up-to-date network status information that you can display to your customers through HP OpenView Service Information Portal (SIP).

For more information about NNM, itself, see *Managing Your Network with NNM* (provided with NNM software). All NNM manuals are available online at the web site:
`http://ovweb.external.hp.com/lpe/doc_serv`

You can integrate any combination of the following modules in SIP portal views. The information displayed within the modules is gathered from one NNM management station, or any number of NNM management stations and/or NNM collection stations:

- **Alarms module**
  A collection of alarm messages gathered from one or more alarm categories within NNM. You filter the alarms in a variety of ways so that only those alarms that are relevant to a particular customer are visible within each SIP portal view.

- **Network Device Health module**
  Custom gauges that track the health of network devices so that your customers can monitor network performance at-a-glance. Several predefined gauges are included, such as router health and server health. You can also write your own gauge definitions to monitor whatever is important to your customers. The data collection configuration required to run the gauges is automated and controlled through SIP configuration settings.

- **Topology (submaps) module**
  A collection of submaps from one or more NNM maps. Each map must be open on the NNM management station before the desired submap can be displayed in SIP. Drill-down through the NNM submap hierarchy can be provided, depending upon the settings in the topology configuration files.

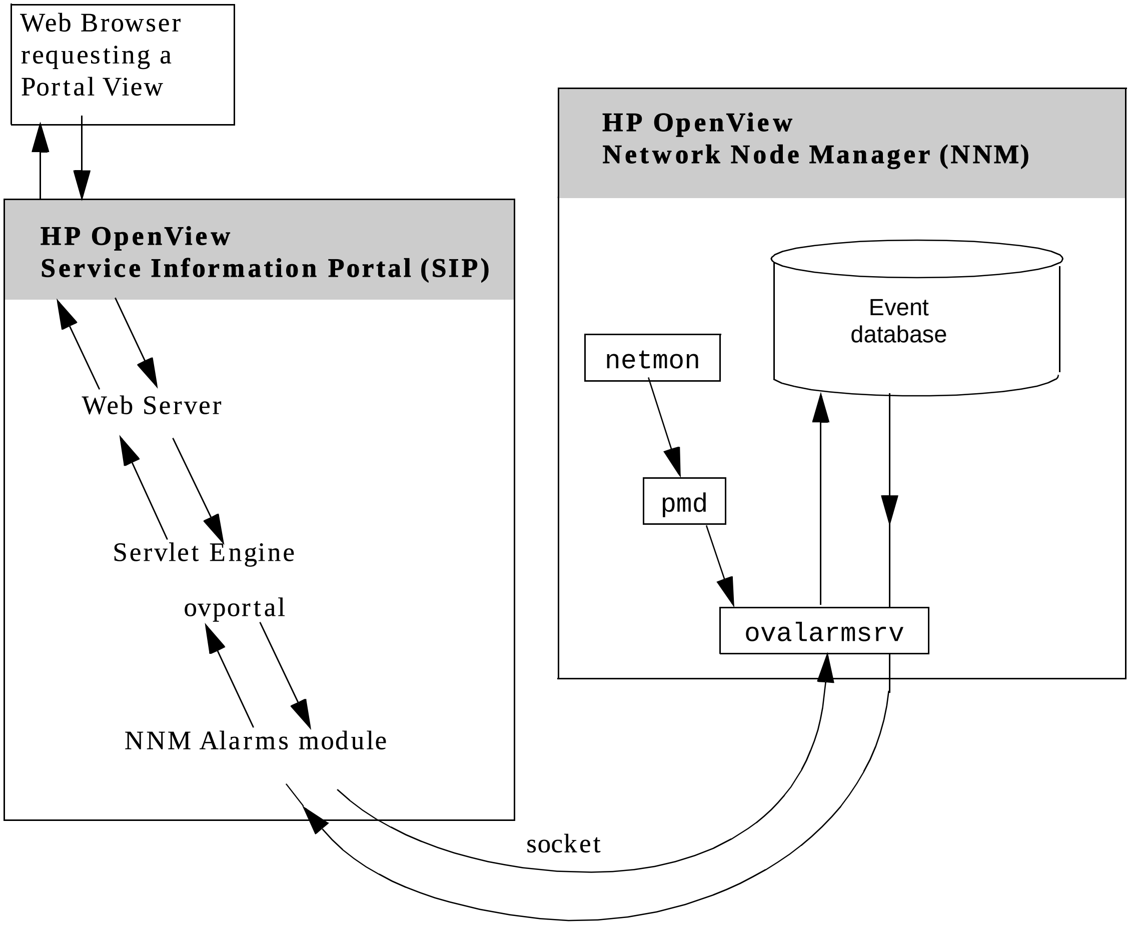
HP OpenView Customer Views and SIP

HP OpenView Customer Views runs on top of NNM. If you are using Customer Views, SIP can leverage the customer model that you have already configured (such as assignment of specific devices to a specific organization) into SIP resource mapping. For more information about leveraging the Customer Views configurations, see Chapter 6, “Segmenting the NNM Data for Your Customers,” on page 135.

Communication Paths Between NNM and SIP

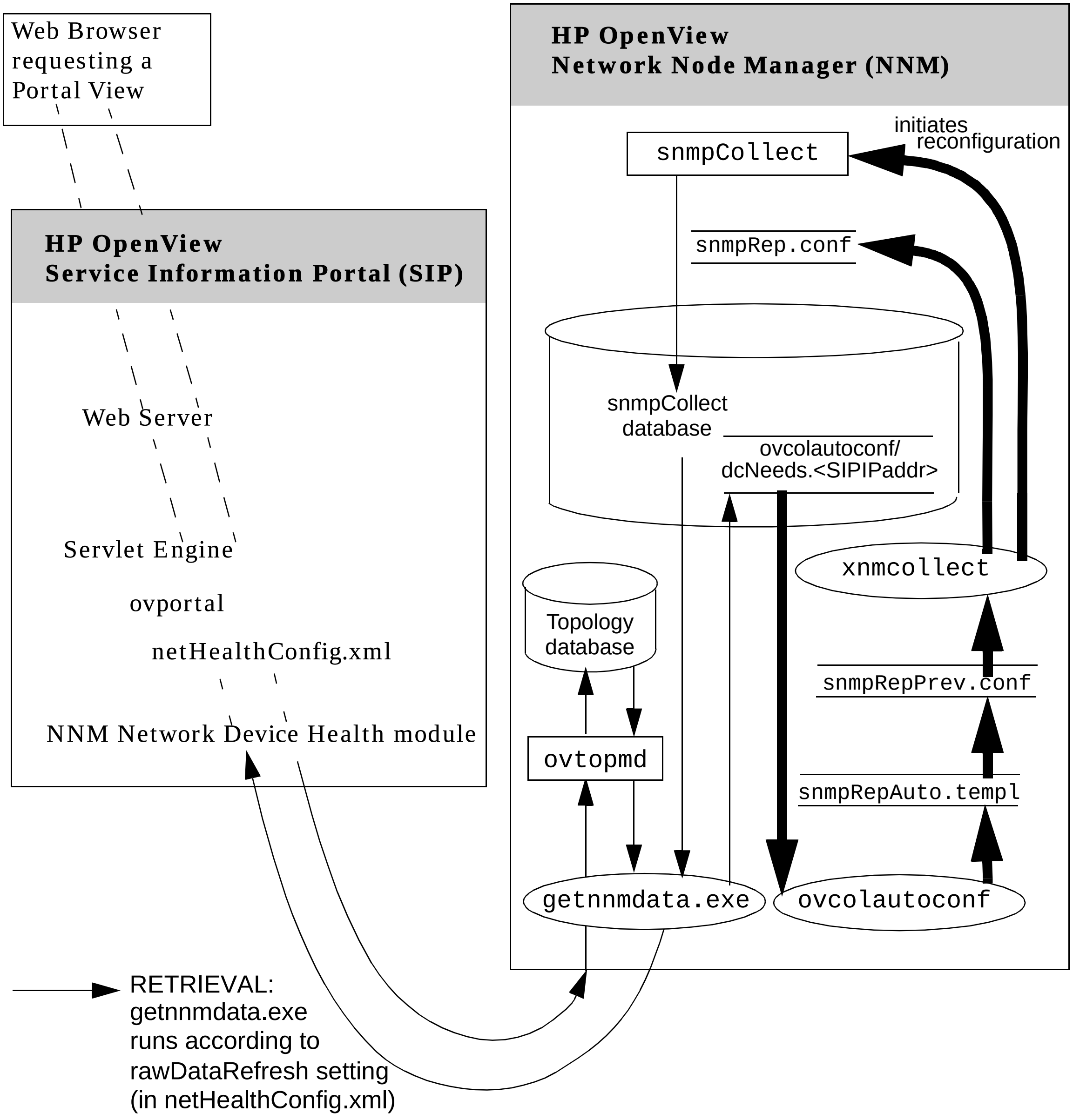
The following three diagrams illustrate the processes involved in communicating data from SIP to HP OpenView Network Node Manager (NNM) and visa versa. The following diagram illustrates the communication path between the Alarms module in SIP and NNM:

Figure 1-1 Communication Process for the Alarms Module


Web Browser requesting a Portal View
HP OpenView Service Information Portal (SIP)
Web Server
Servlet Engine
ovportal
NNM Alarms module
socket
HP OpenView Network Node Manager (NNM)
Event database
netmon
pmd
ovalarmsrv

The following diagram illustrates the communication path between the Network Device Health module in SIP and NNM:

Figure 1-2 Communication Process for the Network Device Health Module

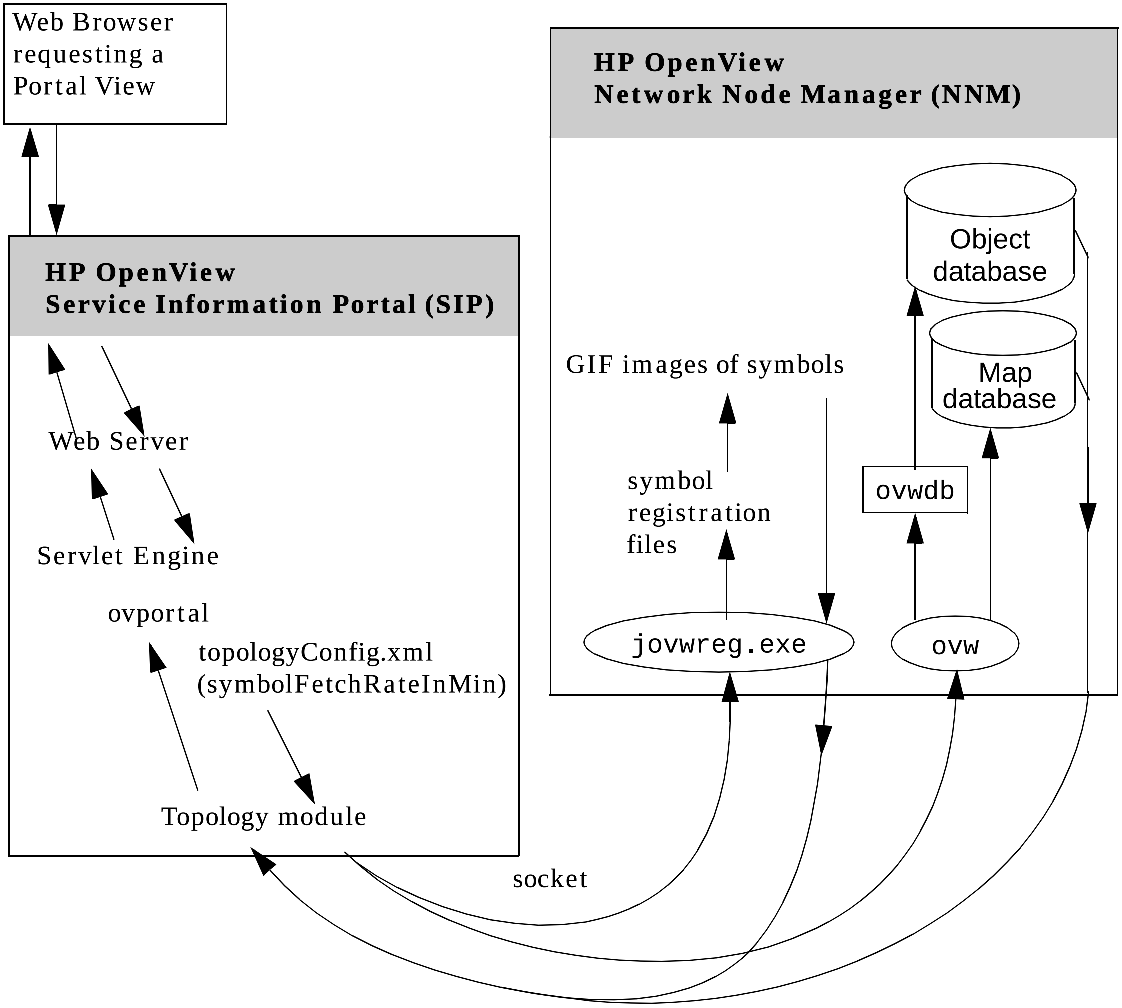

Web Browser requesting a Portal View
HP OpenView Service Information Portal (SIP)
Web Server
Servlet Engine
ovportal
netHealthConfig.xml
NNM Network Device Health module
HP OpenView Network Node Manager (NNM)
initiates reconfiguration
snmpCollect
snmpRep.conf
snmpCollect database
ovcolautoconf/ dcNeeds.<SIPIPaddr>
xnmcollect
Topology database
snmpRepPrev.conf
ovtopmd
snmpRepAuto.templ
getnnmdata.exe
ovcolautoconf

RETRIEVAL: getnnmdata.exe runs according to rawDataRefresh setting (in netHealthConfig.xml)

CONFIGURATION: ovcolautoconf must be manually initiated or scheduled by the NNM administrator

The following diagram illustrates the communication path between the Topology module in SIP and HP OpenView NNM:

Figure 1-3 Communication Process for the Topology Module


Web Browser requesting a Portal View
HP OpenView Service Information Portal (SIP)
Web Server
Servlet Engine
ovportal
topologyConfig.xml (symbolFetchRateInMin)
Topology module
socket
HP OpenView Network Node Manager (NNM)
GIF images of symbols
symbol registration files
jovwreg.exe
Object database
Map database
ovwdb
ovw

Installation of the NNM Modules for SIP

SIP can run on Windows 2000, HP-UX, or Solaris and can communicate with multiple NNM management stations and/or collection stations running on any combination of Windows NT/2000, HP-UX, and/or Solaris. For simplicity of terms, NNM management stations and NNM collection stations are both referred to as NNM management stations in SIP documentation. SIP modules provide the ability to aggregate data from multiple NNM sources to display through portal views.

The three SIP modules that display information from NNM management stations are automatically installed along with SIP. See the SIP *Installation Guide* (`SIP_Install_Guide.pdf`) for SIP installation instructions, and NNM version and patch requirements.

Before using the module, you must configure SIP and NNM to communicate with each other. See "Establishing Communication Between NNM and SIP" on page 28.

2 Configuration Steps

Establishing Communication Between NNM and SIP

To establish communication between SIP and your NNM management stations, you need to take the following steps on the SIP server and on each NNM management station.

SIP can run on Windows 2000, HP-UX, or Solaris and can communicate with multiple NNM management stations running on any combination of Windows NT/2000, HP-UX, and/or Solaris.

On the SIP Server

To enable communication between SIP and NNM, you need to establish the NNM management station configuration settings.

1. On the SIP server, open the SIP Configuration Editor:

   Windows 2000: `Start:Programs:HP OpenView->Service Information Portal->Configuration Editor`

   UNIX: `/opt/OV/SIP/bin/SIPConfig`

2. In the SIP Configuration Editor, navigate to your Management Stations definitions.
3. To add a new NNM management station, right-click the title `Management Stations`, select `New`, and type in the fully-qualified host name of the NNM management station.

   To add NNM settings to an existing management station, right-click the name of the management station, and select `Properties`.

4. Navigate to the `NNM` tab.
5. Select `NNM Is Installed On This System`.
6. Select either `NNM 6.1` or `NNM 6.2 or Later` to indicate which version of NNM is installed on this NNM management station.

   Select either `Microsoft Windows` or `UNIX` to indicate the operating system in use on this NNM management station.

   The default port settings are displayed. These are the ports that SIP uses to communicate with NNM. Verify that your NNM management station is using the default port settings. If not, change the port numbers to match the ports currently in use. Click [Help] if you need more information about NNM's port settings for the listed NNM processes.

7. To notify SIP that the NNM management station is running in a language that uses non-ASCII characters, enter a value in the `Character Encoding` field. If this field is left empty, the default locale (English) is used. This field is used only by the Topology module to notify SIP of the appropriate JDK converter 1.1 values to use.

   To determine the correct non-English encoding value, on the NNM management station, open the following file, which contains a table of JDK Converter 1.1 values for all languages:

   Windows 2000: `NNM_install_directory\www\conf\locales.jconv`

   UNIX: `/etc/opt/OV/share/www/conf/locales.jconv`

   Locate the appropriate value for this NNM management station, for example, `SJIS` for Japanese NNM running on Windows or HP-UX.

8. To enable access to data for the Network Device Health Gauge module, select `Use As SNMP Data Source`.

   When multiple NNM management stations provide raw data to the Network Device Health module, duplication of SNMP data collection is avoided by sending collection requests for a given node to only one of the NNM management stations. The NNM management station chosen is that which first returned raw data for that node. See "Collecting Data for Network Device Health Gauges" on page 105 for more information.

9. To enable access to data for the Alarms module, select `Use As Alarms Data Source`.

10. To enable access to symbol images for the Topology module, select `Use As OVw Symbol Source`. SIP must gather NNM symbol image files at least once. This field indicates that GIF images will be gathered from the NNM management station according to the frequency specified by the `symbolFetchRateInMin` attribute in the `topologyConfig.xml` file (default value is 1 day). Selecting this field instructs SIP to check for and gather the updated GIF images. Because symbol images do not often change, you may want to minimize traffic by selecting `Use As OVw Symbol Source` for only

one of your NNM management stations. (Please note that an NNM management station can be used as the source for topology map information even if OVw Symbol Source is not selected.)

To see which symbols are currently available on a particular NNM management station, type the following URL into your web browser:

NNM Management Station Running on Windows:

```
http://<NNM_management_station_hostName>/OvCgi/jovwreg.exe?symbols
```

NNM Management Station Running on UNIX:

```
http://<NNM_management_station_hostName>:8880/OvCgi/jovwreg.exe?symbols
```

11. Click `[OK]` when you are finished making changes.
12. Repeat from step 2 for each NNM management station with which SIP should communicate.
13. Navigate to your Role definitions (other than `SIP Administrator`).
14. For each role that needs access to the NNM modules, right-click the role name and select `Properties`.
15. Navigate to the `Management Data` tab and make the appropriate selection. The Management Data filter controls the flow of data from your NNM management stations into the NNM modules. See Chapter 6, “Segmenting the NNM Data for Your Customers,” on page 135 if you need more information.
16. Repeat from step 13 for each role that needs access to NNM modules.
17. Save your changes and exit the SIP Configuration Editor.
18. *This step is only required if your NNM management station is running NNM version 6.1 (not required for NNM 6.2):*

    You need to copy the following executable file to a newly created temporary directory on each NNM 6.1 management station that you listed in the `mgmtStations.xml` file. This executable installs several required files (these files ship with NNM 6.2):

    - Copy the following file **from** SIP running on *Windows 2000* for NNM running on:
        - *Windows NT/2000*: `%SIP_HOME%\cgi-bin\WindowsNT\installCGIs.zip`
        - *HP-UX*: `%SIP_HOME%\cgi-bin\HP-UX11\installCGIs.tar.Z`

  - *Solaris*:
    `%SIP_HOME%\cgi-bin\Solaris2.X\installCGIs.tar.Z`

- Copy the following file **from** SIP running on *UNIX* for NNM running on:

  - *Windows NT/2000*:
    `/opt/OV/SIP/cgi-bin/WindowsNT/installCGIs.zip`
  - *UNIX*:
    `/opt/OV/SIP/cgi-bin/HP-UX11/installCGIs.tar.Z`
  - *Solaris*:
    `/opt/OV/SIP/cgi-bin/Solaris2.X/installCGIs.tar.Z`

On each NNM Management Station

NOTE Verify that you are using a version of HP OpenView Network Node Manager that is supported by SIP, see the SIP *Installation Guide* (`SIP_Install_Guide.pdf`) for the list of supported product versions and the latest consolidated patch.

1. Add the SIP server hostname(s) to the following two files. In an ASCII editor, open the following two authorization configuration files (skip this step if SIP is running on the same computer as NNM):

   - The `ovw.auth` file controls which hosts and users are authorized to connect to NNM sessions running on the management station:

     - *Windows NT/2000*:
       `*NNM_install_dir*\conf\ovw.auth`
     - *UNIX*:
       `/etc/opt/OV/share/conf/ovw.auth`

   - The `ovwdb.auth` file controls which hosts and users are authorized to connect to the NNM database processes:

     - *Windows NT/2000*:
       `*NNM_install_dir*\conf\ovwdb.auth`
     - *UNIX*:
       `/etc/opt/OV/share/conf/ovwdb.auth`

TIP If you see a line that simply has two + symbols (+ +), you can skip this step because NNM is configured to allow any computer to request information (security not implemented).

Add a ***`SIPserverHostName`*** **`+`** line to the list for each SIP server that needs to obtain information from this NNM management station.

2. *This step is only required if your NNM management station is running NNM version 6.1 (not required for NNM 6.2):*

   At the command prompt, navigate to the `installCGIs.zip` or `installCGIs.tar.Z` file that you placed on this NNM 6.1 management station in the previous section.

   a. Unzip or uncompress and untar the file.

   b. At the command prompt, type:

   - *Windows NT\2000*:

     ```
     NNM_install_dir\bin\Perl\bin\perl.exe installCGIs.pl
     ```

   - *UNIX*:

     ```
     /opt/OV/bin/Perl/bin/perl installCGIs.pl
     ```

   c. You can now remove the `installCGIs` file and the directory structure around it.

To enable Topology Module Access to NNM Data

Each desired NNM map must be open on the NNM management station before submaps can be displayed in the SIP portal view. Only submaps currently displayed on the NNM management station or *persistent* submaps (those stored in memory on the NNM management station) can be selected for display in the Topology module. However, submaps accessed through drill-down do not need to be *persistent*.

1. NNM (ovw) must be running on the NNM management station containing the map you wish to display in a SIP portal view:

TIP Check the following file for information about running NNM in a virtual window so that you don't have to keep every SIP map open in an ovw session. On the SIP server: `SIP/htdocs/WhitePapers/VirtualWindow-NNM.html`

Windows NT/2000:

- Start the NNM services (if necessary) by clicking `Start:Programs->HP OpenView->Network Node Manager Admin->NNM Services-Start`.
- Start the NNM interface by clicking `Start:Programs->HP OpenView->Network Node Manager`.

UNIX:

- To start the NNM background processes, log in as `root` and type:

  `/opt/OV/bin/ovstart -c`

- To start the NNM interface, type:

  `/opt/OV/bin/ovw`

2. Open an NNM session for each map (such as the *Default* map) that will be accessed through the Topology module in the SIP portal.
3. Ensure that each submap that you wish to select for display in the Topology module (such as the *Internet* submap) is set to *persistent* (stored in RAM, not *transient* -- generated upon request).

NOTE Submaps accessed through drill-down do not need to be *persistent*.

To check or change persistence, do one of the following:

- Configure the IP Map application to enable the on-demand level:

  Windows NT/2000: `Map:Properties`. From the `Applications` tab, double click on `IP Map` and select an `On-Demand` level.

  UNIX: `Map:Properties`. Select `IP Map`, click `[Configure For This Map]`, and select an `On-Demand` level.

- Make the individual submap persistent:

  Windows NT/2000: `Map:Submap:Properties`. From the `View` tab, select the `Persistent` check box.

UNIX: `Map:Submap:Make the Submap Persistent`.

4. Create and customize any desired maps and submaps. (For information about map customization, see the NNM manual *Managing Your Network with NNM*.)

   Consider creating a few general purpose submaps, and using SIP Management Data filters to display only the information that is important for a specific customer (see Chapter 6, “Segmenting the NNM Data for Your Customers,” on page 135).

5. A submap’s background graphic, if any, can be automatically displayed in SIP. The graphic must be in either JPEG or GIF format, and the graphic must be placed in the following location on the NNM management station:

   - *Windows NT/2000*: *`NNM_install_dir`*`\backgrounds\*`
   - *UNIX*: `/usr/OV/backgrounds/*`

   To add a background graphic to a submap:

   a. On the NNM management station, open the NNM submap to which you wish to add a background graphic.

   b. Select `Map`: `Submap->Properties`.

      Only Windows NT/2000: click the `View` tab.

   c. In the `Background Graphics` list, select the graphic you want to apply to the current submap, then click the `[OK]` button.

   For more detailed information, see the NNM’s online help or the *Managing Your Network with NNM* manual.

6. If the NNM management station is restarted, you must restart each NNM session to display the submaps in the SIP portal views.

To enable the Network Device Health Module to Configure NNM Data Collection

The Network Device Health gauges that ship with SIP require NNM to collect data using several SNMP MIB expressions. MIB expressions are a feature of Network Node Manager that allow for the creation of mathematical formulas comprised of MIB objects. MIB expressions allow you to derive more meaningful information than you could gather from individual MIB objects.

As long as you do not already have MIB expressions with the same names as the MIB expressions provided with SIP, SIP data collections will not conflict with any current settings in the NNM Data Collector.

On the SIP server:

Open a SIP portal and display at least one Network Device Health module. Leave this portal open while completing the following steps on the NNM management station.

On the NNM management station:

1. To verify that you do not already have MIB expressions by these names, open the NNM Data Collections and Thresholds window by either:

   - Typing the following at the command prompt:

     ```
     xnmcollect
     ```

   - *Windows NT/2000:* Selecting `Edit:MIB Object->New.Clicking the Expressions radio button to display the list of loaded MIB expressions.`

     UNIX: Selecting `Edit:Add->MIB Object.Clicking the Expressions radio button to display the list of loaded MIB expressions.`

   Scroll down the list and check for these MIB expression names:

   - p_if%util
   - p_if%inerrors
   - p_if%outerrors
   - p_cisco5minavgbusy

2. To load SIP's MIB Expressions, at the command prompt type (no hard returns included):

   - *Windows NT/2000:*

     ```
     NNM_install_dir\bin\xnmcollect.exe -loadExpr
     NNM_install_dir\conf\ovcolautoconf\mibExprAuto.conf
     ```

   - *UNIX:*

     ```
     /opt/OV/bin/xnmcollect -loadExpr
     /etc/opt/OV/share/conf/ovcolautoconf/mibExprAuto.conf
     ```

3. To verify that these MIB expressions were successfully loaded, open the NNM Data Collections and Thresholds window by either:

   - Typing the following at the command prompt:

     `xnmcollect`

   - *Windows NT/2000:* Selecting `Edit:MIB Object->New.Clicking the Expressions radio button to display the list of loaded MIB expressions.`

     UNIX: Selecting `Edit:Add->MIB Object.Clicking the Expressions radio button to display the list of loaded MIB expressions.`

   Look for these new MIB expression names in the list:

   - p_if%util
   - p_if%inerrors
   - p_if%outerrors
   - p_cisco5minavgbusy

   To learn about the mathematical formulas behind the SIP MIB expressions, highlight the name and click [`Describe`].

   For interface metrics, different formulas are used depending upon the attributes of the interface (such as speed of the interface, half-duplex versus full-duplex, etc.). In the case of CPU utilization, the expression is really just a single Cisco MIB object: `local.system.augBusy5`. It is described as the “5 minute exponentially-decayed moving average of the CPU busy percentage.”

   For more information about MIB Expressions, select `Help:Online Manuals-->Managing Your Network with NNM`.

NOTE

Continue with the remaining steps in this section if you wish to enable *automatic* configuration of NNM’s Data Collector to meet SIP data requirements. For information about how this process works, see “The Data Collection Process for the Network Device Health Module” on page 40.Otherwise, stop here and see “Manually Configuring NNM’s Data Collector to Provide the Required SIP Data” on page 38.

4. To enable *automatic* configuration of SIP's data collection requirements, create the following directory:

   - *Windows NT/2000:*

     NNM_install_dir\databases\snmpCollect\ovcolautoconf

   - *UNIX:*

     /var/opt/OV/share/databases/snmpCollect/**ovcolautoconf**

   After a few minutes (10 by default), SIP populates the ovcolautoconf directory with one or more dcNeeds.*SIPServerIPAddress* files containing the current list of data collection requests from open portal views on each SIP server. These files are created by getnnmdata.exe (see Figure 1-2 on page 23).

5. *For UNIX only:* Make the ovcolautoconf directory writable by the web server process used by NNM (such as apache's httpd) and the user or scheduler program responsible for running the ovcolautoconf command (see the next step). Verify that the directory you just created has the permissions set correctly. For example:

```
drw-rw----   2 bin        adm             24 Aug  9 16:01 ovcolautoconf
```

   This UNIX permissions example allows the web server running as user bin (the default for NNM on HP-UX and Solaris) to write to the ovcolautoconf directory, and allows a user with bin permissions or a cron job running as bin to write to the ovcolautoconf directory.

6. To update NNM's Data Collector configuration files, run ovcolautoconf on each NNM management station. The ovcolautoconf command must be executed on the NNM management station either manually or as a scheduled task that you define. At the command prompt, type one of the following:

   ovcolautoconf or

   ovcolautoconf -verbose

   ovcolautoconf creates the snmpRepPrev.conf file in the ovcolautoconf directory. In this file all SIP requests are formatted so that they can be uploaded into NNM's Data Collector configuration files. This file is a record of the most recent configurations uploaded from SIP into the NNM snmpRep.conf file.

TIP To change the number of days SIP waits before deleting any inactive data collection configurations (default 30), type the following command. There is no way to permanently change this setting. Include this command in your scheduled script or each time you manually run `ovcolautoconf`:

```
ovcolautoconf -maxConfAge #ofdays
```

7. You can modify the SIP collection configurations; for example, change collection intervals (15 minutes by default) or add thresholds. To modify the SIP collection configurations, edit the `snmpRepAuto.templ` file. This file is a template used by the `ovcolautoconf` program when formatting SIP data collection requests for NNM's Data Collector program. It contains one entry for each MIB object or MIB expression upon which NNM's Data Collector needs to collect data.

   To view the list of configured collections and make any necessary changes, at the command prompt type the following (no hard returns included):

   - *Windows NT/2000*:
     ```
     xnmcollect -snmpColConfFile
     NNM_install_dir\conf\ovcolautoconf\snmpRepAuto.templ
     ```
   - *UNIX*: log in as root and then type,
     ```
     xnmcollect -snmpColConfFile
     /etc/opt/OV/share/conf/ovcolautoconf/snmpRepAuto.templ
     ```

   Review the list. In the *Source* field you will see the variable _NODE_, which is automatically replaced with any specific devices requested by SIP. Do not change the *Source* field variable. You may change the other fields.

Manually Configuring NNM's Data Collector to Provide the Required SIP Data

If you chose not to automate data collections for SIP (see previous section), it is possible to manually configure NNM data collection configurations.

First you need to decide which Network Device Health gauges will be displayed within your SIP portal. Make a list of the MIB expressions used by each of those gauges. Then, make a list of the network devices that should be monitored by each gauge.

Configure data collections for each network device under the relevant SIP MIB expression:

1. Open NNM and select `Options:Data Collection & Thresholds`.
2. Configure the data collections for each network device.

   If you need more information about how to do this, select the `Help:On Window`. See also, from any NNM submap, select `Help:Online Manuals-->Managing Your Network with NNM`.

The Data Collection Process for the Network Device Health Module

HP OpenView Network Node Manager (NNM) collects all SNMP data requested by HP OpenView Service Information Portal (SIP) and provides current information about device status.

Network Device Health gauges calculate the health of specific network devices using information gathered by NNM management stations. Changes are visible in the SIP's Network Device Health gauges each time the portal view is displayed or refreshed.

SIP depends upon two programs that reside on each NNM management station (getnnmdata.exe and ovcolautoconf.exe) to collect requested data:

1. Each time a Network Device Health gauge is displayed, SIP logs the underlying data requests.

   A list of requested MIB objects and MIB expressions from any Network Device Health module gauge is compiled by SIP. The list documents which MIB objects and MIB expressions are being requested for which network devices from which NNM management stations.

NOTE The underlying MIB objects and MIB expressions appear in Network Device Health gauge definitions in the PortalView.xml file as the Component elements' href attributes. Each href attribute must have a corresponding Metric element defined in the netHealthConfig.xml file that specifies exactly which MIB object or MIB expression is being requested.

2. SIP contacts the getnnmdata.exe on each NNM management station that is configured in the mgmtStations.xml file. The frequency of this action is determined by the rawDataRefresh parameter setting in the netHealthConfig.xml file on the SIP server (by default, every 10 minutes).
3. SIP receives the most recent data collection results from the NNM database. SIP also places the current request log file in the

`ovcolautoconf` directory. Requests from each SIP server are gathered (`dc.needs<SIPserverIPaddress>`).

TIP You must create the `ovcolautoconf` directory before this step works. See “To enable the Network Device Health Module to Configure NNM Data Collection” on page 34 for more information.

4. To complete the automatic configuration process, run the `ovcolautoconf.exe` command. The `ovcolautoconf` command must be executed on the NNM management station, either manually or as a scheduled task that you define. `ovcolautoconf` does the following:

   - All SIP servers’ data collection needs are processed. The list of data collection requests is configured using the information in the `snmpRepAuto.templ` file and placed in the `snmpRepPrev.conf` file.
   - If necessary, NNM’s Data Collector configurations are updated by making SIP additions or changes to the `snmpRep.conf` file (one of two configuration files used by the NNM Data Collector program). The `snmpRep.conf` file is used by the SNMP Data Collector as a guide for gathering data. The entries from the HP OpenView Service Information Portal do not interfere with data collection configurations that were entered directly through NNM.
   - Data collections are configured on an *as-needed* basis, rather than a *potentially* needed basis. In other words, until a gauge is displayed in a portal view, no data collection is initiated.
   - If a gauge is not displayed for 30 days (default setting), the data collections are discontinued (provided they are not needed by other OpenView products).

TIP If you are worried that SIP might modify critical data collections already defined on your NNM management stations, see “Manually Configuring NNM’s Data Collector to Provide the Required SIP Data” on page 38.

Selectively Disabling SIP Data Collection Configurations

You can selectively turn off the automatic data collections configuration for any SIP MIB expression. On the SIP server, open the netHealthConfig.xml file and search for the MIB expression (Metric) that you wish to modify.

For that particular Metric, set the autoConfig="no"

Now SIP requests the information for that Metric from the databases on your NNM management station, but does not modify the data collection settings for that metric within NNM.

Monitoring the Size of NNM’s snmpCollect Database

The NNM snmpCollect database on the NNM management station grows without bounds unless you take precautions.

SIP-requested data collection data is automatically trimmed if NNM’s reporting feature is in use on your NNM management station. (Check to see if one or more NNM Performance Reports are configured on the NNM system.) Open NNM and select Help:Online Manuals-->Reporting and Data Analysis for more information.

By default, data older than one week is deleted if the NNM Reporting feature is active.

See the *ovdwtrend* reference page in NNM’s online help (or the UNIX manpage) for more information.

If Performance Reports are *not* active, the following command trims the data in the snmpCollect database. It can be run manually, or scheduled to run periodically, on the NNM system:

```
ovcoltosql -q -N -D <trim depth in hours> -exportset NNM_Reporting
```

For example, the following deletes all reporting/SIP data collector data in the snmpCollect database older than one week in age (there are 168 hours in 7 days):

```
ovcoltosql -q -N -D 168 -exportset NNM_Reporting
```

See the *ovcoltosql* reference page in NNM’s online help (or the UNIX manpage) for more information.

Removing SIP Data Collection Configurations from NNM

If you want to remove SIP data collection configuration entries from NNM's Data Collector program, at the command prompt on the NNM management station, navigate to the `ovcolautoconf` directory and type the following:

```
xnmcollect -report -delete snmpRepPrev.conf
xnmcollect -event
```

SIP Distribution Model

SIP can be configured in a tiered distribution model. For example:

- Web Browser Tier
- Web Server Tier (optional, UNIX only)
- SIP Server Tier
- Management Server Tier

For more information about the tiered distribution model, see the "Distribution Model" section of the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`).

The web browser to SIP server communication can go through a firewall and only requires HTTP or HTTPS.

The SIP server to NNM management station communication can also go through a firewall, if desired. The ports that need to be opened through the firewall to gather data for the NNM modules are specified through the SIP Configuration Editor program, `Management Station` configuration settings, on the NNM tab (see "On the SIP Server" on page 28).

Table 2-1 Alarms Module Port Requirements

Protocol	Default Port	Configuration Location
ovalarmsrv	2953 for NNM 6.1 2345 for NNM 6.2	On the SIP server, use the SIP Configuration Editor to configure ports (OVAlarmSrv Port on NNM tab for management stations).

Table 2-2 Network Device Health Module Port Requirements

Protocol	Default Port	Configuration Location
http	8880/UNIX 80/Windows	On the SIP server, use the SIP Configuration Editor to configure port (Web Server Port on NNM tab for management stations).

Table 2-3 Topology Module Port Requirements

Protocol	Default Port	Configuration Location
ovw	3700 to 3700+n	n = highest OVW session number
ovwdb	9999 for NNM 6.1 2447 for NNM 6.2	On the SIP server, use the SIP Configuration Editor to configure ports (OVwDB Port on NNM tab for management stations)
http	8880/UNIX 80/Windows	On the SIP server, use the SIP Configuration Editor to configure protocol and port (Web Server Port on the NNM tab for management stations).

The following ports may be in use for your SIP Customer Model data collection, depending upon how you configure the Management Data filters (see Chapter 6, “Segmenting the NNM Data for Your Customers,” on page 135 for more information).

Table 2-4 CustomerViews Import Program Port Requirements

Protocol	Default Port	Configuration Location
http	8880/UNIX 80/Windows	On the SIP server, server and port configurable through URL specified as customer model source in the Customer Model tab of the SIP Admin GUI: http://server:8880/OvCgi/getcvdata.exe

Table 2-5 NNM Object Database Import Program Port Requirements

Protocol	Default Port	Configuration Location
ovwdb	9999 for NNM 6.1 2447 for NNM 6.2	On the SIP server, NNM servers and ovwdb ports configurable through the file: conf/share/modules/NM/NNMData.xml

Table 2-6 NNM Data Warehouse Export Program Port Requirements

Protocol	Default Port	Configuration Location
http	8880/UNIX 80/Windows	On the SIP server, server and port configurable through the URL specified as customer model export destination in the Customer Model tab of the SIP Admin GUI: http://server:8880/OvCgi/ovsipexport.exe

Running in Languages Other Than English

Following are required configuration tasks to prepare the NNM modules to operate in non-English language mode.

For information about configuring SIP and your web browser for non-English language mode, see the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “Running SIP in Non-English Language Mode” section.

Configuring SIP to Access NNM Data

Completion of the following configuration tasks allows the Alarms module and Topology module to access non-ASCII data from NNM. (No additional steps are required for the Network Device Health module.) These instructions assume that you have completed the steps in the previous sections of this chapter.

Configuring the Alarms Module to Access Non-English NNM Data

Within each SIP alarm category definition file (*`NmAlarmCat`*`.xml`), the `NNMBaseCategory` attribute must *exactly* match NNM’s alarm category strings in the `trapd.conf` file. Therefore, if an NNM management station (from which the NNM Alarms module gathers information) is running in a language that uses non-ASCII characters, you must provide a set of *`NmAlarmCat`*`.xml` files for the localized alarm categories.

SIP requires that the non-ASCII characters be in the UTF-8 codeset. NNM, itself, uses the traditional OS codeset (for example, Shift-JIS, EUC, or ISO 88591) and does not support UTF-8 or Unicode codesets.

You can copy the NNM alarm category strings from NNM’s `trapd.conf` file and paste them into the desired *`NmAlarmCat`*`.xml` files, provided you follow these directions to ensure that the *`NmAlarmCat`*`.xml` files are saved in UTF-8 codeset.

NOTE If you are displaying alarms in SIP from NNM management stations running in multiple languages, you need to provide multiple sets of the *`NmAlarmCat`*`.xml` files. If all the NNM management stations providing information to SIP run under the same language, you only need one set of *NmAlarmCat*.xml files.

A set of Japanese *NmAlarmCat_ja*.xml files (already set up with the standard NNM alarm categories) is provided in the SIP/contrib directory. You will also find a file named NmAlarmCatsIndex_ja.xml from which you can copy and paste the new alarm categories into your NmAlarmCatsIndex.xml file. See “Relevant Files” on page 75.

1. Open the following two files in an editor that is running under the codeset that is in use on the NNM management station and is capable of converting or saving a file in UTF-8 codeset:

TIP

UNIX: vi is capable of opening a file in any codeset. After editing a file using vi (in a codeset other than UTF-8), UNIX has a command called iconv that converts most codesets into UTF-8.

Windows 2000: Notepad runs under various language settings by changing Windows 2000’s Regional Options, Locale setting. Notepad provides a Save as menu item to the UTF-8 codeset.

- On the NNM management station:
  - *Windows NT/2000:* <*NNM_install_dir*>\conf\$LANG\trapd.conf
  - *UNIX:* /etc/opt/OV/share/conf/$LANG/trapd.conf
- On the SIP server, open the NmAlarmCat.xml file you wish to modify:
  - *Windows 2000:* %SIP_HOME%\conf\share\modules\alarms\*
  - *UNIX:* /opt/OV/SIP/conf/share/modules/alarms/*

2. On the NNM management station, in the trapd.conf file, locate the CATEGORY settings, for example:

```
CATEGORY 2 "Error Alarms" "LOCALIZED-STRING-FOR-Error Alarms"
CATEGORY 3 "Threshold Alarms" "LOCALIZED-STRING-FOR-Threshold Alarms"
CATEGORY 4 "Status Alarms" "LOCALIZED-STRING-FOR-Status Alarms"
CATEGORY 5 "Configuration Alarms" "LOCALIZED-STRING-FOR-Configuration Alarms"
CATEGORY 6 "Application Alert Alarms" "LOCALIZED-STRING-FOR-Appl Alert Alarms"
```

3. Copy the string from the second set of quotes, and on the SIP server, paste it into the `NNMBaseCategory` attribute in the *`NmAlarmCat`*`.xml` file. For example:

```
<?xml version='1.0' encoding='utf-8' standalone='no' ?>
<!DOCTYPE AlarmCategoryDef SYSTEM "NmAlarmCat.dtd">

<AlarmCategoryDef DisplayTitle="Error Alarms"
                  NNMBaseCategory="LOCALIZED-STRING-FOR-Error Alarms">
    <Severities critical="1"
                major="1"
                minor="1"
                warning="1"
                normal="1"/>
    <Acknowledgement acknowledged="1"
                     unacknowledged="1"/>
</AlarmCategoryDef>
```

4. If you want the SIP alarm category title to be localized as well, modify the `DisplayTitle` string in the same manner.

5. Save or convert the *`NmAlarmCat`*`.xml` file in the UTF-8 codeset. Place it in the same directory as the other *`NmAlarmCat`*`.xml` files.

6. Open the `NmAlarmCatsIndex.xml` file and add your new *`NmAlarmCat`*`.xml` file name to the list.

7. After making changes in the `../conf/share/modules/alarms` directory, you must stop and restart the SIP servlet engine before changes take effect. See “Restarting the Servlet Engine” on page 220.

8. You must now open any *`PortalView`*`.xml` files and the default Alarms module (`OVDefaultAlarms.xml`) file and edit the Alarms module instances to point to the new localized alarm category names. See “Editing the NNM Alarms Module” on page 60.

Configuring the Topology Module to Access Non-English NNM Data

If an NNM management station (from which SIP gathers information for the NNM Topology Map module) is running in a language that uses non-ASCII characters, the `encoding` attribute in the SIP Configuration

Editor must specify which codeset the JDK Converter should use. If no codeset is configured, no conversion is done.

After the `encoding` attribute is set, NNM map data is automatically translated into UTF-8 characters by the portal.

The Java JDK Converter 1.1 can be configured on a per-NNM-management-station basis to determine how to interpret the incoming codeset.

1. On the NNM management station, open the following file:

   Windows NT/2000: `<`*`NNM_install_dir`*`>\www\conf\locales.jconv`

   UNIX: `/etc/opt/OV/share/www/conf/locales.jconv`

   This file contains a table of JDK Converter 1.1 values for all languages.

   Locate the appropriate value for your NNM management station. Find the OS locale in which NNM is running (for example on Windows NT, "Japanese_Japan.932") and the corresponding JDK 1.1 Converter (for example, "SJIS"). In this case, in the next step, you would enter:

   `SJIS`

2. On the SIP server, open the SIP Configuration Editor. Select your NNM management station and navigate to the `NNM` tab. Enter the appropriate `encoding` attribute for your NNM management station.

3. To make the localized submap information visible in your portal views, make sure that the submap name uses the appropriate UTF-8 characters:

   - In the default `OVDefaultTopology.xml` file, update the submap name (such as “Internet”) with the localized string as it appears on the NNM management station.
   - You *must* use UTF-8 codeset characters when entering the string for the submap name.

NOTE A Japanese language version of the default Topology module file is provided for your convenience (see `OVDefaultTopology_ja.xml` “Relevant Files” on page 131).

- In any existing Topology module's in your *PortalView*.xml files: Open the Topology - Edit window. The localized submap names appear in the selection list box. To convert the submaps to the localized names, simply reinsert the submap ("Editing the Topology Module" on page 122).

Secure Socket Layer (SSL) Support

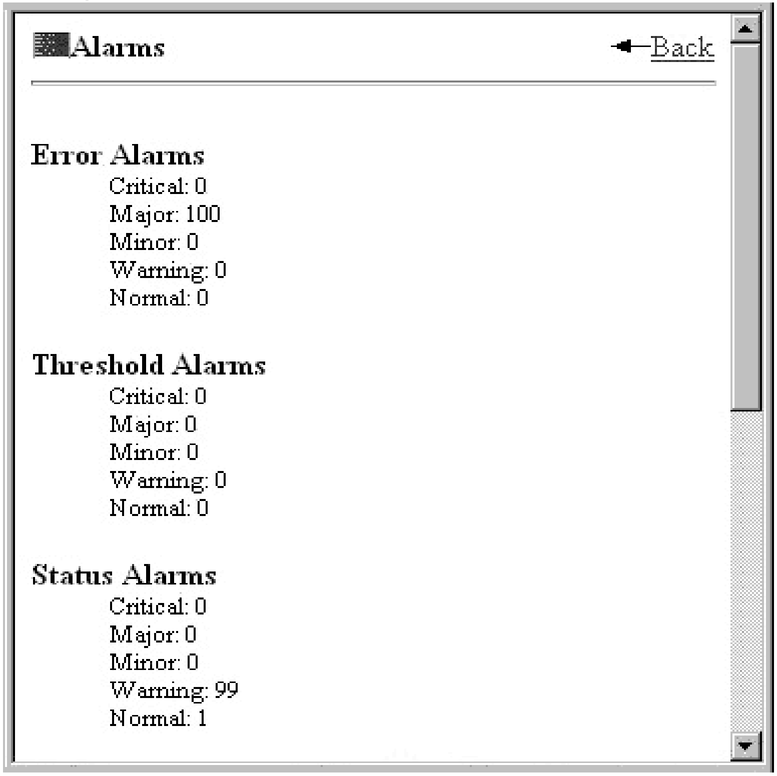
The SIP server to NNM management station communication cannot be configured to use Secure Socket Layer (SSL) at this time.

Running the NNM modules in a Wireless Environment

If you configure SIP for portals that are accessed through wireless devices, two of the NNM modules are available and displayed as shown below. See the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`) for general information about setting up SIP to run on wireless devices.

Less information is displayed in wireless environments than in the full HTML web version of these modules. No links to details are available with the PDA (hand-held device) or WML (cell phone) display types.

Figure 2-1 PDA and WML Format for the Alarms Module

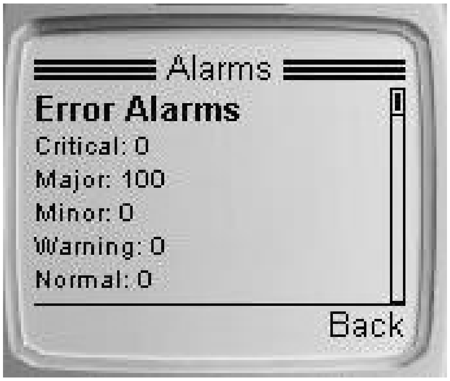


Alarms
◄—Back

Error Alarms
Critical: 0
Major: 100
Minor: 0
Warning: 0
Normal: 0

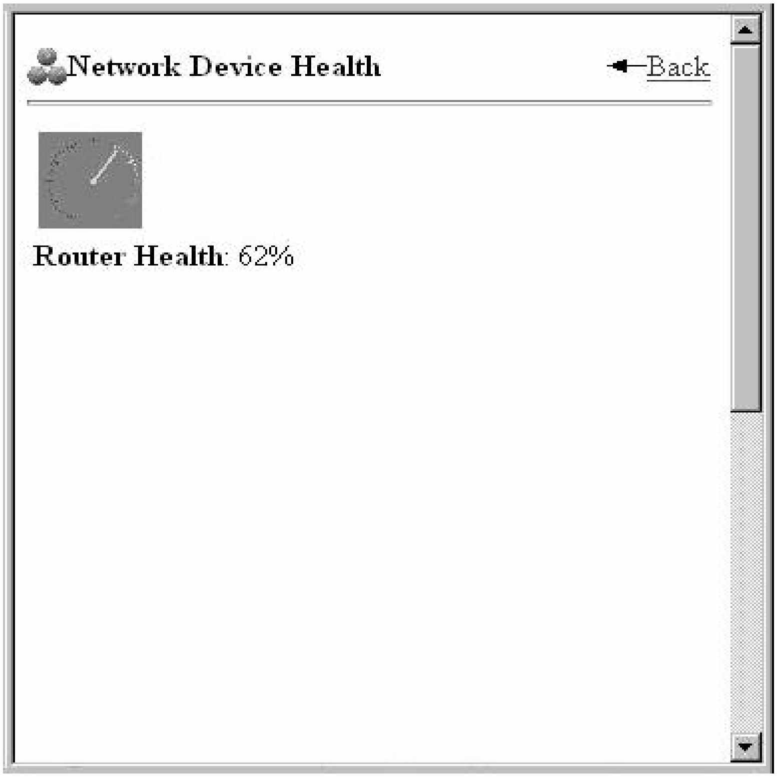
Threshold Alarms
Critical: 0
Major: 0
Minor: 0
Warning: 0
Normal: 0

Status Alarms
Critical: 0
Major: 0
Minor: 0
Warning: 99
Normal: 1

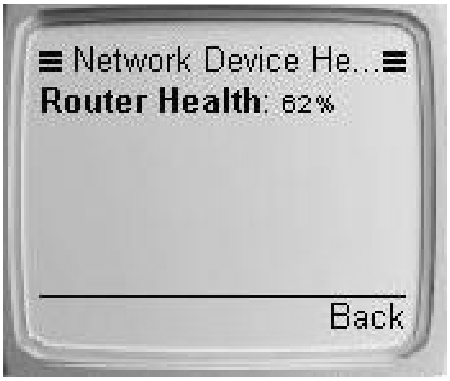


Alarms
Error Alarms
Critical: 0
Major: 100
Minor: 0
Warning: 0
Normal: 0
Back

Figure 2-2 PDA and WML Format for the Network Device Health Gauges Module



Router Health: 62%



Router Health: 62%

Using the NNM SSO Authentication Provider for SIP

If you use the NNM SSO Authentication Provider, no user name and password are required as long as the user has already been authenticated by the NNM Session Manager web login mechanism. See the “Configuring Authentication in SIP” section of the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`) for more information.

NOTE Be aware that using the NNM SSO Authentication Provider requires that NNM is running on the same host as SIP.

3 NNM Alarms Module

Understanding NNM Alarms Data

The Alarms module presents network alarms from HP OpenView Network Node Manager (NNM) running on one or more NNM management stations within your management domain.

Changes are visible in the NNM Alarms module each time the portal view is displayed or refreshed. The alarm lists are continually updated in SIP memory.

Adding an Alarms Module to a Portal View

To insert the Alarms module into a portal view:

1. Access the portal view by logging on to SIP as a user with access to the appropriate role. If this user has access to multiple roles, switch to the appropriate role.

   Your currently assigned SIP role must have `ViewAdmin` editing permissions.

2. Navigate to the appropriate tab.
3. At the bottom of any wide column, either:
   - Select `Alarms` from the `Select Module to Add` list box, and click `[Add]`, or
   - Click `[Edit]` to access the `Edit Column` page. Insert the Alarms module and place it into the desired location among other modules in the column. Click `[OK]` to save the changes and return to the main portal page.

A copy of the default Alarms module is inserted into your *`PortalView`*`.xml` file.

- If you want to modify this module instance, turn to “Editing the NNM Alarms Module” on page 60.
- If you want to change the default module, see “Relevant Files” on page 75.

TIP If you want to add a module to the list of available modules, see "Relevant Files" on page 75. You can create and add another instance of any module.

Editing the NNM Alarms Module

Using the Alarms - Edit Page

You can easily modify the Alarms module in your portal view:

1. Access the portal view by logging on to SIP as a user with access to the appropriate role. If this user has access to multiple roles, switch to the appropriate role.

   Your currently assigned SIP role must have `ViewAdmin` editing permissions.

2. Navigate to the appropriate tab.

3. In the title bar of the Alarms module, click the edit button:

4. Select `Choose from List`. Make any desired changes. Click the `[Help]` button if you need more information:

   - To display an alarm category, select an alarm category in the `Available Alarm Categories` list, and click the `[Add]` button. You can use `Shift-Click` and `Ctrl-Click` to make multiple selections.

   - To remove an alarm category from view, select an alarm category in the `Displayed Alarm Categories` list, and click the `[Remove]` button. You can use `Shift-Click` and `Ctrl-Click` to make multiple selections.

   - To adjust the order in which the alarm categories are displayed, select an alarm category in the `Displayed Alarm Categories` list and click the `[Up]` or `[Down]` button.

5. To save the changes and return to the main portal page, click `[OK]`.

6. Log out of the SIP portal.

7. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Directly Editing the PortalView.XML File

TIP For the following adjustments, you must edit the XML file. It is recommended that you use the `Alarms - Edit` page for all other editing changes.

- Change the displayed title for this module instance.
- Add your own online help to the `[?]` button for this module.

To directly modify the XML code for a Alarms module:

1. Make a backup of XML files before you make changes. If you edit the XML file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.
2. Open your *`PortalView`*`.xml` file with an ASCII or XML editor. Portal view files are stored in the following directory or subdirectories below this one:

   Windows 2000: `%SIP_HOME%\conf\share\views`

   UNIX: `/opt/OV/SIP/conf/share/views`

   If a portal view file does not yet exist see the “Customizing Portal Views” section of the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`) and follow the procedure for creating a portal view.
3. Search for the following string to find your existing module to edit:

   `classid="com.hp.ov.portal.modules.alarms"`

   Module instances are wrapped in the `ModuleInstance` element. The `ModuleInstance id` must be unique among all module instances in the portal view file. For information about the `ModuleInstance` element, see the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “PortalView DTD” section.

   For example:

```
<ModuleInstance
   classid="com.hp.ov.portal.modules.alarms"
   display="yes"
   help="/OvSipDocs/C/help//NNM/alarmsView.html"
   id="module3"
   rollupState="down"
   title="Alarms">
</ModuleInstance>
```

4. Copy the contents from your `NmAlarmCatsIndex.xml` file, and paste it between the `ModuleInstance` starting and closing tags. Your `NmAlarmCatsIndex.xml` file contains a list of all currently valid alarm categories:

   - *Windows 2000:*

     `%SIP_HOME%\conf\share\modules\alarms\NmAlarmsCatIndex.xml`

   - *UNIX:*

     `/opt/OV/SIP/conf/share/modules/alarms/NmAlarmsCatIndex.xml`

   The text in the `NmAlarmCatsIndex.xml` file looks something like:

   ```
   <AlarmDisplay>
      <CategoryDefName href="AllAlarms.xml"/>
      <CategoryDefName href="ApplicationAlertAlarms.xml"/>
      <CategoryDefName href="ConfigurationAlarms.xml"/>
      <CategoryDefName href="ErrorAlarms.xml"/>
      <CategoryDefName href="StatusAlarms.xml"/>
      <CategoryDefName href="ThresholdAlarms.xml"/>
   </AlarmDisplay>
   ```

   See the comments in the `OVAlarms.dtd` file for more information about the correct XML syntax:

   - *Windows 2000:* `%SIP_HOME%\conf\share\views\OVAlarms.dtd`

   - *UNIX:* `/opt/OV/SIP/conf/share/views/OVAlarms.dtd`

NOTE To make changes to an alarm category configuration or add a custom alarm category, see “Editing Alarm Categories” on page 67.

5. To change the title of this Network Device Health module instance, change the `title` attribute:
   `<ModuleInstance title="new title">`

   To change the title of all Alarms modules, change the `title` attribute in the registration file, see “Relevant Files” on page 75.

6. To launch your own help topic from the module’s `[?]` button, insert the `help` attribute into the `<ModuleInstance>`:

   `help="/OVSipDocs/C/help/NNM/`*`topic`*`.html"`

   Replace *`topic`*`.html` with the name of your help file. The `help` attribute allows you to override the default help URL defined in the module registration file. See the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “Adding and Customizing Module Help Topics” for more information about writing your own online help.

7. To remove an alarm category from this module, delete the appropriate `CategoryDefName` line.

8. To change the order in which alarm categories are displayed, reorder the `CategoryDefName` lines into the desired order.

9. Save the XML file.

10. After you make modifications to XML files, validate the syntax. See Appendix , “Validating XML Files,” on page 225 for more information.

11. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Establishing Global Settings for Alarms Modules

The settings in the `NmAlarmConfig.xml` file affect all Alarm modules in all SIP portals. To modify the global settings, locate the file:

- *Windows 2000:*

  `%SIP_HOME%\conf\share\modules\alarms\NmAlarmConfig.xml`

- *UNIX:*

  `/opt/OV/SIP/conf/share/modules/alarms/NmAlarmConfig.xml`

1. Make a backup of XML files before you make changes. If you edit the XML file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.
2. Open `NmAlarmConfig.xml` with an ASCII or XML editor.
3. The attributes defined in this file affect all alarm categories in every instance of an Alarms module in all `PortalView.xml` files. Make any desired changes:

   - `maxConnections`
     The maximum number of (socket) connections that a SIP server is allowed to establish with all the NNM management stations it needs to communicate with, for gathering alarm information.

     A connection and a thread is established between the SIP server and each NNM station for each active alarm-category/role pair. Multiple portal users viewing alarms that originate from the same NNM management station share a connection as long as they all are assigned to the same role definition. When the specified maximum is reached, the least used connection is closed and a new one is opened, as needed.

     For example, an NNM Alarms module that has 6 alarm categories and gathers alarms from 6 NNM management stations would require 36 socket connections per role. If there are 2 roles, 78 socket connections are required. Multiple users can share the same role, and therefore share socket connections. When the specified number is reached, for each new subsequent connection required: (1) the least used connection is closed and (2) a new one is opened.

- `showSummaryLine`
  yes means a message displays at the bottom of each alarm category explaining current configuration settings ("configured for x alarms, received y.")
  no means no such message is displayed.
- `shortDateFormat`
  yes means the current locale setting's short date format is used. For example, US English: `mm/dd/yy hh:mm:ss am/pm`
  no means the current locale's long date format is used. For example, US English: `Tuesday March 20 2001 hh:mm:ss am/pm tz`
- `connTimeOut` (zero or greater)
  The number of seconds to pause after each socket connection is opened. The smaller the number, the faster the Alarms module opens when a *`PortalView`*`.xml` is first accessed. However, the `connTimeout` value may be so short that no alarms are displayed until the portal is refreshed. Too short a `connTimeOut` value may cause the `"Data currently unavailable"` error message to display, rather than the alarm text.
- `addSyncTime` (zero or greater)
  The number of seconds to add to `connTimeout` when making a synchronous call to get data from the `ovalarmsrv` on each NNM management station. Synchronous calls are required when the you set the `OlderThanXMinutes` attribute to a non-zero value in any *`NmAlarmCat`*`.xml` file. Once an `OlderThanXMinutes` attribute is specified, alarm data cannot be cached because the time value in the filter request changes with every refresh.
- `socketTimeout` (zero or greater)
  The number of seconds to wait for a socket connection to be made.
- `replyTimeout` (zero or greater)
  The number of seconds to wait each time for any response (protocol or data) from `ovalarmsrv`.
- `maxWaitTime` (zero or greater)
  The maximum number of seconds to wait for a data response from `ovalarmsrv`. The value for this attribute should be greater than the value for the `responseTimeout` attribute to allow for delays due to network traffic.

- `catCfgInterval` (zero or greater)
  The number of seconds to wait between checking for changes to the alarms index file or any of the alarm category definition files.

  See the `NmAlarmConfig.dtd` and `NmAlarmConfig.xml` for more information.

4. If you make changes to this file, save the XML file.
5. After you make modifications to this XML file, you must restart the Tomcat engine. See Appendix A, “Restarting Tomcat,” on page 219 for more information.
6. After you make modifications to this XML file, validate the syntax. See “Validating XML Files” on page 225 for more information.

Editing Alarm Categories

SIP alarm categories are based upon existing NNM alarm categories, such as “Status Alarms” or “Threshold Alarms.” SIP alarm categories must be defined in an `*NmAlarmCat*.xml` file and be listed in the `*NmAlarmCatsIndex*.xml` file before they can be displayed in SIP portal views through Alarms modules. Although they are based upon a specific NNM alarm category, the SIP alarm category name can be different from the base NNM alarm category name; such as “Accounting Department’s Network Problems” or “Internet Availability Alarms.”

You can modify SIP alarm categories in a variety of ways:

“Modifying Existing Alarm Categories” on page 67

“Creating New Alarm Categories” on page 69

“Eliminating an Alarm Category” on page 73

Modifying Existing Alarm Categories

To modify a SIP Alarm Category:

1. Make a backup of XML configuration files before you customize them. If you edit the file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.
2. Before you start, access the following two files as a reference:
   - *Windows 2000:*
     `%SIP_HOME%\conf\share\modules\alarms\`
     `SampleNmAlarmCat.xml` and
     `NmAlarmCat.dtd`
   - *UNIX:*
     `/opt/OV/SIP/conf/share/modules/alarms/`
     `SampleNmAlarmCat.xml` and
     `NmAlarmCat.dtd`
3. In an ASCII or XML editor, open the `*NmAlarmCat*.xml` file that defines the alarm category you wish to modify. These definition files must be stored in the following location:

- *Windows 2000:* `%SIP_HOME%\conf\share\modules\alarms\`
- *UNIX:* `/opt/OV/SIP/conf/share/modules/alarms/`

4. Following the rules as explained in the `SampleNmAlarmCat.xml` and `NmAlarmCat.dtd` files, make the desired modifications.

   For example:

```
<?xml version='1.0' encoding='utf-8' standalone='no' ?>
<!DOCTYPE AlarmCategoryDef SYSTEM "NmAlarmCat.dtd">

<AlarmCategoryDef
     DisplayTitle="Sample Alarms"
     NNMBaseCategory="Sample Alarms"
     NumAlarms="15"
     MatchDescSubstring="string text"
     OlderThanXMinutes="10">
  <Severities critical="1"
              major="1"
              minor="1"
              warning="1"
              normal="1"/>
  <Acknowledgement
              acknowledged="1"
              unacknowledged="1"/>
  <NNMStationList>
        <NNMStation hostname="NNM-management-station"/>
  </NNMStationList>
  <NodeSelection>
        <OrganizationFilter>
            <OrganizationRef href="a-reference-to-a-defined-organization"/>
        </OrganizationFilter>
        <IPHostFilter>
            <IPHost hostname="a-machine-from-which-to-collect-data"/>
        </IPHostFilter>
        <CapabilityFilter>
            <Capability field="NNM-database(ovwdb)-capability-field-value"/>
        </CapabilityFilter>
  </NodeSelection>
</AlarmCategoryDef>
```

   For information about `<NodeSelection>` filtering specifications, see “Filtering Possibilities for the Alarms Module” on page 167.

CAUTION If you are gathering alarms from an NNM management station running in a language other than English, see "Configuring the Alarms Module to Access Non-English NNM Data" on page 47 for important information about localization of the Alarms module.

5. Close and save the *`NmAlarmCat`*`.xml` file.
6. After you make modifications to this XML file, you must restart the Tomcat engine. See Appendix A, "Restarting Tomcat," on page 219 for more information.
7. After you make modifications to this XML file, validate the syntax. See "Validating XML Files" on page 225 for more information.
8. In a browser, log into the portal to verify that the alarms appear as desired.

You may wish to review the current global alarm category settings (see "Establishing Global Settings for Alarms Modules" on page 64). One of the global choices specifies whether or not a message is displayed, at the bottom of each alarm category, explaining the number of alarms currently allowed to be displayed.

Creating New Alarm Categories

Each SIP alarm category is defined in a separate XML file.

When configuring a new alarm category, first decide which NNM Alarms Category you want to use in a given portal view and which NNM management stations to collect alarms from. SIP ships with predefined alarm category files for the standard NNM alarm categories. You can copy any of these to use as a starting point.

To create a new SIP alarm category:

1. Make a backup of XML files before you make changes. If you edit the XML file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.
2. Before you start, access the following two files as a reference:

  - *Windows 2000:*
    `%SIP_HOME%\conf\share\modules\alarms\SampleNmAlarmCat.xml` and `NmAlarmCat.dtd`
  - *UNIX:*
    `/opt/OV/SIP/conf/share/modules/alarms/SampleNmAlarmCat.xml` and `NmAlarmCat.dtd`

3. Copy and rename one of the following *`NmAlarmCat`*`.xml` files as a starting point for your new alarm category definition file:
   - `SampleNmAlarmCat.xml`
   - `ApplicationAlertAlarms.xml`
   - `ConfigurationAlarms.xml`
   - `ErrorAlarms.xml`
   - `StatusAlarms.xml`
   - `ThresholdAlarms.xml`
4. In an ASCII or XML editor, open your new *`NmAlarmCat`*`.xml` file. This file must be stored in the following location:
   - *Windows 2000:*
     `%SIP_HOME%\conf\share\modules\alarms\`
   - *UNIX:*
     `/opt/OV/SIP/conf/share/modules/alarms/`
5. Following the rules as explained in the `SampleNmAlarmCat.xml` and `NmAlarmCat.dtd` files, make the desired modifications.

   For example:

```
<?xml version='1.0' encoding='utf-8' standalone='no' ?>
<!DOCTYPE AlarmCategoryDef SYSTEM "NmAlarmCat.dtd">

<AlarmCategoryDef
     DisplayTitle="Sample Alarms"
     NNMBaseCategory="Sample Alarms"
     NumAlarms="15"
     MatchDescSubstring="string text"
     OlderThanXMinutes="10">
  <Severities critical="1"
              major="1"
              minor="1"
              warning="1"
              normal="1"/>
  <Acknowledgement
              acknowledged="1"
              unacknowledged="1"/>
  <NNMStationList>
        <NNMStation hostname="NNM-management-station"/>
  </NNMStationList>
  <NodeSelection>
        <OrganizationFilter>
            <OrganizationRef href="a-reference-to-a-defined-organization"/>
        </OrganizationFilter>
        <IPHostFilter>
            <IPHost hostname="a-machine-from-which-to-collect-data"/>
        </IPHostFilter>
        <CapabilityFilter>
            <Capability field="NNM-database(ovwdb)-capability-field-value"/>
        </CapabilityFilter>
  </NodeSelection>
</AlarmCategoryDef>
```

For information about `<NodeSelection>` filtering specifications, see "Filtering Possibilities for the Alarms Module" on page 167.

CAUTION If you are gathering alarms from an NNM management station running in a language other than English, see "Configuring the Alarms Module to Access Non-English NNM Data" on page 47 for important information about localization of the Alarms module.

6. Close and save the *NmAlarmCat*.`xml` file.

7. In an ASCII or XML editor, open the `NmAlarmsCatIndex.xml` file that contains the list of all valid alarm categories available for display in your portal view. This file must be stored in the following location:

   - *Windows 2000:*

     `%SIP_HOME%\conf\share\modules\alarms\NmAlarmsCatIndex.xml`

   - *UNIX:*

     `/opt/OV/SIP/conf/share/modules/alarms/NmAlarmsCatIndex.xml`

8. The `NmAlarmsCatIndex.xml` file specifies the list of SIP alarm category definitions that are available through for use in the Alarms module. Any alarm categories requested within Alarm module instances, but not listed in this file, are ignored. Enter a new CategoryDefName for your new *`NmAlarmCat`*`.xml` file. For example:

```
<AlarmDisplay>
   <CategoryDefName href="NmAlarmCat.xml"/>
   <CategoryDefName href="AllAlarms.xml"/>
   <CategoryDefName href="ApplicationAlertAlarms.xml"/>
   <CategoryDefName href="ConfigurationAlarms.xml"/>
   <CategoryDefName href="ErrorAlarms.xml"/>
   <CategoryDefName href="StatusAlarms.xml"/>
   <CategoryDefName href="ThresholdAlarms.xml"/>
</AlarmDisplay>
```

9. Close and save the `NmAlarmsCatIndex.xml` file.

10. After you make modifications to this XML file, you must restart the Tomcat engine. See Appendix A, “Restarting Tomcat,” on page 219 for more information.

11. After you make modifications to this XML file, validate the syntax. See “Validating XML Files” on page 225 for more information.

12. In a browser, log into the portal to verify that the alarms appear as desired.

You may wish to review the current global alarm category settings (see “Establishing Global Settings for Alarms Modules” on page 64). One of the global choices specifies whether or not a message is displayed, at the bottom of each alarm category, explaining the number of alarms currently allowed to be displayed.

Eliminating an Alarm Category

1. In an ASCII or XML editor, open the `*NmAlarmsCatsIndex*.xml` file:

   - *Windows 2000:*
     `%SIP_HOME%\conf\share\modules\alarms\`
   - *UNIX:*
     `/opt/OV/SIP/conf/share/modules/alarms/`

2. The `NmAlarmsCatIndex.xml` file specifies the list of SIP alarm category definitions that are available through for use in the Alarms module. Any alarm categories requested within Alarm module instances, but not listed in this file, are ignored. Delete the `CategoryDefName` line referring to the SIP alarm category that you wish to eliminate:

   ```
   <AlarmDisplay>
      <CategoryDefName href="NmAlarmCat.xml"/>
      <CategoryDefName href="AllAlarms.xml"/>
      <CategoryDefName href="ApplicationAlertAlarms.xml"/>
      <CategoryDefName href="ConfigurationAlarms.xml"/>
      <CategoryDefName href="ErrorAlarms.xml"/>
      <CategoryDefName href="StatusAlarms.xml"/>
      <CategoryDefName href="ThresholdAlarms.xml"/>
   </AlarmDisplay>
   ```

3. Close and save the `NmAlarmsCatIndex.xml` file.

4. After you make modifications to this XML file, you must restart the Tomcat engine. See Appendix A, “Restarting Tomcat,” on page 219 for more information.

5. After you make modifications to this XML file, validate the syntax. See “Validating XML Files” on page 225 for more information.

6. In a browser, log into the portal to verify that the alarms appear as desired.

This alarm category is no longer allowed to display in SIP portal views, even if it is specifically listed in an Alarms module within a `PortalView.xml` file.

If you want to totally eliminate any trace of this SIP alarm category:

1. In an ASCII or XML editor, open the following files and delete all instances of the `CategoryDefName` line referring to the obsolete SIP alarm category and save the modified version of the files:

- *Windows 2000:*

```
%SIP_HOME%\conf\share\views\*.xml
%SIP_HOME%\registration\defaults\OVDefaultAlarms.xml
```

- *UNIX:*

```
/opt/OV/SIP/conf/share/views/*.xml
/opt/OV/SIP/registration/defaults/OVDevaultAlarms.xml
```

2. Delete the `*NmAlarmCat*.xml` file that defines the obsolete alarm category. These definition files must be stored in the following location:

    - *Windows 2000:* `%SIP_HOME%\conf\share\modules\alarms\`
    - *UNIX:* `/opt/OV/SIP/conf/share/modules/alarms/`

3. After you make modifications to this XML file, you must restart the Tomcat engine. See Appendix A, “Restarting Tomcat,” on page 219 for more information.

Relevant Files

The Alarms module must follow the rules defined in the following DTD files. See the comments in the DTD files for an explanation of each element used in the XML files:

- `mgmtStations.dtd & nmConfig.dtd & mgmtStations.xml`

  This XML file contains the list of all NNM management stations with which SIP is allowed to communicate. Use the SIP Configuration Editor program to make changes to this file. You must specify whether or not the Alarms module is allowed to request data. You must provide information about which ports are being used by NNM processes to communicate alarm data on each management station. See “Establishing Communication Between NNM and SIP” on page 28.

- `OVModuleRegistraton.dtd & OVRegAlarms.xml`

  This XML file grants access to the Alarms module through the SIP framework so that it is available for your use. To add another instance of the Alarms module to the SIP module selection list, you copy and rename the `OVRegAlarms.xml` and the `OVDefaultAlarms.xml` files. Then update the `description`, `title`, `classid`, `help`, and `defaultConfigXML` attribute values in the new registration file.

  If you make any changes to a registration file, you must follow the directions in “Restarting the Servlet Engine” on page 220.

- `OVAlarms.dtd & OVDefaultAlarms.xml`

  This DTD defines the rules for configuring any of the Alarms module. The XML file contains the *default* Alarms module. The contents of the default file is inserted into your portal each time you use the `[Add]` button to insert the Alarms module.

  You can modify the `OVDefaultAlarms.xml` file to meet your needs. Either:

  — Directly edit the XML code in the `OVDefaultAlarms.xml` file, or

  — Insert an Alarms module into any portal. Modify the module to meet your needs. Then, copy the modified XML code for the module from your portal view file, and paste it into the `OVDefaultAlarms.xml` file.

See “Directly Editing the PortalView.XML File” on page 61 for more information

- `OVAlarms.dtd & NmAlarmsCatIndex.xml & NmAlarmsCatIndex_ja.xml` (Japanese version)

  This XML file specifies the list of SIP alarm category definitions that are available through for use in the Alarms module. Any alarm categories requested within Alarm module instances, but not listed in this file, are ignored. (See the comments in the `OVAlarms.dtd` and `SampleNmAlarmCat.xml` file for more information.) A Japanese version is provided, see Table 3-1.

- `NmAlarmConfig.dtd & NmAlarmConfig.xml`

  This XML file contains the global setting used by all Alarms modules. See “Establishing Global Settings for Alarms Modules” on page 64 for more information.

- `nmAlarmCat.dtd & SampleNmAlarmCat.xml &` *`nmAlarmCat`*`.xml &` *`nmAlarmCat`*`_ja.xml` (Japanese versions)

  There must be one *`nmAlarmCat`*`.xml` file for each NNM alarm category that you wish to display in SIP portals.

  Two sets of configuration files for the standard NNM alarm categories are provided: one set for use with NNM management stations running in English, the other set for use with NNM management stations running in Japanese (see “Running in Languages Other Than English” on page 47 for more information, and see Table 3-1).

  (See the comments in the `nmAlarmCat.dtd` and `SampleNmAlarmCat.xml` file for more information.)

- `PortalView.dtd &` *`PortalView`*`.xml`

  This DTD provides the rules for formatting the XML code in your portal view files. See the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “Customizing Portal Views” section for more information about creating portal view files.

- `/htdocs/C/help/NNM/*.html`

  This directory contains the Help topics for Alarms modules, accessed by clicking the `[?]` button. If you want to supply your own customized help files, see the *SIP Deployment and Integration Guide*

(`SIP_Deployment_Integration.pdf`), “Adding and Customizing Module Help Topics” section.

Table 3-1 Alarms Module Files on the SIP Server

File Name	**Windows 2000 Location** **%SIP_HOME%\....**	**UNIX Location** **/opt/OV/SIP/....**
mgmtStations.dtd	conf\share\stations\	conf/share/stations/
nmConfig.dtd	conf\share\stations\	conf/share/stations/
mgmtStations.xml	conf\share\stations\	conf/share/stations/
OVModuleRegistration.dtd	registration\	registration/
OVRegAlarms.xml	registration\	registration/
NmAlarmCat.dtd	conf\share\modules\alarms\	conf/share/modules/alarms/
SampleNmAlarmCat.dtd	conf\share\modules\alarms\	conf/share/modules/alarms/
NmAlarmCat.xml	conf\share\modules\alarms\	conf/share/modules/alarms/
AllAlarms.xml	conf\share\modules\alarms\	conf/share/modules/alarms/
ApplicationAlertAlarms.xml	conf\share\modules\alarms\	conf/share/modules/alarms/
ConfigurationAlarms.xml	conf\share\modules\alarms\	conf/share/modules/alarms/
ErrorAlarms.xml	conf\share\modules\alarms\	conf/share/modules/alarms/
StatusAlarms.xml	conf\share\modules\alarms\	conf/share/modules/alarms/
ThresholdAlarms.xml	conf\share\modules\alarms\	conf/share/modules/alarms/
AllAlarms_ja.xml	contrib\conf\alarms\	contrib/conf/alarms/
ApplicationAlertAlarms_ja.xml	contrib\conf\alarms\	contrib/conf/alarms/
ConfigurationAlarms_ja.xml	contrib\conf\alarms\	contrib/conf/alarms/
ErrorAlarms_ja.xml	contrib\conf\alarms\	contrib/conf/alarms/
StatusAlarms_ja.xml	contrib\conf\alarms\	contrib/conf/alarms/
ThresholdAlarms_ja.xml	contrib\conf\alarms\	contrib/conf/alarms/

Table 3-1 Alarms Module Files on the SIP Server

File Name	Windows 2000 Location %SIP_HOME%\....	UNIX Location /opt/OV/SIP/....
NmAlarmConfig.dtd	conf\share\modules\alarms\	conf/share/modules/alarms/
NmAlarmConfig.xml	conf\share\modules\alarms\	conf/share/modules/alarms/
OVAlarms.dtd	conf\share\views\	conf/share/views/
OVDefaultAlarms.xml	registration\defaults\	registration/defaults/
OVDefaultAlarms_ja.xml	contrib\conf\alarms\	contrib/conf/alarms/
NmAlarmCatsIndex.xml	conf\share\modules\alarms\	conf/share/modules/alarms/
NmAlarmCatsIndex_ja.xml	contrib\conf\alarms\	contrib/conf/alarms/
PortalView.xml	conf\share\views\	conf/share/views
*.html	htdocs\C\help\NNM\	htdocs/C/help/NNM/

4 Network Device Health Module

Understanding Network Device Health Gauges

Network health is scored as a value from 0-100, with 0 being the poorest health and 100 being the best health.

Two types of information are available for Network Device Health:

- **Gauges** are displayed when the tab first displays in your portal, and indicate the overall health rating for all devices being monitored by the particular gauge.
- **Detail pages** show the details that were included in the health calculation. The detail pages are displayed by clicking a gauge or the gauge's title.

The Gauges

You can customize the Network Device Health module to display specific gauges. You can modify the predefined gauges or write your own to monitor any aspect of the network that is of concern to your customers.

A network health gauge represents the mean health of all network devices being monitored by a particular gauge. For example, `Router Health` represents the mean health of all routers. If you have two routers, one with a health score of 100% and one with a score of 60%, Router Health gauge points to 80%.

The following Network Device Health gauges are preconfigured and provided with SIP:

- **Router Health**
  This gauge monitors the health of every device in the NNM database that has the `isRouter` capability setting.
- **Server Health**
  This gauge monitors the health of every device in the NNM database that has the `isServer` capability setting.
- **Key Device Health**
  This gauge monitors the health of every device in the NNM database that has the `isKeyDevice` capability setting. If HP OpenView Customer Views is installed and configured on your NNM

management station, you probably have devices with the `isKeyDevice` capability. Within Customer Views, see the online help or the web-based *Concepts Guide* for more information about key devices.

- **CPE Health**
  This gauge monitors the health of every device in the NNM database that has the `isCPE` (customer premises equipment) capability setting. If HP OpenView Customer Views is installed and configured on your NNM management station, you probably have devices with the `isCPE` capability. Within Customer Views, see the online help or the web-based *Concepts Guide* for more information about CPE devices.

- **Interface Health**
  This gauge monitors the health of every interface in the NNM database that passes the interface filter assigned to the current SIP role. This gauge cannot operate unless a specific list of interfaces is established through at least one of the filtering levels allowed within SIP configuration. This limitation is imposed so that you don't accidently set up data collections on every interface in the whole NNM management domain. See Chapter 6, "Segmenting the NNM Data for Your Customers," on page 135 and "Filtering Possibilities for the Network Device Health Module" on page 176.

You can also write your own Network Device Health gauges to monitor SNMP MIBs or MIB expressions and calculate whatever you want. See "Creating Your Own Network Device Health Gauge" on page 96.

Details View

Health gauges may have two levels of health detail drill-down. If available:

- To view the first level health details, click on the gauge or the health gauge title above the gauge.
- To view the second level of health detail, click the health score values that are links in the first level health detail. Only certain values provide links.

The first column of a detail table—**Resource**—displays the name of the network resource (for example, the name of the Interface, Router, Server, Key Device, or Customer Premises Equipment).

The second column—**Overall Health**—contains the resource’s health score. This score is based upon the weighted mean of a set of metrics measured on that resource.

The remaining columns display the health score for each metric used to compute network resource health. The score is a value from 0-100 derived from analysis of the metric value. You may also see the raw data value columns. You choose whether or not to present raw data in your portal views. By default, raw data is not presented.

The tables below describe the metrics used by the default gauges. The metrics identify the specific SNMP MIBs and MIB expressions for NNM to monitor on each device (see “Prerequisites to Creating Your Own Network Device Health Gauge” on page 97 for more information about MIBs and MIB expressions).

NOTE For interface metrics, to obtain the most accurate reading, NNM uses a variety of formulas depending upon the attributes of the interface (such as speed of the interface: half-duplex versus full-duplex).

Table 4-1 Interface Health Metrics (used by all gauges)

Statistic **(MIB expression used)**	**Default Settings Description**
Up/Down Status (ping response)	An indication of whether the interface is up or down. An interface that is up has a status health score of 100%. An interface that is down has a status *health* score of 0%. Because this is an important measure of health, status is given double the weight (by default) of the other statistics when overall interface health is computed.
Utilization Health (p_if%util)	The percent utilization of an interface. For example, a 50% utilization rate means that NNM measured the available bandwidth on an interface, and found that 50% was being used. Higher utilization rates translate into lower utilization *health* scores.
Inbound Error Health (p_if%inerrors)	The error rate (percent) for inbound data on the interface. High error rates translate into lower inbound error *health* scores.

Table 4-1 Interface Health Metrics (used by all gauges)

Statistic (MIB expression used)	Default Settings Description
Outbound Error Health (p_if%outerrors)	The error rate (percent) for outbound data on the interface. High error rates translate into lower outbound error *health* scores.

Table 4-2 Additional Metric used by Router Health Gauge Only

Statistic (MIB Expression used)	Default Settings Description
CPU Utilization Health (p_cisco5minavgbusy)	The percent utilization of the router's CPU. For example, a 50% utilization rate means that NNM measured the available CPU bandwidth, and found that 50% was being used. High utilization rates translate into low utilization *health* scores. This is a measurement of the Cisco MIB object: `local.system.augBusy5`, the "5 minute exponentially-decayed moving average of the CPU busy percentage."

Adding the Network Device Health Module

To insert the Network Device Health module into a portal view:

1. Access the portal view by logging on to SIP as a user with access to the appropriate role. If this user has access to multiple roles, switch to the appropriate role.

   Your currently assigned SIP role must have `ViewAdmin` editing permissions.

2. Navigate to the appropriate tab.
3. At the bottom of any narrow column, either:
   - Select `Network Device Health` from the `Select Module to Add` list box, and click `[Add]`, or
   - Click `[Edit]` to access the `Edit Column` page. Insert the Network Device Health module and place it into the desired location among other modules in the column. Click `[OK]` to save the changes and return to the main portal page.

A copy of the default Network Device Health module is inserted into your *PortalView*.xml file.

- If you want to modify this module instance, turn to “Editing Network Device Health Modules” on page 85.
- If you want to change the default module, see “Relevant Files” on page 114.

A newly displayed gauge is not fully functioning until the next data collection configuration update occurs so that NNM can supply the requested SNMP data. If you remove a gauge, NNM’s data collection is discontinued once the gauge has not been displayed in any portal view for 30 days. See “Collecting Data for Network Device Health Gauges” on page 105 and “ovcolautoconf.exe” on page 112.

When multiple NNM management stations provide raw data to the Network Device Health module, duplication of SNMP data collection is avoided by sending collection requests for a given node to only one of the NNM management stations. The NNM management station chosen is that which first returned raw data for that node. See “Collecting Data for Network Device Health Gauges” on page 105 for more information.

TIP

If you want to add a module to the list of available modules, see “Relevant Files” on page 114. You can create and add another instance of any module.

Editing Network Device Health Modules

Using the Network Device Health - Edit Page

You can easily modify the Network Device Health module in your portal view:

1. Access the portal view by logging on to SIP as a user with access to the appropriate role. If this user has access to multiple roles, switch to the appropriate role.

   Your currently assigned SIP role must have ViewAdmin editing permissions.

2. Navigate to the appropriate tab.
3. In the title bar of the Network Device Health module, click the edit button:
4. Select Choose from List. Make any desired changes. Click the [Help] button for specific instructions:

   - To add a gauge, select a gauge from the Available Health Categories list and click [Add]. Repeat until the Displayed Health Categories list contains all the gauges you want to display.
   - To remove a gauge, select the gauge in the Available Health Categories list and click [Remove]. Repeat until the Displayed Health Categories list contains only the gauges you want to display.
   - To rearrange the order of the gauges, select a gauge in the Displayed Health Categories list, and click [Up] or [Down]. Repeat until gauges are in the order you prefer.

5. To save the changes and return to the main portal page, click [OK].
6. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

A newly displayed gauge is not fully functioning until the next data collection configuration update occurs so that NNM can supply the requested data. If you remove a gauge, NNM's data collection is discontinued once the gauge has not been displayed in any portal view for 30 days. See “Collecting Data for Network Device Health Gauges” on page 105 and “ovcolautoconf.exe” on page 112.

When multiple NNM management stations provide raw data to the Network Device Health module, duplication of SNMP data collection is avoided by sending collection requests for a given node to only one of the NNM management stations. The NNM management station chosen is that which first returned raw data for that node. See “Collecting Data for Network Device Health Gauges” on page 105 for more information.

Directly Editing the PortalView.XML File

TIP

For the following adjustments, you must edit the XML file. It is recommended that you use the `Network Device Health - Edit` page for all other editing changes.

- Change the displayed title for this module instance.
- Add your own online help to the `[?]` button for this module.
- Specify the layout of all detail pages accessed through links in this Network Device Health module.

  NOTE: For global layout controls for the detail pages, see “Health detail pages setting” on page 93.

For each gauge:

- Change the displayed title.
- Control access to the detail pages.
- Change the metric components that determine how health is calculated.
- Change the filter that determines which devices are monitored for the health calculation.

To directly modify the XML code for a Network Device Health module:

1. Make a backup of XML files before you make changes. If you edit the XML file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.
2. Open your *`PortalView`*`.xml` file with an ASCII or XML editor. Portal view files are stored in the following directory or subdirectories below this one:

   Windows 2000: `%SIP_HOME%\conf\share\views`

   UNIX: `/opt/OV/SIP/conf/share/views`

   If a portal view file does not yet exist see the “Customizing Portal Views” section of the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`) and follow the procedure for creating a portal view.
3. Search for the following string to find your existing module to edit:

   `classid="com.hp.ov.portal.modules.health"`

   Module instances are wrapped in the `ModuleInstance` element. The `ModuleInstance id` must be unique among all module instances in the portal view file. For information about the `ModuleInstance` element, see the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “PortalView DTD” section.

   For example:

   ```
   <ModuleInstance
      classid="com.hp.ov.portal.modules.health"
      display="yes"
      help="/OvSipDocs/C/help//NNM/healthView.html"
      id="module17"
      rollupState="down"
      title="Network Device Health">
   </ModuleInstance>
   ```
4. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags. Alternately, copy the contents from the Network Device Health module’s default XML file (see “Relevant Files” on page 114):

   ```
   <NetworkHealth showRawData="no" showUnknown="no">
    <Summary display="yes" displayDepth="3"
             id="RouterHealth" title="Router Health">
       <Component href="#IfHealth"
                  vital="no" weight="1"/>
   ```

```
   <Component href="#CiscoCpuUtil"
              vital="no" weight="1"/>
   <NodeSelection id="Routers" op="AND" title="Routers">
     <CapabilityFilter op="OR">
             <Capability field="isRouter" value="true"/>
     </CapabilityFilter>
   </NodeSelection>
</Summary>
<Summary display="yes" displayDepth="3"
         id="ServerHealth" title="Server Health">
   <Component href="#IfHealth" vital="no" weight="1"/>
   <NodeSelection id="Servers" op="AND" title="Servers">
      <CapabilityFilter op="OR">
             <Capability field="isServer" value="true"/>
      </CapabilityFilter>
    </NodeSelection>
</Summary>
<Summary display="yes" displayDepth="3"
           id="InterfaceHealth" title="Interface Health">
   <Component href="#IfStatus" vital="yes" weight="2"/>
   <Component href="#IfUtil" vital="no" weight="1"/>
   <Component href="#IfInErrors" vital="no" weight="1"/>
   <Component href="#IfOutErrors" vital="no" weight="1"/>
   <InterfaceSelection id="AllInterfaces" op="AND"
           title="All Interfaces"/>
</Summary>
<Summary display="yes" displayDepth="3"
           id="CPEHealth" title="CPE Health">
   <Component href="#IfHealth" vital="no" weight="1"/>
   <NodeSelection id="CPE" op="AND"
             title="Customer Premise Equipment">
     <CapabilityFilter op="OR">
         <Capability field="isCPE" value="true"/>
     </CapabilityFilter>
    </NodeSelection>
</Summary>
<Summary display="yes" displayDepth="3"
          id="KeyDeviceHealth" title="Key Device Health">
   <Component href="#IfHealth" vital="no" weight="1"/>
   <NodeSelection id="KeyDevices" op="AND"
           title="Key Devices">
      <CapabilityFilter op="OR">
```

```
          <Capability field="isKeyDevice" value="true"/>
        </CapabilityFilter>
     </NodeSelection>
 </Summary>
</NetworkHealth>
```

See the comments in the OVNetworkHealth.dtd file for more information about the correct XML syntax (see “Relevant Files” on page 114).

5. To change the title of this Network Device Health module instance, change the title attribute:
   <ModuleInstance title="new title">.

   To change the title of all Network Device Health modules, change the title attribute in the registration file, see “Relevant Files” on page 114.

6. To launch your own help topic from the module’s [?] button, insert the help attribute into the <ModuleInstance>:

   help="/OVSipDocs/C/help/NNM/*topic*.html"

   Replace *topic*.html with the name of your help file. The help attribute allows you to override the default help URL defined in the module registration file. See the *SIP Deployment and Integration Guide* (SIP_Deployment_Integration.pdf), “Adding and Customizing Module Help Topics” section for more information about writing your own online help.

7. Specify the layout of the details pages (accessed by clicking the gauge icons in this module instance):

   - showRawData controls the display of the columns displaying the raw data returned from each Metric element (MIB object or MIB expression) referenced in each <Component> element of each gauge. The default setting is to hide these columns and only display the final *health* score.

   - showUnknown controls whether or not nodes or interfaces whose health score cannot be computed (usually, because that object’s status in NNM’s object database is set to unknown) are displayed in the gauge’s details page. The default setting is to exclude information derived from devices with "unknown" status. If this attribute is "on", rows for nodes/interfaces with "unknown" health status are added to the end of the detail table *if* the maxDetail attribute (in netHealthConfig.xml) allows.

8. To change the order in which the gauges are displayed, move the `<Summary>` through `</Summary>` blocks into the desired order between the `<ModuleInstance>` and the `</ModuleInstance>` elements.
9. To display or hide a particular gauge, set the `<Summary>` element's `display` attribute to `Yes` or `No`.
10. To change the title of a particular Network Device Health gauge, change the `title` attribute (`<Summary title="new title">`).
11. To control access to the Details Tables of each Network Device Health gauge, set the `displayDepth` attribute. Designate the appropriate setting:

    1=gauge only (no links to the details pages)

    2=link provided to node-only or interface-only details

    3=links provided to node *and* to interface details (if available)
12. The `Component` elements control how the gauge's health score is calculated, for example:

    ```
    <Component href="#CiscoCpuUtil" vital="no" weight="1"/>
    <Component href="#IfHealth" vital="no" weight="1"/>
    <Component href="#IfOutErrors" vital="no" weight="1"/>
    <Component href="#IfStatus" vital="yes" weight="2"/>
    <Component href="#IfUtil" vital="no" weight="1"/>
    ```

    You can change any of the following:

    - `href=" "`
      Specifies what is to be measured. The `href` values must point to `<Metric>` or `<ComponentGroup>` elements defined in the `netHealthConfig.xml` file. These elements specify exactly which MIB object or MIB expression is being requested by SIP. Each `ComponentGroup` combines several `Metrics.` Each `Metric` represents one of the following:

      — **MIB Objects**

      MIB objects are attributes that an SNMP agent on a network device allows to be queried by an NNM management station. Currently, any MIB object that returns a numeric value is supported. (Strings are not supported.)

— **MIB Expressions**

MIB expressions are a feature of NNM that allow for the creation of mathematical formulas comprised of MIB objects. MIB expressions allow more meaningful information to be gathered than is possible from individual MIB objects.

- `vital="yes"`
  If `yes`, when this `Metric` measures zero, the resource's health score is set to zero regardless of other health score measures. If `no`, this attribute is ignored.
- `weight`
  Controls how much emphasis is placed upon each `Component` (MIB object, MIB expression, or device status) being measured by a gauge. For example, if an interface's status is *down*, the status has more of an impact on the device health calculation than a high utilization measurement.

See the comments in the `netHealthConfig.xml` file for more information.

13. The `NodeSelection` and `InterfaceSelection` elements provide the display filtering for each gauge. See "Filtering Possibilities for the Network Device Health Module" on page 176 for more information.
14. Save the XML file.
15. After you make modifications to this XML file, validate the syntax. See "Validating XML Files" on page 225 for more information.
16. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

A newly displayed gauge is not fully functioning until the next data collection configuration update occurs so that NNM can supply the requested data. If you remove a gauge, NNM's data collection is discontinued once the gauge has not been displayed in any portal view for 30 days. See "Collecting Data for Network Device Health Gauges" on page 105 and "ovcolautoconf.exe" on page 112.

When multiple NNM management stations provide raw data to the Network Device Health module, duplication of SNMP data collection is avoided by sending collection requests for a given node to only one of the NNM management stations. The NNM management station chosen is that which first returned raw data for that node. See "Collecting Data for Network Device Health Gauges" on page 105 for more information.

Establishing Global Settings for All Network Device Health Modules

The netHealthConfig.xml file contains configuration information that applies, or potentially applies, to all health gauges in all portal views:

This information includes:

- Operational controls.
- Health detail pages setting.
- Rating specifications for gathered Metrics.
- Metric elements for data collection.
- Component groups that combine Metric elements.

See the netHealthConfig.dtd and netHealthConfig.xml files for more information.

Make a backup of XML files before you make changes. If you edit the XML file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.

1. With an ASCII or XML editing program, open the netHealthConfig.xml file:

   - *Windows 2000:*

     `%SIP_HOME%\conf\share\modules\health\netHealthConfig.xml`

   - *UNIX:*

     `/opt/OV/SIP/conf/share/modules/health/netHealthConfig.xml`

2. **Operational controls**

   Locate the NetworkHealthConfig element. For example:

```
<NetworkHealthConfig maxDetail="20" maxAge="60" rawDataRefresh="10" >
```

   The maxAge and rawDataRefresh attributes control how the Network Device Health modules interact with NNM:

   - maxAge
     SNMP data collected and supplied by NNM is checked to ensure that it is no older than the specified number of minutes. Older

data is ignored for health calculation purposes. This defines *near real-time data*. If young enough data does not exist for a particular metric, the metric is not used to compute overall health and the `"Data Unavailable"` message appears for that metric in Network Device Health detail pages.

- `rawDataRefresh`
  Sets the frequency with which raw data from NNM management stations should be updated. Expressed in minutes.

3. **Health detail pages setting**

The `maxDetail` attribute sets the maximum number of rows (one per node or interface) allowed in the detail pages. If more devices pass the filtering requirements of a gauge, only the specified number of devices with the poorest health score rating are displayed.

See also the per-module-instance detail page settings, page 86. You can specify layout of the detail pages in each module instance. You can control access to the detail pages within each gauge.

To change the icons in the Network Device Health detail pages, change the images in the following directory. Maintain the names of the images:

Windows 2000: *`%SIP_HOME%`*`\htdocs\C\images\health\`

UNIX: `/opt/OV/SIP/htdocs/C/images/health/`

Table 4-3 Health Status Icons for Details Pages

Filename	**Status**
unknown.gif	unknown
sad.gif	critical
ok.gif	minor
happy.gif	normal

4. **Rating specifications for gathered Metrics**

If desired, modify the rating scale. The `Rating` elements translate health *scores* to health *ratings*. The rating controls the color of the needle on the gauge, the width of each color around the outside edge of the gauge, as well as controlling which icon displays in each row of the health detail table. See previous step for more information about the icons.

The Rating Scale is used to translate the *scores* to health *ratings*. For example:

```
<Rating>
    <Scale lower="-1"  upper="-1" translation="unknown"/>
    <Scale lower="0"  upper="40"  translation="critical"/>
    <Scale lower="40" upper="80"  translation="minor"/>
    <Scale lower="80" upper="100" translation="normal"/>
</Rating>
```

You can change the lower and upper values, but do not change the translation values in the <Rating> block.

5. **Metric elements for data collection**

There is one Metric element specifying each MIB object and each MIB expression used by any gauge. The Metric elements are reusable in multiple gauges. The Metric elements also define the rules for calculating the score associated with each returned MIB value.

```
<Metric id="IfUtil" title="Utilization Health" autoConfig="yes"
                   href="snmp://%item%[IfIndex]/p_if%util">
        <Scale lower="0"  upper="25" translation="100"/>
        <Scale lower="25" upper="50" translation="70"/>
        <Scale lower="50" upper="75" translation="40"/>
        <Scale lower="75" upper="100" translation="0"/>
    </Metric>
```

You can change any of the lower, upper, or translation values in the <Scale> element.

6. **Component groups that combine Metric elements**

The ComponentGroup element allows you to combine multiple Metrics to make them easier to reference in gauge definitions. The Component element assigns a weight to each Metric, as well as identifies whether or not this particular Metric is considered vital (if this Metric measures zero, the resource’s health score is set to zero regardless of other health score measures):

```
<ComponentGroup id="IfHealth" title="Interface Health">
    <Component weight="2" vital="yes" href="#IfStatus"/>
    <Component weight="1" href="#IfUtil"/>
    <Component weight="1" href="#IfInErrors"/>
    <Component weight="1" href="#IfOutErrors"/>
</ComponentGroup>
```

7. After you finish making changes, save the file.
8. After you make modifications to the `netHealthConfig.xml` file, validate the syntax. See “Validating XML Files” on page 225 for more information.
9. After you make modifications to the `netHealthConfig.xml` file, you must restart the Tomcat engine. See Appendix A, “Restarting Tomcat,” on page 219 for more information.
10. Log into the SIP portal to ensure that you have the desired results.

Creating Your Own Network Device Health Gauge

You can create additional Network Device Health gauges. You define the status information and/or the SNMP data collections that your gauge measures. You also define which devices the gauge monitors. Please read the following topics before you start creating gauges:

- Overview for Creating Network Device Health Gauges
- Prerequisites to Creating Your Own Network Device Health Gauge
- Create Your Own Network Device Health Gauge

Overview for Creating Network Device Health Gauges

The components of Network Device Health Gauges are defined in the following files:

- `OVDefaultNetHealth.xml` (on the SIP server)

  This file defines the default set of health gauges for a new instance of the Network Device Health module. Each gauge is defined by a `Summary` element within this file. When a Network Device Health module is added to a tab with the `[Add]` button in the user interface, a *copy* of all gauges defined in the `OVDefaultNetHealth.xml` file is added to your current *`PortalView`*`.xml` file. These gauges show up in the selection list on the Network Device Health module's `Edit` page. You can show or hide any combination of the defined gauges.

TIP You can also customize specific instances of the gauges by directly editing *`PortalView`*`.xml` files.

- `netHealthConfig.xml` (on the SIP server)

  This file contains configuration information that applies, or potentially applies, to all health gauges for all portal users. It also contains a `Metric` element for each MIB object and each MIB expression used by any gauge `Summary` element. The `Metric` elements are reusable in multiple gauges. The `Metric` determines the rules for calculating and displaying the `Score` associated with each returned MIB value. If you don't find a `Metric` element for the MIB object or MIB expression that you wish to use, you need to write one.

- `mibExprAuto.conf` (on the NNM management station)

  This file defines SNMP MIB expressions used by the Service Information Portal. If you are using a MIB expression (mathematical formula comprised of MIB objects) that is not already defined in NNM, you need to define your MIB expression.

- `snmpRepAuto.templ` (on the NNM management station)

  This file is used by the `ovcolautoconf` program to automatically update NNM's Data Collector program to meet the current SIP requirements. Create an entry for each MIB object or MIB expression upon which you wish to collect data.

Prerequisites to Creating Your Own Network Device Health Gauge

Before creating your own network device health gauge:

1. Determine the set of nodes/interfaces from which this network device health gauge computes health scores. You write filters to define your list. For information about available filters, see Chapter 6, "Segmenting the NNM Data for Your Customers," on page 135 and "Filtering Possibilities for the Network Device Health Module" on page 176.

2. Determine which statistics should be used to compute health. Determine if the statistics can be provided by individual SNMP MIB objects or whether a mathematical formula using MIB objects is necessary.

   - **MIB Objects**

     MIB objects are attributes that an SNMP agent on a network device allows to be set or queried by an NNM management station. Currently, any MIB object that returns a numeric value is supported. (Strings are not supported.)

   - **MIB Expressions**

     MIB expressions are a feature of Network Node Manager that allow for the creation of mathematical formulas comprised of MIB objects and explicit numeric values. MIB expressions allow you to derive more meaningful information than you could gather from individual MIB objects.

NNM provides a variety of predefined MIB expressions. In addition, SIP provides the following MIB expressions (see Table 4-1 and Table 4-2 for more information):

— Interface % Utilization (`p_if%util`)

— Interface % Inbound Errors (`p_if%inerrors`)

— Interface % Outbound Errors (`p_if%outerrors`).

— Cisco CPU Utilization (`p_if%p_cisco5minavgbusy`).

Service Information Portal preconfigured MIB expressions are defined in the following file on the NNM management station:

— *Windows NT/2000:*

```
NNM_install_dir\conf\ovcolautoconf\mibExprAuto.conf
```

— *UNIX:*

```
/etc/opt/OV/share/conf/ovcolautoconf/mibExprAuto.conf
```

Create Your Own Network Device Health Gauge

To create your own network device health gauge, complete the following steps. Note that some steps are carried out on the SIP server and others on each NNM management station that supplies data for the new gauge.

Before you start this series of steps, complete the prerequisite steps. Have your list of nodes and interfaces to be monitored and your list of statistics to be gathered.

Steps on each NNM management station

1. Check NNM’s Data Collection configuration to see if the statistics that you need are already being collected. If you find all the statistics that you need, skip to “Steps on the SIP server” on page 100.

   From any NNM submap, select `Options:Data Collections & Thresholds`. Review the list of currently configured collections. See the *Managing Your Network with NNM* book provided with NNM for more information. See also the Help information from within the Data Collections & Thresholds window.

2. Make sure that the MIB specification files, whose objects you wish to use, are loaded into NNM. From any NNM submap, select `Options:Load/Unload MIBs`. See *Managing Your Network with NNM* for more information about loading MIB specification files into NNM.

3. *OPTIONAL:* If you need to write your own MIB expression, add it to the `mibExprAuto.conf` file (or copy and modify one of the MIB expressions supplied), for information about writing MIB expressions, please see the *Managing Your Network with NNM* book provided with NNM, and the *mibExpr.conf* and the *mib.coerce* reference pages in NNM's online help (or the UNIX manpages).

   Before adding your new MIB expression to the `mibExprAuto.conf` file, save a copy of the original file. After writing your new MIB expression, you must load the new expression into NNM by typing the following at the command prompt on the NNM management station. This command checks the syntax of your MIB expression and forces an update to NNM's `mibExpr.conf` file which allows data collections to be enabled:

   - *Windows NT/2000:*

```
xnmcollect -loadExpr NNM_install_dir\conf\ovautocolconf\mibExprAuto.conf
```

   - *UNIX:*

```
xnmcollect -loadExpr /etc/opt/OV/share/conf/ovautocolconf/mibExprAuto.conf
```

4. Add an entry for each new MIB object or MIB expression to the `snmpRepAuto.templ` file to enable automatic configuration of NNM Data Collector. This ensures that the SNMP data needed to drive your gauges is available. For information about the attributes needed for each data collection, see "snmpRepAuto.templ" on page 111.

   When executed, the `ovcolautoconf.exe` program uses this file to configure the `snmpRep.conf` file with the most recent SIP data collection requests. This keeps NNM's Data Collector in sync with changes in the *`PortalView`*`.xml` files. For more information about how the data collection process works, see "Collecting Data for Network Device Health Gauges" on page 105.

Steps on the SIP server

Make a backup of XML files before you make changes. If you edit the XML file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.

1. Check the `netHealthConfig.xml` file.

   Make sure that there is a `Metric` element for each MIB object and each MIB expression that you intend to use. If not, add a `Metric` element to the `netHealthConfig.xml` file that defines the rules for determining the `score` associated with each newly defined MIB value. See the `netHealthConfig.dtd` file and `netHealthConfig.xml` file for information about the attributes needed for each `Metric` element. Example:

```
<Metric id="IfUtil" title="Interface Utilization" autoConfig="yes"
   href="snmp://%item%[IfIndex]/p_if%util">
      <Scale lower="0"  upper="25" translation="100"/>
      <Scale lower="25" upper="50" translation="70"/>
      <Scale lower="50" upper="75" translation="40"/>
      <Scale lower="75" upper="100" translation="0"/>
</Metric>
```

2. After modifying the `netHealthConfig.xml` file, save the file.

3. After you make modifications to the `netHealthConfig.xml` file, validate the syntax. See “Validating XML Files” on page 225.

4. After you make modifications to the `netHealthConfig.xml` file, you must restart the Tomcat engine. See Appendix A, “Restarting Tomcat,” on page 219 for more information.

5. Modify the `OVDefaultNetHealth.xml` file by adding a `Summary` element that defines the new network device health gauge. See the `OVNetworkHealth.dtd` file for information about the attributes needed within each `Summary` element. Example:

```
<Summary title="Interface Health" display="yes" displayDepth="3">
   <Component weight="2" href="#IfStatus"/>
   <Component weight="1" href="#IfUtil"/>
   <Component weight="1" href="#IfInErrors"/>
   <Component weight="1" href="#IfOutErrors"/>
   <InterfaceSelection title="Access Links" id="AccessLinks" op="AND">
        Your filters would be defined here
   </InterfaceSelection>
</Summary>
```

6. After you make modifications to the `OVDefaultNetHealth.xml` file, validate the syntax. See “Validating XML Files” on page 225 for more information.

7. You are now ready to insert the new gauge into any *`PortalView`*`.xml` file.

   - Existing Network Device Health module instances in *`PortalView`*`.xml` files: copy your new gauge’s `Summary` element from the `OVDefaultNetHealth.xml` file, then open the *`PortalView`*`.xml` file and paste the new gauge’s `Summary` into the desired location.

   - New Network Device Health module instances: after creating a new *`PortalView`*`.xml` file, log into the portal view and navigate to the desired tab. Select `Network Device Health` in the list of available modules and click `[Add]`. If necessary, click the `[Edit]` button in the title bar of the newly added Network Device Health module, and select the new gauge from the list of available gauges.

   A newly displayed gauge is not fully functioning until the next data collection configuration update occurs so that NNM can supply the requested data. If you remove a gauge, NNM’s data collection is discontinued once the gauge has not been displayed in any portal view for 30 days. See “Collecting Data for Network Device Health Gauges” on page 105 and “ovcolautoconf.exe” on page 112.

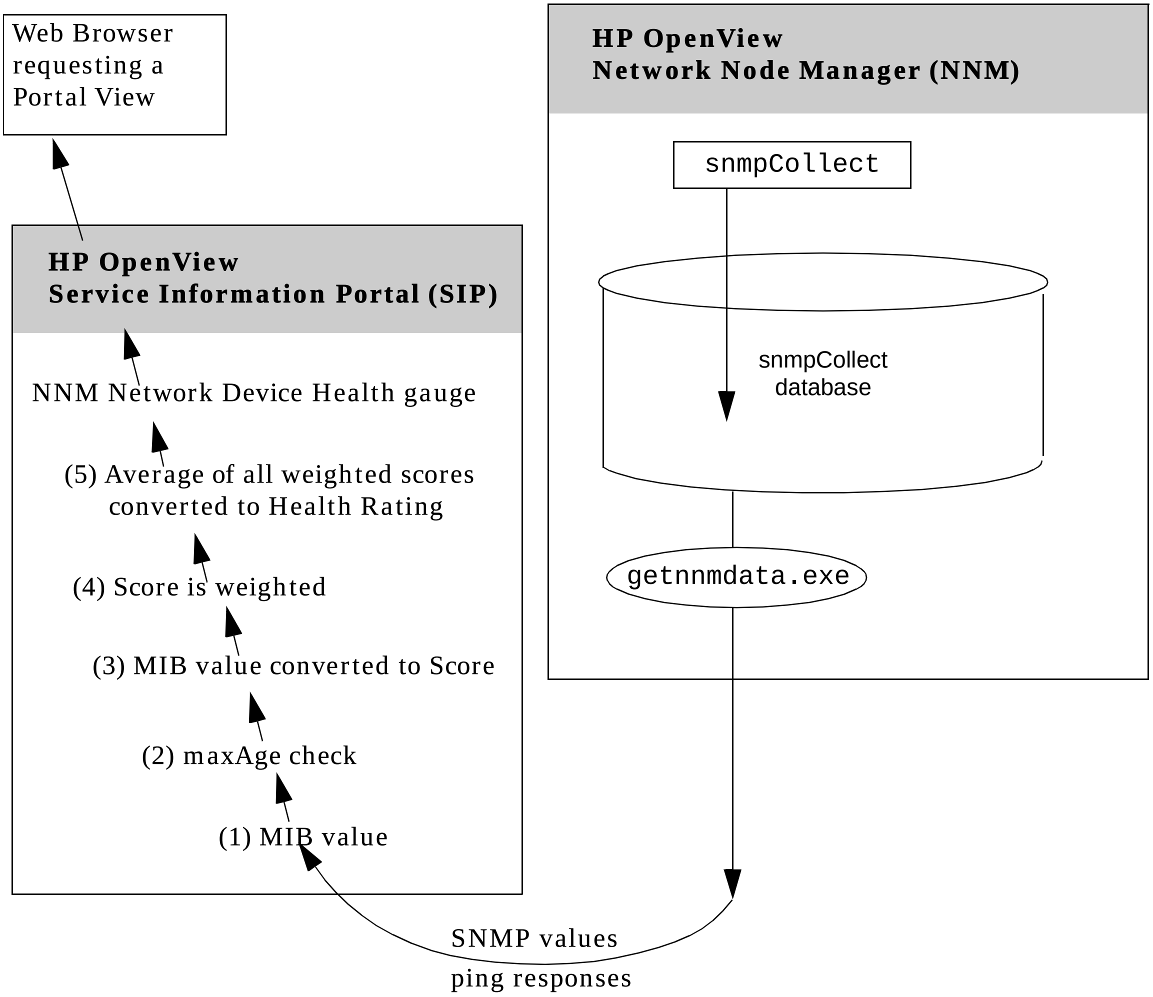
   When multiple NNM management stations provide raw data to the Network Device Health module, duplication of SNMP data collection is avoided by sending collection requests for a given node to only one of the NNM management stations. The NNM management station chosen is that which first returned raw data for that node. See “Collecting Data for Network Device Health Gauges” on page 105 for more information.

NOTE You can edit the *`PortalView`*`.xml` file directly after inserting the new gauge and modify the `Summary` element to further customize the desired results. See “Filtering Possibilities for the Network Device Health Module” on page 176.

Controlling How Health Is Calculated

There are five steps in calculating the health rating. You can make modifications to any combination of these steps.

Figure 4-1 **Calculating Health Ratings to Display in Gauges**


Web Browser requesting a Portal View
HP OpenView Network Node Manager (NNM)
snmpCollect
snmpCollect database
getnnmdata.exe
HP OpenView Service Information Portal (SIP)
NNM Network Device Health gauge
(5) Average of all weighted scores converted to Health Rating
(4) Score is weighted
(3) MIB value converted to Score
(2) maxAge check
(1) MIB value
SNMP values
ping responses

1. Data is gathered through HP OpenView Network Node Manager and the requested *values* are returned to the Service Information Portal.

For more information about controlling this process, see "Collecting Data for Network Device Health Gauges" on page 105.

2. For SNMP data, the returned values are checked to ensure that they are valid by noting how many *minutes* have passed since they were collected by NNM. This is controlled by the `maxAge="minutes"` attribute in the `netHealthConfig.xml` file. Example:

   ```
   <NetworkHealthConfig maxAge="60">
   ```

   The `maxAge` attribute affects all health gauges defined within all portal views. See "Establishing Global Settings for All Network Device Health Modules" on page 92 for more information.

3. In the `netHealthConfig.xml` file, each requested MIB object or MIB expression has a corresponding `<Metric>` element that includes a scale for converting the returned MIB *value* to a `Score` (`translation="`0-100"). For example:

   ```
   <!-- This scale maps utilization percentage to health   -->
   <!-- scores 0-100. Note that higher utilization results -->
   <!-- in a lower score.                                  -->
           <Scale lower="0"  upper="25" translation="100"/>
           <Scale lower="25" upper="50" translation="70"/>
           <Scale lower="50" upper="75" translation="40"/>
           <Scale lower="75" upper="100" translation="0"/>


   <!-- This scale maps inbound error percentage to health -->
   <!-- scores 0-100. Note that higher error rates result   -->
   <!-- in a lower score.                                  -->
           <Scale lower="0" upper="2"  translation="100"/>
           <Scale lower="2" upper="5"  translation="75"/>
           <Scale lower="5" upper="10" translation="25"/>
           <Scale lower="10"           translation="0"/>
   ```

4. In the *`portalView`*`.xml` files, each gauge's `<Summary>` element assigns a `weight` to each `Score`. For example:

   ```
   <Component href="#CiscoCpuUtil" vital="no" weight="1"/>
   <Component href="#IfHealth" vital="no" weight="1"/>
   <Component href="#IfOutErrors" vital="no" weight="1"/>
   <Component href="#IfStatus" vital="yes" weight="2"/>
   <Component href="#IfUtil" vital="no" weight="1"/>
   ```

5. Based upon the weighted average (mean) of all Scores, a health Rating is computed for each node or interface being monitored by the

gauge. The gauge's health Rating is computed as the average (mean) health Rating of all nodes/interfaces represented by the gauge. The `Rating Scale` (as defined in the `Rating` element of the `netHealthConfig.xml file`) is used to translate the *scores* to health *ratings*. Example:

```
<Rating>
    <Scale lower="-1"  upper="-1" translation="unknown"/>
    <Scale lower="0"  upper="40"  translation="critical"/>
    <Scale lower="40" upper="80"  translation="minor"/>
    <Scale lower="80" upper="100" translation="normal"/>
</Rating>
```

The rating scale affects *all* gauges defined within any portal.

If desired, modify the rating scale. The rating controls the color of the needle on the gauge, the width of each color around the outside edge of the gauge, as well as controlling which icon displays in each row of the health detail table. See "Establishing Global Settings for All Network Device Health Modules" on page 92 for more information about the icons.

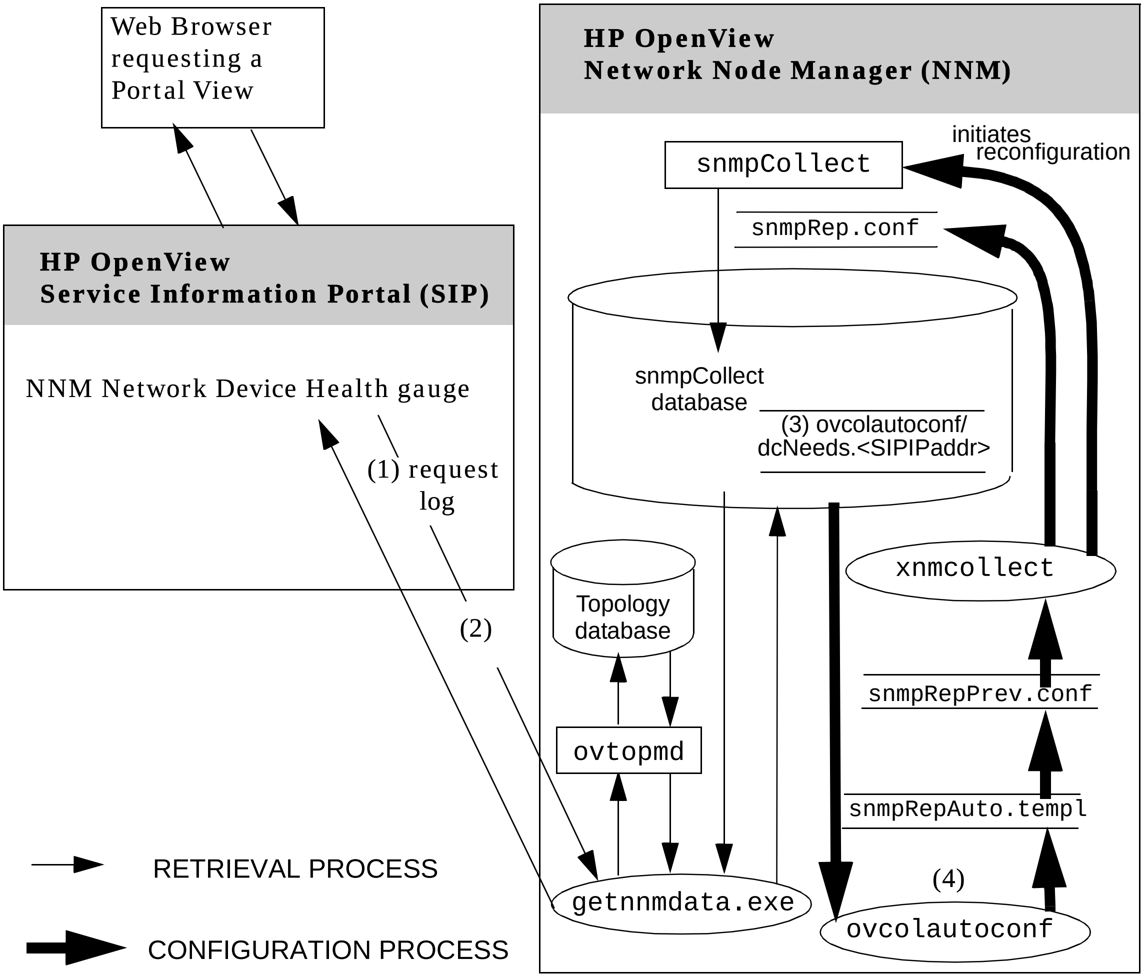
You can change the `lower` and `upper` values, but do not change the `translation` values in the `<Rating>` block.

6. After you make modifications to any of these XML file, validate the syntax. See "Validating XML Files" on page 225 for more information.
7. After you make modifications to the `netHealthConfig.xml` file, you must restart the Tomcat engine. See Appendix A, "Restarting Tomcat," on page 219 for more information.

Collecting Data for Network Device Health Gauges

HP OpenView Network Node Manager (NNM) collects all SNMP data requested by HP OpenView Service Information Portal (SIP) and returns current information about device status.

Figure 4-2 Communication Process for the Network Device Health Module


Web Browser requesting a Portal View
HP OpenView Service Information Portal (SIP)
NNM Network Device Health gauge
(1) request log
(2)
HP OpenView Network Node Manager (NNM)
initiates reconfiguration
snmpCollect
snmpRep.conf
snmpCollect database
(3) ovcolautoconf/ dcNeeds.<SIPIPaddr>
Topology database
ovtopmd
xnmcollect
snmpRepPrev.conf
snmpRepAuto.templ
(4)
getnnmdata.exe
ovcolautoconf
RETRIEVAL PROCESS
CONFIGURATION PROCESS

When multiple NNM management stations provide raw data to the Network Device Health module, duplication of SNMP data collection is avoided by sending collection requests for a given node to only one of the NNM management stations. The NNM management station chosen is that which first returned raw data for that node. See “Collecting Data for Network Device Health Gauges” on page 105 for more information.

SIP depends upon two programs that reside on each NNM management station (`getnnmdata.exe` and `ovcolautoconf.exe`) to collect requested data:

1. Each time a Network Device Health gauge is displayed, SIP logs the underlying data requests.

   A list of requested MIB objects and MIB expressions from any Network Device Health module gauge is compiled by SIP. The list documents which MIB objects and MIB expressions are being requested for which network devices from which NNM management stations.

NOTE The underlying MIB objects and MIB expressions appear in Network Device Health gauge definitions in the `PortalView.xml` file as the `Component` elements’ `href` attributes. Each `href` attribute must have a corresponding `Metric` element defined in the `netHealthConfig.xml` file that specifies exactly which MIB object or MIB expression is being requested.

2. SIP contacts the `getnnmdata.exe` on each NNM management station that is configured through the SIP Configuration Editor. The frequency of this action is determined by the `rawDataRefresh` parameter setting in the `netHealthConfig.xml` file on the SIP server (by default, every 10 minutes).
3. SIP receives the most recent data collection results from the NNM database. SIP also places the current request log file in the `ovcolautoconf` directory. Requests from each SIP server are gathered here (`dc.needs<SIPserverIPaddress>`).

TIP You must create the `ovcolautoconf` directory before this step works. See "To enable the Network Device Health Module to Configure NNM Data Collection" on page 34 for more information.

4. To complete the automatic configuration process, run the `ovcolautoconf.exe` command. The `ovcolautoconf` command must be executed on the NNM management station, either manually or as a scheduled task that you define. `ovcolautoconf` does the following:

   - All SIP servers' data collection needs are processed. The list of data collection requests is configured using the information in `snmpRepAuto.templ` file and placed in the `snmpRepPrev.conf` file.
   - If necessary, NNM's Data Collector configurations are updated by making SIP additions or changes to the `snmpRep.conf` file (one of two configuration files used by the NNM Data Collector program). The `snmpRep.conf` file is used by the SNMP Data Collector as a guide for gathering data. The entries from the HP OpenView Service Information Portal do not interfere with data collection configurations that were entered directly through NNM.
   - Data collections are configured on an *as-needed* basis, rather than a *potentially* needed basis. In other words, until a gauge is displayed in a portal view, no data collection is initiated.
   - If a gauge is not displayed for 30 days (default setting), the data collections are discontinued (provided they are not needed by other OpenView products). See "ovcolautoconf.exe" on page 112 for more information.

Network Device Health gauges calculate the health of specific network devices using information gathered by NNM management stations (see "Controlling How Health Is Calculated" on page 102). Changes are visible in the SIP's Network Device Health gauges each time the portal view is displayed or refreshed.

TIP

See “The Data Collection Process for the Network Device Health Module” on page 40 for important additional information about the SIP data collection process for the Network Device Health module.

mibExprAuto.conf

This file resides on the NNM management station. It contains the MIB expression definitions that are being used by SIP for Network Device Health calculations. MIB expressions are a feature of Network Node Manager that allow for the creation of mathematical formulas comprised of MIB objects. MIB expressions allow you to derive more meaningful information than you could gather from individual MIB objects.

Service Information Portal preconfigured MIB expressions are defined in this file (see "The Data Collection Process for the Network Device Health Module" on page 40, for information about installing this file onto your NNM 6.1 management station). More information is available within the file itself:

- *Windows NT/2000:*

`NNM_install_dir\conf\ovcolautoconf\mibExprAuto.conf`

- *UNIX:*

`/etc/opt/OV/share/conf/ovcolautoconf/mibExprAuto.conf`

OPTIONAL: If you need to write your own MIB expression:

1. Before you modify the `mibExprAuto.conf` file, save a copy of the original.
2. Add your MIB expression to the `mibExprAuto.conf` file (TIP: copy and modify one of the MIB expressions supplied).

   For information about writing MIB expressions, see the *Managing Your Network with NNM* book provided with NNM. See also the *mibExpr.conf* and the *mib.coerce* reference pages in NNM's online help (or the UNIX manpages).
3. Verify that the MIB files, whose objects you wish to use, are loaded into NNM. See the *Managing Your Network with NNM* book provided with NNM for more information about loading MIB files into NNM.

4. After writing your new MIB expression, you must load the new expression into NNM by typing the following at the command prompt. This command checks the syntax of your MIB expression and forces an update to NNM's `mibExpr.conf` file which allows data collections to be enabled:

   - *Windows NT/2000:*

```
xnmcollect -loadExpr
NNM_install_dir\NNM\conf\ovcolautoconf\mibExprAuto.conf
```

   - *UNIX:*

```
xnmcollect -loadExpr /etc/opt/OV/share/conf/ovcolautoconf/mibExprAuto.conf
```

5. Make a new entry into the `snmpRepAuto.templ` file so that NNM could begin collecting the requested information (see "snmpRepAuto.templ" on page 111).

6. Review the following section for possible additional required steps: "Create Your Own Network Device Health Gauge" on page 98.

snmpRepAuto.templ

This file exists on each NNM management station. (See "The Data Collection Process for the Network Device Health Module" on page 40 for installation instructions.) If you create any new gauges, you must ensure that there is one entry in the `snmpRepAuto.templ` file for each MIB object and each MIB expression that needs to be collected. (See "Relevant Files" on page 114.)

To view the list of configured collections and make any necessary additions, at the command line type the following:

- *Windows NT/2000:*

  ```
  xnmcollect -snmpColConfFile snmpRepAuto.templ
  ```

- *UNIX:* log in as root and then type,

  ```
  xnmcollect -snmpColConfFile snmpRepAuto.templ
  ```

Review the list. In the `Source` field you will see the variable _NODE_, which is automatically replaced with any specific devices requested by SIP.

If you do not see each MIB object and/or MIB expression that you are using in your gauge, create a new Data Collector entry:

1. Highlight any MIB Object in the top half of the window and select `Edit:MIB Object->Copy`.
2. Select the new MIB object or MIB expression that you wish to collect data upon.
3. You can change the collection interval setting, otherwise leave the settings as they are. You should see the variable `_NODE_` in the `Source` field.

See also "Creating Your Own Network Device Health Gauge" on page 96, "Collecting Data for Network Device Health Gauges" on page 105, and "ovcolautoconf.exe" on page 112.

ovcolautoconf.exe

ovcolautoconf.exe configures the NNM SNMP Data Collector (snmpCollect) to gather data requested by the HP OpenView Service Information Portal (SIP).

SYNOPSIS

```
ovcolautoconf [-verbose] [-outfile <filename>] [-maxConfAge
<#ofdays>]
```

DESCRIPTION

ovcolautoconf is a Network Node Manager (NNM) command that configures the NNM SNMP Data Collector (snmpCollect) to gather data requested by SIP. If invoked without the -outfile option, ovcolautoconf updates NNM's data collection configuration to reflect SIP SNMP data needs. Specifically, ovcolautoconf processes SIP server configuration request files found in $OV_DB/snmpCollect/ovcolautoconf. These files have names of the form dcNeeds.<SIP Server IP Addr>. [For information about how these request files are placed in this directory, see "The Data Collection Process for the Network Device Health Module" on page 40]. The template file $OV_CONF/ovcolautoconf/snmpRepAuto.templ is used to construct data collector configuration entries corresponding to these requests. The configuration entries are then loaded into the data collector configuration file $OV_CONF/snmpRep.conf, and snmpCollect is notified that its configuration has been modified. ovcolautoconf clears the SIP server request files after successfully processing them. The most recent data collector configuration submitted by ovcolautoconf can be found in the file $OV_DB/snmpCollect/ovcolautoconf/snmpRepPrev.conf

If the data collection configuration needs have not changed since the last execution of ovcolautoconf, no changes to snmpRep.conf are made and no reconfiguration event is sent to snmpCollect.

ovcolautoconf automatically removes data collector configuration entries are no longer needed by SIP. See the discussion of the -maxConfAge option below for details

To change the number of days SIP waits before deleting any inactive data collection configurations (default 30), type the following command. There is no way to permanently change this setting. Include this command in your scheduled script or each time you manually run `ovcolautoconf`:

`ovcolautoconf -maxConfAge` ***`#ofdays`***

OPTIONS

`-maxConfAge <#ofdays>` Removes configuration entries that have gone unrequested for the specified number of days. Applies only to configuration entries submitted by `ovcolautoconf`. Default is 30 days.

`-outfile <filename>` Don't update NNM's data collection configuration, but instead write the configuration to the specified file.

`- verbose` Send verbose output, including notification of configuration entries that have been aged out, to `stdout`.

TROUBLESHOOTING

Warning and error messages are sent to `stderr`.

FILES ON THE NNM MANAGEMENT STATION

Windows NT/2000:

`NNM_install_dir``\conf\ovcolautoconf\snmpRepAuto.templ`

`NNM_install_dir``\conf\ovcolautoconf\mibExprAuto.conf`

`NNM_install_dir``\databases\snmpCollect\ovcolautoconf\snmpRepPrev.conf`

`NNM_install_dir``\databases\snmpCollect\ovcolautoconf\dcNeeds.<`*`SIPserverIPaddress`*`>`

UNIX:

`/etc/opt/OV/share/conf/ovcolautoconf/snmpRepAuto.templ`

`/etc/opt/OV/share/conf/ovcolautoconf/mibExprAuto.conf`

`/var/opt/OV/share/databases/snmpCollect/ovcolautoconf/snmpRepPrev.conf`

`/var/opt/OV/share/databases/snmpCollect/ovcolautoconf/dcNeeds.<`*`SIPserverIPaddress`*`>`

See also the *ovrequestd*, *snmpCollect*, *snmpCol.conf*, *mibExpr.conf*, and the *mib.coerce* reference pages in NNM's online help (or the UNIX manpages) for information about NNM's data collection process.

Relevant Files

The Network Device Health module must follow the rules defined in the following DTD files. See the comments in the DTD files for an explanation of each element used in the XML files:

- `mgmtStations.dtd & nmConfig.dtd & mgmtStations.xml`
  This XML file contains the list of all NNM management stations with which SIP is allowed to communicate. Use the SIP Configuration Editor program to make changes to this file. You must specify whether or not the Network Device Health module is allowed to request SNMP data. You must provide information about which ports are being used by the NNM `OVwDB` process and NNM web server. See "Establishing Communication Between NNM and SIP" on page 28.
- `OVModuleRegistraton.dtd & OVRegNetHealth.xml`
  This XML file grants access to the Network Device Health module through the SIP framework so that it is available for your use. To add another instance of the Network Device Health module to the SIP module selection list, you copy and rename the `OVRegNetHealth.xml` and the `OVDefaultNetHealth.xml` files. Then update the `description`, `title`, `classid`, `help`, and `defaultConfigXML` attribute values in the new registration file.

  If you make any changes to a registration file, you must follow the directions in "Restarting the Servlet Engine" on page 220.
- `OVNetworkHealth.dtd & OVDefaultNetHealth.xml`

  This DTD defines the rules for configuring any Network Device Health modules. The XML file contains the *default* Network Device Health module. The contents of the default file is inserted into your portal each time you use the [Add] button to insert the Network Device Health module.

  You can modify the `OVDefaultNetHealth.xml` file to meet your needs. Either:

  — Directly edit the XML code in the `OVDefaultNetHealth.xml` file, or

  — Insert a Network Device Health module into any portal. Modify the module to meet your needs. Then, copy the modified XML code for the module from your portal view file, and paste it into the `OVDefaultNetHealth.xml` file.

See "Editing Network Device Health Modules" on page 85 for more information.

- `netHealthConfig.dtd & netHealthConfig.xml`
  This XML file contains the global settings used by all Network Device Health modules. See "Establishing Global Settings for All Network Device Health Modules" on page 92 for more information.
- `PortalView.dtd &` *`PortalView`*`.xml`

  This DTD provides the rules for formatting the XML code in your portal view files. See the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), "Customizing Portal Views" section for more information about creating portal view files.
- `/htdocs/C/help/NNM/*.html`

  This directory contains the Help topics for Network Device Health modules, accessed by clicking the `[?]` button. If you want to supply your own customized help files, see the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), "Adding and Customizing Module Help Topics" section.

Table 4-4 Network Device Health Module Files on the SIP Server

File Name	**Windows 2000 Location** **%SIP_HOME%/....**	**UNIX Location** **/opt/OV/SIP/....**
mgmtStations.dtd	conf/share/stations/	conf/share/stations/
nmConfig.dtd	conf/share/stations/	conf/share/stations/
mgmtStations.xml	conf/share/stations/	conf/share/stations/
OVModuleRegistration.dtd	registration/	registration/
OVRegNetHealth.xml	registration/	registration/
netHealthConfig.dtd	conf/share/modules/health/	conf/share/modules/health/
netHealthConfig.xml	conf/share/modules/health/	conf/share/modules/health/
OVNetworkHealth.dtd	conf/share/views/	conf/share/views/
OVDefaultNetHealth.xml	registration/defaults/	registration/defaults/

Table 4-4 Network Device Health Module Files on the SIP Server

File Name	Windows 2000 Location %SIP_HOME%/....	UNIX Location /opt/OV/SIP/....
PortalView.xml	conf/share/views	conf/share/views
*.html	htdocs\C\help\NNM\	htdocs/C/help/NNM/

The files in Table 4-5 reside on the NNM management station:

- `getnnmdata.exe`

  This is the NNM program that receives data collection requests from and communicates data collection information to SIP. See “Collecting Data for Network Device Health Gauges” on page 105.

- `dcNeeds.<SIPserver>`

  These NNM files are the data request logs received from SIP. See “Collecting Data for Network Device Health Gauges” on page 105.

- `ovcolautoconf.exe & snmpRepAuto.templ`

  This NNM program must be run manually or scheduled to run on a regular basis in order to upload the SIP data collection requests into NNM’s data collection program. See “Collecting Data for Network Device Health Gauges” on page 105 and “ovcolautoconf.exe” on page 112.

  The `snmpRepAuto.templ` file on the NNM management station contains the data collection settings that are assigned to each MIB or MIB expression requested by SIP. You can modify these default settings. See “Creating Your Own Network Device Health Gauge” on page 96, “Collecting Data for Network Device Health Gauges” on page 105, and “snmpRepAuto.templ” on page 111.

- `mibExprAuto.conf`

  This NNM configuration file defines SIP’s MIB expressions for NNM’s data collection program. See “To enable the Network Device Health Module to Configure NNM Data Collection” on page 34 and “mibExprAuto.conf” on page 109.

Table 4-5 Data Collection Process Files on the NNM Management Station

File Name	Windows NT/2000 Location *NNM_install_dir*/...	UNIX Location....
getnnmdata.exe	www/cgi-bin/	/opt/OV/www/cgi-bin/
dcNeeds.<SIPserver>	databases/snmpCollect/ovcolautoconf/	/var/opt/OV/share/databases//snmpCollect/ovcolautoconf/
ovcolautoconf.exe	bin/	/opt/OV/bin
snmpRepAuto.templ	conf/ovcolautoconf/	/etc/opt/OV/share/conf/ovcolautoconf
mibExprAuto.conf	conf/ovcolautoconf/	/etc/opt/OV/share/conf/ovcolautoconf

5 Topology Module

Understanding Topology Data

The Topology module displays one or more Network Node Manager (NNM) submaps. Submaps provide a graphical view of the network environment or system management information. Each submap displays a different perspective of the environment. You may be able to display another submap by clicking on a symbol; for example display a submap showing all interfaces within a router by clicking on the router symbol. Click the browser's [Back] button to return to the previous submap.

Each submap that you display is associated with an NNM management station and a map. NNM management stations provide and maintain the operational SNMP/network management information.

Changes in network configuration and device status are visible in the Topology module submaps each time the portal view is displayed or refreshed. NNM sends the most recent information to SIP upon demand.

If changes are made within NNM to the “Symbol Type” assigned to particular devices, SIP receives the changes according to the schedule established by the `symbolFetchRateInMin` attribute in the `topologyConfig.xml` file.

Several things are important to know about the submaps displayed through the Topology module:

- NNM must be configured for use with the Topology module. For detailed information, see “To enable Topology Module Access to NNM Data” on page 32.
- Submaps that are targeted within Topology modules must either be currently displayed on the NNM management station or be configured as *persistent* (not *transient*) within NNM before they display in the portal. This means that the submaps must be stored in RAM on the NNM management station and not generated on-the-fly upon request.
- Submaps that are accessed through drill-down (optional behavior, default = no drill-down) might be *transient* within NNM, depending upon the global settings you choose.
- If your submaps have *auto-layout* turned off in NNM, the New Object Holding Area does not display in SIP. You must move symbols out of the New Object Holding Area to make them visible in SIP.

- The submaps displayed in SIP are actually completely new redrawn versions. Outer shapes for symbols are not dynamically generated. SIP supported outer shapes are circle, square, diamond, hexagon and octagon. Square is the generic shape for any symbol from NNM that uses a shape that is unsupported in SIP.

Adding a Topology Module to a Portal View

To insert the Topology module into a portal view:

1. Access the portal view by logging on to SIP as a user with access to the appropriate role. If this user has access to multiple roles, switch to the appropriate role.

   Your currently assigned SIP role must have `ViewAdmin` editing permissions.

2. Navigate to the appropriate tab.
3. At the bottom of any wide column, either:
   - Select `Topology` from the `Select Module to Add` list box, and click `[Add]`, or
   - Click `[Edit]` to access the `Edit Column` page. Insert the Topology module and place it into the desired location among other modules in the column. Click `[OK]` to save the changes and return to the main portal page.

A copy of the default Topology module is inserted into your *`PortalView`*`.xml` file.

- If you want to modify this module instance, turn to “Editing the Topology Module” on page 122.
- If you want to change the default module, see “Relevant Files” on page 131.

TIP If you want to add a module to the list of available modules, see “Relevant Files” on page 131. You can create and add another instance of any module.

Editing the Topology Module

Using the Topology - Edit Page

You can easily modify the Topology module in your portal view:

1. Access the portal view by logging on to SIP as a user with access to the appropriate role. If this user has access to multiple roles, switch to the appropriate role.

   Your currently assigned SIP role must have `ViewAdmin` editing permissions.

2. Navigate to the appropriate tab.

3. In the title bar of the Topology module, click the edit button:
   ✎

4. Make any desired changes. Click the `[Help]` button if you need more information.

5. Set the `Show Status in submap` check box. All displayed submaps in this Topology module are affected by this symbol status setting:

   - If selected, all symbols and connection lines display their current status color from NNM.
   - If deselected, all symbols assume the NNM *administrative status* of "unmanaged" (cream colored by default). All connection lines remain black.

6. Set the `Drill Down in submaps` check box. All displayed submaps in this Topology module are affected by this setting:

   - If selected, all submaps allow drill-down access through the NNM hierarchy.

     The submaps that can be accessed through drill-down are controlled by the global settings in the `topologyConfig.xml` file. See "Establishing Global Settings for All Topology Modules" on page 129 for more information.

   - If deselected, none of the displayed submaps provide drill-down access.

7. To add a submap, select the name of the NNM management station that has the submap you wish to access. The list contains all NNM management stations you configured in “Establishing Communication Between NNM and SIP” on page 28.

8. In the `Map` field, type the name of the map, then click `[List Submaps]`.

   NOTE: Submaps must be currently displayed on the NNM management station or configured as *persistent* (not *transient*) within NNM before they are available for display in the SIP portal. This means that the submap must be stored in RAM on the NNM management station and not generated on-the-fly upon request.

9. From the `Available Submaps` list, select a submap (if duplicate submap names occur in the list, verify the path displayed after the submap name).

10. Click `[Add]`. The submap name is moved to the `Submaps to Display` list and added to the bottom of the `Displayed Submaps` list.

11. To select a submap from a different NNM management station, return to step 7 and select the next NNM management station.

12. To adjust the order in which the submaps are displayed, in the `Displayed Submaps` list, select the submap name and use the `[Up]` and `[Down]` buttons to navigate the new submap into the correct display location.

13. To remove a submap from the Topology module, in the `Displayed Submaps` list, select the submap name and use the `[Delete]` button.

14. To save the changes and return to the main portal page, click `[OK]`.

15. Log out of the SIP portal.

16. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Directly Editing the PortalView.XML File

TIP For the following adjustments, you must edit the XML file. It is recommended that you use the `Topology - Edit` page for all other editing changes.

- Change the displayed title for this module instance.
- Add your own online help to the `[?]` button for this module.
- Changing the title of a submap.
- Bypassing the Management Data filter for a submap.
- Changing the displayed size of a submap.

To directly modify the XML code for a Topology module:

1. Make a backup of XML files before you make changes. If you edit the XML file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.
2. Open your *`PortalView`*`.xml` file with an ASCII or XML editor. Portal view files are stored in the following directory or subdirectories below this one:

   Windows 2000: `%SIP_HOME%\conf\share\views`

   UNIX: `/opt/OV/SIP/conf/share/views`

   If a portal view file does not yet exist see the “Customizing Portal Views” section of the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`) and follow the procedure for creating a portal view.
3. Search for the following string to find your existing module to edit:

   `classid="topomap"`

   Module instances are wrapped in the `ModuleInstance` element. The `ModuleInstance id` must be unique among all module instances in the portal view file. For information about the `ModuleInstance` element, see the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “PortalView DTD” section.

For example:

```
<ModuleInstance
   classid="com.hp.ov.portal.modules.topomap"
   display="yes"
   help="/OvSipDocs/C/help/NNM/mapsView.html"
   id="module5"
   rollupState="down"
   title="Topology">
```

4. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags. Alternately, copy the contents from this module's default XML file (see "Relevant Files" on page 131):

```
<TopologyMap showStatus="yes" drillDown="no">
   <Submap href="ovw://NNMhostName/mapName/submap"
       filter="yes"
       title="my submap"
       width="#pixels"
       height="#pixels"
   />
   <Submap href="ovw://NNMhostName/mapName/submap"
       filter="yes"
       title="my submap"
       width="#pixels"
       height="#pixels"
   />
</TopologyMap>
```

NOTE If your NNM management station is running in a language other than English, see "Configuring the Topology Module to Access Non-English NNM Data" on page 49.

See the comments in the `OVTopology.dtd` file for more information about the correct XML syntax:

- *Windows 2000:* `%SIP_HOME%\conf\share\views\OVTopology.dtd`
- *UNIX:* `/opt/OV/SIP/conf/share/views/OVTopology.dtd`

5. To change the title of this Topology module instance, change the `title` attribute: `<ModuleInstance title="new title">`.

   To change the title of all Topology modules, change the `title` attribute in the registration file, see “Relevant Files” on page 131.

6. To launch your own help topic from the module’s `[?]` button, insert the `help` attribute into the `<ModuleInstance>`:

   `help="/OVSipDocs/C/help/NNM/`*`topic`*`.html"`

   Replace *`topic`*`.html` with the name of your help file. The `help` attribute allows you to override the default help URL defined in the module registration file. See the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “Adding and Customizing Module Help Topics” section for more information about writing your own online help.

7. Set the `showStatus` attribute. This attribute affects all submaps displayed in this module instance. For example:

   `showStatus="yes"`

   `"yes"` = NNM's current symbol status colors display through the SIP portal view.

   `"no"` = all symbols displayed in SIP assume NNM's administrative-Unmanaged status color (cream color by default).

8. Set the `drillDown` attribute. This attribute affects all submaps displayed in this module instance. For example:

   `drillDown="yes"`

   `"yes"` = all submaps allow drill-down access through the NNM hierarchy. The submaps that can be accessed through drill-down are controlled by the global settings in the `topologyConfig.xml` file. See “Establishing Global Settings for All Topology Modules” on page 129 for more information.

   `"no"` = none of the displayed submaps provide drill-down access.

9. To add a submap to the Topology module, configure the `ovw:`*`//NNMhostName/mapName/submap`* path to the submap. The *`NNMhostName`* must be the fully-qualified hostname as entered in “Establishing Communication Between NNM and SIP” on page 28.

NOTE If you add your submaps “Using the Topology - Edit Page” on page 122, the path to the submap is automatically determined.

10. Set the `filter` attribute for the submap. For example:

    `filter="yes"`

    `"yes"` = the symbols on this NNM submap are filtered according to the settings in the applicable `Management Data` filter for the particular SIP role (Chapter 6, “Segmenting the NNM Data for Your Customers,” on page 135).

    `"no"` = the `Management Data` filter is ignored. This SIP submap displays all symbols as they appear when this submap is displayed on your NNM management station.

11. Set the `title` attribute for the submap to override the submap title within NNM. To use the NNM title, delete this attribute.
12. The default width and height values (600 and 400, respectively) are assigned in the `topologyConfig.xml` file (see “Establishing Global Settings for All Topology Modules” on page 129).

    If you want to override these values, set the number of pixels in the `height="xx"` and/or `width="xx"` attributes.

    If you wish to use the default settings, delete these attributes.

13. Cut and paste the `<Submap>` blocks into the desired display order.
14. To remove a submap from the Topology module, delete the `<Submap>` block.
15. Save the XML file.
16. After you make modifications to this XML file, validate the syntax. See “Validating XML Files” on page 225 for more information.
17. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Displaying a GIF File Instead of an NNM Submap

To display a GIF file through the Topology module, you must directly edit the XML file.

Simply use the following syntax for the `href` path attribute in any `<Submap>` block:

```
<Submap href="http://URLforTheImage" >
```

Follow the directions in “Directly Editing the PortalView.XML File” on page 124 for information about all other settings. If only GIF files are displayed in the Topology module, you can delete the `showStatus`, `drillDown`, and `filter` attributes:

```
<TopologyMap >
   <Submap href="http://URLforTheImage"
       title="my submap"
       width="#pixels"
       height="#pixels"
   />
</TopologyMap>
```

Establishing Global Settings for All Topology Modules

This section focuses on the following file:

Windows 2000:
`%SIP_HOME%\conf\share\modules\topologyConfig.xml`

UNIX: `/opt/OV/SIP/conf/share/modules/topologyConfig.xml`

After you make modifications to this file, you must restart the Tomcat engine. See Appendix A, “Restarting Tomcat,” on page 219 for more information.

topologyConfig.xml/dtd

There are two ways to use the Topology module. You can display submaps from NNM management stations (or collection stations) or you can display GIF files. The `topologyConfig.xml` file sets default settings for both of these (see the comments in the `topologyConfig.dtd` file for more information).

The following attributes control the frequency with which SIP Topology modules request updated information from NNM management stations:

- `numMapRetries`
  SIP starts checking for a specified map at port 3700. If a map is not running, SIP increments the counter and checks on the next port. If a map was running on 3700, but not the desired map, SIP resets the counter to 0 and checks the next port. Once the counter equals `numMapRetries`, SIP quits searching for the requested map.
- `symbolFetchRateInMin`
  Sets the frequency (in minutes) with which SIP contacts NNM management stations to check for changes in symbol registration files and gathers any new symbol GIF images.

The following three attributes control how the submaps look and behave:

- `defaultWidth & Height`
  Sets the dimensions of submaps when the dimension is not specified by the submap element in the Topology module.

- `drillDownWidth & Height`
  Sets the dimensions of submaps accessed through drill-down behavior.
- `loadTransientSubmaps`
  Toggles drill-down access to *transient* submaps (those generated on-demand in NNM), as opposed to only allowing drill-down access to *persistent* submaps (those stored in RAM on the NNM management station) or transient submaps that are currently displayed on the NNM management station.

The following two attributes control filtering for all Topology modules:

- `defaultFilter`
  `"yes"` = the NNM submaps are filtered according to the settings in the applicable `Management Data` filter for the particular customer.

  `"no"` = the `Management Data` filter is ignored, and the SIP submaps include all symbols that appear on the NNM management station.

NOTE It is possible to override the `defaultFilter` setting for a particular submap. See "Directly Editing the PortalView.XML File" on page 124.

- `filterConSymbols`
  `"yes"` = only those interfaces specifically listed in any applicable `InterfaceList` filter element, and that passed through the current SIP Role's assigned `Management Data` filter are displayed as connective lines in the SIP submap.

  `"no"` = connective lines representing interfaces are displayed in SIP if the node to which they are connected passes the current SIP Role's assigned `Management Data` filter, irrespective of any limitations specified in the `InterfaceList`.

  See Chapter 6, "Segmenting the NNM Data for Your Customers," on page 135 for `Management Data` filter information. See also "Filtering Possibilities for the Topology Module" on page 184 for `InterfaceList` filter information.

Relevant Files

The Topology module must follow the rules defined in the following DTD files. See the comments in the DTD files for an explanation of each element used in the XML files:

- `nmConfig.dtd & mgmtStations.dtd & mgmtStations.xml`
  This XML file contains the list of all NNM management stations with which SIP is allowed to communicate. Use the SIP Configuration Editor program to make changes to this file. You must specify whether or not the Topology module is allowed to request data. You must provide information about which ports are being used by NNM processes to communicate topology data on each management station. See "Establishing Communication Between NNM and SIP" on page 28.
- `OVModuleRegistraton.dtd & OVRegTopology.xml`

  This XML file grants access to the Topology module through the SIP framework so that it is available for your use. To add another instance of the Topology module to the SIP module selection list, you copy and rename the `OVRegTopology.xml` and the `OVDefaultTopology.xml` files. Then update the `description`, `title`, `classid`, `help`, and `defaultConfigXML` attribute values in the new registration file.

  If you make any changes to a registration file, you must follow the directions in "Restarting the Servlet Engine" on page 220.
- `OVTopology.dtd & OVDefaultTopology.xml& OVDefaultTopology_ja.xml` (Japanese version)

  This DTD defines the rules for configuring the Topology module. The XML file contains the *default* Topology module. The contents of the default file are inserted into your portal each time you use the `[Add]` button to insert the Topology module.

  You can modify the `OVDefaultTopology.xml` file to meet your needs. Either:

  — Directly edit the XML code in the `OVDefaultTopology.xml` file, or

— Insert a Topology module into any portal. Modify the module to meet your needs. Then, copy the modified XML code for the module from your portal view file, and paste it into the `OVDefaultTopology.xml` file.

See “Directly Editing the PortalView.XML File” on page 124 for more information.

Two default Topology module files are provided: one for use with NNM management stations running in English, the other for use with NNM management stations running in Japanese (see “Running in Languages Other Than English” on page 47 for more information)

- `topologyConfig.dtd & topologyConfig.xml`
  The settings in this XML file affect all Topology modules in all portal views. See “Establishing Global Settings for All Topology Modules” on page 129 and the comments in the `topologyConfig.dtd` file for more information.

- `PortalView.dtd &` *`PortalView`*`.xml`

  This DTD provides the rules for formatting the XML code in your portal view files. See the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “Customizing Portal Views” section for more information about creating portal view files.

- `/htdocs/C/help/NNM/*.html`

  Help topics for Topology modules, accessed by clicking the `[?]` button. If you want to supply your own customized help files, see the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “Adding and Customizing Module Help Topics” section.

Table 5-1 Topology Module Files on the SIP Server

File Name	Windows 2000 Location %SIP_HOME%\....	UNIX Location /opt/OV/SIP/....
mgmtStations.dtd	conf\share\stations\	conf/share/stations/
nmConfig.dtd	conf\share\stations\	conf/share/stations/
mgmtStations.xml	conf\share\stations\	conf/share/stations/
OVModuleRegistration.dtd	registration\	registration/

Table 5-1 Topology Module Files on the SIP Server

File Name	**Windows 2000 Location** **%SIP_HOME%\....**	**UNIX Location** **/opt/OV/SIP/....**
OVRegTopology.xml	registration\	registration/
topologyConfig.dtd	conf\share\modules\topology\	conf/share/modules/topology/
topologyConfig.xml	conf\share\modules\topology\	conf/share/modules/topology/
OVTopology.dtd	conf\share\views\	conf/share/views/
OVDefaultTopology.xml	registration\defaults\	registration/defaults/
OVDefaultTopology_ja.xml	contrib\conf\topology\	contrib/conf/topology/
PortalView.dtd	conf\share\views\	conf/share/views/
PortalView.xml	conf\share\views\	conf/share/views/
*.html	htdocs\C\help\NNM\	htdocs/C/help/NNM/

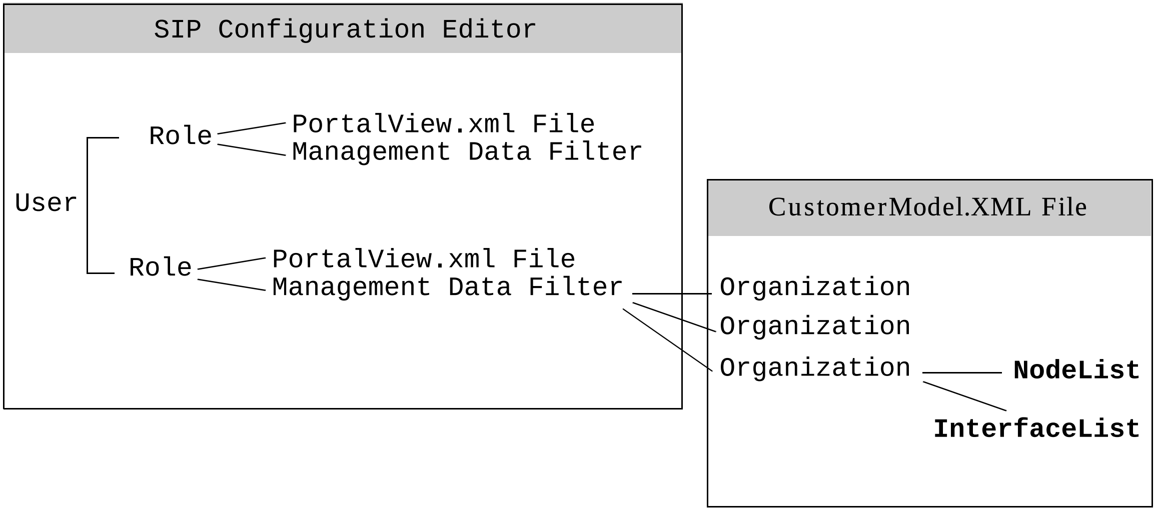
6 Segmenting the NNM Data for Your Customers

Integrating NNM Data into Your Customer Model

The SIP Customer Model allows you to associate resources (nodes and interfaces) with users so that data is automatically filtered appropriately when the user displays any of the NNM modules. Figure 6-1 illustrates how the SIP Customer Model works.

Before you proceed, decide for which groups you need to segment data. For example, you may need to provide portals for several divisions within your company: accounting, marketing, R&D, legal, support. You could assign lists of nodes and/or interfaces to each of these groups. Because of the assigned resource lists, each of these groups could view the same instance of an Alarm module, Network Device Health module, and/or Topology module, yet see only the data appropriate for them.

Figure 6-1 SIP Customer Model



NOTE

With the exception of the Interface Health gauge in the Network Device Health module, this process is *optional*. If you do not wish to use the Interface Health gauge in the Network Device Health module and do not wish to segment data by customer, simply specify `AllData` for the

`Management Data Filter` assigned to particular SIP Roles. See the *SIP Deployment and Integration Guide*, Chapter 6, section “Configuring Users and Roles” (`SIP_Deployment_Integration.pdf`), for more information about how Management Data filters are assigned to particular roles.

The remainder of this chapter explains how to create `<NodeList>` and `<InterfaceList>` elements for use in your `<Organization>` definitions.

If the currently assigned Management Data filter includes:

- Alarms module
  - `<NodeList>`, only those alarms from nodes that pass through the management data filter are displayed.
  - `<InterfaceList>`, are ignored.
- Network Device Health module
  - `<NodeList>`, only those nodes that pass through the management data filter are included in the health calculation.
  - `<InterfaceList>`, for interface-oriented gauges (such as Interface Health), only those interfaces that pass through the management data filter are included in the health calculation.
- Topology
  - `<NodeList>`, only those nodes that pass through the management data filter are displayed on submaps.
  - `<InterfaceList>`, only those interfaces that pass through the management data filter are displayed on submaps (as connection lines).

Windows 2000:

```
%SIP_HOME%\conf\share\organizations\SimpleCustomerModel.dtd
%SIP_HOME%\conf\share\roles\UserRole.dtd
```

UNIX:

```
/opt/OV/SIP/conf/share/organizations/SimpleCustomerModel.dtd
/opt/OV/SIP/conf/share/roles/UserRole.dtd
```

Your `<NodeList>` and `<InterfaceList>` elements can be defined in one or more XML files or a mix of CGIs, servlets, URLs, and files. SIP provides several tools to help simplify the process of defining

`<NodeList>` and `<InterfaceList>` elements by gathering the required data from the NNM object database. This chapter explains multiple options. These approaches are not mutually exclusive and can be used in combination:

- "Manually Creating NodeList and InterfaceList Elements" on page 140

  Write your own XML files that conform to the `SimpleCustomerModel.dtd`. You manually define lists of nodes and interfaces within your NNM management station's network environment.

- "Dynamically Gathering Customer Views Organization Data" on page 147

  This CGI program works only if the OpenView Customer Views program is installed and configured on your NNM management station. If you are using Customer Views, you already defined a customer model and can export that information dynamically for use in the SIP customer model. Use the supplied CGI program to dynamically output the Customer Views customer model data as valid XML.

- "Exporting Customer Views Organization Data to an XML File" on page 151

  This CGI program works only if the OpenView Customer Views program is installed and configured on your NNM management station. If you are using Customer Views, you already defined a customer model. Use the supplied CGI program to perform a one-time migration of your Customer Views customer model to an XML file. Essentially, this gives you a one-time snapshot of the Customer Views customer model to be used as a starting point for your SIP customer model. Use this approach if you want to leverage the NNM Customer Views customer model, but also want the flexibility to make changes.

TIP By using a combination of the provided CGI programs and manually created XML files, you can leverage OpenView Customer Views mappings and expand upon them to include non-IP interfaces (for example, switch ports, not supported by Customer Views).

- "Dynamically Gathering NNM Node and Interface Data" on page 154

  Use the supplied servlet to dynamically generate lists of nodes and interfaces from the NNM object database. Use this approach if you want to retrieve information from the NNM object database (ovwdb) and automatically return XML content for the SIP customer model. The generated `<NodeList>` and `<InterfaceList>` elements are formatted according to the `SimpleCustomerModel.dtd`.

- "Exporting NNM Node and Interface Data to an XML File" on page 159

  Use the supplied servlet to perform a one-time migration of your NNM object database information into XML files. Essentially, this gives you a one-time snapshot of the data from the NNM object database (ovwdb) to be used as a starting point for the SIP customer model definitions. The generated XML file is a partial SIP customer model of `<NodeList>` and `<InterfaceList>` elements formatted according to the `SimpleCustomerModel.dtd`.

- Developing a Custom Customer Model Source

  Create your own program (CGI or servlet) to generate a mapping from an arbitrary data store or provisioning system, and express it in XML that conforms to the `SimpleCustomerModel.dtd`. See the "Developing a Custom Customer Model Source" section in the *SIP Deployment and Integration Guide*, (`SIP_Deployment_Integration.pdf`) for more information.

NOTE An even finer level of data control is available at the module-type or module-instance level. This finer level of control is called *display filtering* and determines what the user *actually* sees in a particular module. For more information, see Chapter 7, "Display Filtering for NNM Modules," on page 165. You can combine the SIP customer model and display filtering to get the desired results in the NNM modules.

Manually Creating NodeList and InterfaceList Elements

The SIP customer model allows you to associate lists of resources with `<Organizations>`. This section explains how to create lists of nodes and lists of interfaces specifically for use with the NNM modules. You can create one XML file that contains all the various `<NodeList>` elements and `<InterfaceList>` elements, or you can create multiple XML files containing the various list elements. Your list elements can be directly inserted into your `<Organization>` definitions, or inserted by reference. For example:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SimpleCustomerModel SYSTEM "SimpleCustomerModel.dtd">

<SimpleCustomerModel>
    <Organization name="Marketing">

        <NodeList name="Your Choice 1">
            <Node name="host.acme.com"      />
            <Node name="server.acme.com"    />
        </NodeList>

        <InterfaceList name="Your Choice 2">
            <Interface name="15.40.10.2" type="ov-ipv4"/>
            <Interface name="35.10.10.2" type="ov-ipv4"/>
        </InterfaceList>

    </Organization>
    <Organization name="Accounting">
       <NodeListRef href=”Your Choice 1“ />
       <InterfaceListRef href="Your Choice 2" />
    </Organization>
</SimpleCustomerModel>
```

Customer-to-Node Mappings

To create <NodeList> elements that can be associated with <Organization> elements in your SIP Customer Model, do the following.

NOTE

The OVO Messages module that communicates with OpenView Operations (OVO) also responds to <NodeList> elements.

See the *OVO and OVSN Integration with SIP* manual (OVO_and_OVSN_Integration.pdf) for more information. If this causes a problem, you can assign two roles to a particular user. Optimize one role's <NodeList> element for the NNM modules and the second role's node <NodeList> element for the OVO Messages module.

On the NNM management station

If you do not already know the fully-qualified hostname or IP address of the nodes you want to use in your <NodeList> elements, do one of the following to gather that information:

- Examine one or more NNM submaps.
- Use NNM's Edit:Find->Object by Attribute feature.
- NNM's Inventory Report (accessed through the NNM Report Presentor).
- Use the NNM ovobjprint command. See the *ovobjprint* reference page in NNM's online help (or the UNIX manpage) for more information for more information.
- Use the ovtopodump command. See the *ovtopodump* reference page in NNM's online help (or the UNIX manpage) for more information.

On the SIP server

You are now ready to create <NodeList> elements. For example:

```
<NodeList name="your choice" >
    <Node name="fully-qualified-hostname" />
    <Node name="IP-address" />
</NodeList>
```

1. Make a backup the *CustomerModel*.xml configuration files before making changes. If you edit the file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.

   Using an ASCII or XML editor, either edit one of your existing *CustomerModel*.xml files or create an XML file based upon the SimpleCustomerModel.dtd. These files are located in the following directory:

   Windows 2000:
   %SIP_HOME%\conf\share\organizations\

   UNIX:
   /opt/OV/SIP/conf/share/organizations/

2. Save your new XML file to the /conf/share/organizations directory. If you place your XML file in any other location, update the path information when you get to step 6 and step 10.

3. Create a <NodeList> element for each group of nodes. Give each node list a name that you can reference later when you are assigning node lists to specific <Organization> definitions:

   ```
   <NodeList name="yourChoice" >
       <Node name="fully-qualified-hostname" />
       <Node name="IP-address" />
   </NodeList>
   ```

4. *Required:* for each node in this <NodeList> element, enter either a fully-qualified hostname or IP address into the name attribute:

   ```
    <Node name="fully-qualified hostname or IP address" />
   ```

5. Insert the following lines at the top of each XML file that you create:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SimpleCustomerModel SYSTEM "SimpleCustomerModel.dtd">

<SimpleCustomerModel>
```

6. The path to the SimpleCustomerModel.dtd in the DOCTYPE statement *must* correctly reference the location of the SimpleCustomerModel.dtd file in the /conf/share/organizations directory on the SIP server. If you place your XML file in any other location, use the DOCTYPE statement in the second example below.

- Example of DTD reference in XML file located in the `organizations` directory:

```
<!DOCTYPE SimpleCustomerModel SYSTEM "SimpleCustomerModel.dtd">
```

- Example of DTD reference in XML file located other than the `organizations` directory (change *`SIPserver.co.com`* to your SIP server's fully-qualified hostname):

```
<!DOCTYPE SimpleCustomerModel PUBLIC "SimpleCustomerModel"
"http://SIPserver.co.com/ovportal/servlet/DTDServer/conf/share/organizations/
SimpleCustomerModel.dtd\">
```

7. Insert the following line at the end of each XML file that you create:

   `</SimpleCustomerModel>`

8. Save the XML file.

9. After you make modifications to XML files, validate the syntax. See "Validating XML Files" on page 225 for more information.

10. Register the source of the customer model data with SIP. For detailed instructions, see "Registering SIP Customer Model Sources" on page 162.

11. You are now ready to associate these `<NodeList>` elements with `<Organization>` elements in your Customer Model, as appropriate. See the "Mapping Organizations to Their Resources" section in the *SIP Deployment and Integration Guide*, (`SIP_Deployment_Integration.pdf`) for more information.

    See example on page 140.

Customer-to-Interface Mappings

To create `<InterfaceList>` elements that can be associated with `<Organization>` elements in your SIP Customer Model, do the following.

On the NNM management station

For each interface, you need to gather either:

- IP-address, or
- `hostname/ifAlias::ifDescr`

  `hostname` = fully-qualified hostname or IP address

  `ifAlias` = from the SNMP IF-MIB (rfc2863)

  `ifDescr` = the first word in SNMP MIB-II (rfc1213) `ifDescr` string

  NOTE: Either `ifAlias` or `ifDescr` can be empty, but the combination must uniquely identify the interface.

To determine the required information for your `<InterfaceList>` elements, use one of the following:

- Examine one or more NNM submaps. Right-click on an interface symbol, and select `Interface Properties`. The address, `ifAlias`, and `ifDescr` values are displayed.
- Use NNM's `Edit:Find->Object by Attribute` feature.
- NNM's Inventory Report (accessed through the NNM Report Presentor).
- Use the NNM `ovobjprint` command. See the *ovobjprint* reference page in NNM's online help (or the UNIX manpage) for more information.
- Use the `ovtopodump` command. See the *ovtopodump* reference page in NNM's online help (or the UNIX manpage) for more information.

On the SIP server

You are now ready to create `<InterfaceList>` elements. For example:

```
<InterfaceList name="your choice" >
    <Interface
          name="IP-address"
          type="ov-ipv4" />
    <Interface
          name="hostname/ifAlias::1st_word_of_ifDescr"
          type="ov-ifv4" />
</InterfaceList>
```

1. Make a backup the *CustomerModel*.xml configuration files before making changes. If you edit the file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.

   Using an ASCII or XML editor, either edit one of your existing *CustomerModel*.xml files or create an XML file based upon the SimpleCustomerModel.dtd. These files are located in the following directory:

   Windows 2000:
   %SIP_HOME%\conf\share\organizations\

   UNIX:
   /opt/OV/SIP/conf/share/organizations/

2. Save your new XML file to the /conf/share/organizations directory on the SIP server. If you place your XML file in any other location, update the path information when you get to step 6 and step 10.

3. Create an <InterfaceList> element for each group of interfaces. Give each interface list a name that you can reference later when you are assigning interface lists to specific <Organization> definitions:

   ```
   <InterfaceList name="yourChoice" >
       <Interface name="IP-address" />
       <Interface name="IP-address" />
   </InterfaceList>
   ```

4. *Required:* for each interface that needs to be included in this <InterfaceList> element, enter an ov-ipv4 or ov-ifv4 address into the name attribute:

   ```
       <Interface
             name="IP-address"
             type="ov-ipv4" />
       <Interface
             name="hostname/ifAlias::1st_word_of_ifDescr"
             type="ov-ifv4" />
   ```

5. *Required only for ov-ifv4:* add the type attribute to each ov-ifv4 interface specification:

   ```
    <Interface name="IP-address" type="ov-ifv4" />
   ```

   This attribute is also displayed through the Managed Resources module. For information about the Managed Resources module, see “Configuring the Managed Resources Module” in the *SIP Deployment and Integration Guide* (SIP_Deployment_Integration.pdf).

6. Insert the following lines at the top of each XML file that you create:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SimpleCustomerModel SYSTEM "SimpleCustomerModel.dtd">

<SimpleCustomerModel>
```

7. The path to the `SimpleCustomerModel.dtd` in the `DOCTYPE` statement *must* correctly reference the location of the `SimpleCustomerModel.dtd` file in the `/conf/share/organizations` directory. If you place your XML file in any other location, use the `DOCTYPE` statement in the second example below.

   - Example of DTD reference in XML file located in the `organizations` directory:

```
<!DOCTYPE SimpleCustomerModel SYSTEM "SimpleCustomerModel.dtd">
```

   - Example of DTD reference in XML file located other than the `organizations` directory (change *`SIPserver.co.com`* to your SIP server's fully-qualified hostname):

```
<!DOCTYPE SimpleCustomerModel PUBLIC "SimpleCustomerModel"
"http://SIPserver.co.com/ovportal/servlet/DTDServer/conf/share/organizations/
SimpleCustomerModel.dtd\">
```

8. Insert the following line at the end of each XML file that you create:

   ```
   </SimpleCustomerModel>
   ```

9. Save the XML file.

10. After you make modifications to XML files, validate the syntax. See "Validating XML Files" on page 225 for more information.

11. Register the source of the customer model data with SIP. For detailed instructions, see "Registering SIP Customer Model Sources" on page 162.

12. You are now ready to associate these `<InterfaceList>` elements with `<Organization>` elements in your Customer Model, as appropriate. See the "Mapping Organizations to Their Resources" section in the *SIP Deployment and Integration Guide,* (`SIP_Deployment_Integration.pdf`) for more information.

    See example on page 140.

Dynamically Gathering Customer Views Organization Data

Through use of a supplied CGI program, you can dynamically gather customer model data from one or more remote Customer Views servers. This program—`getcvdata.exe`—returns a complete SIP customer model. That is, the data in the Customer Views database is exported to XML mappings that are complete and conform to the SIP `SimpleCustomerModel.dtd.`

Configure NNM Customer Views

NOTE Customer Views running on an NNM *6.1* server:

If you have not already done so, you must carefully follow the directions in "Establishing Communication Between NNM and SIP" on page 28 in order to install the `getcvdata.exe` program onto the NNM management station. This program is automatically installed with NNM 6.2 or greater.

The `getcvdata.exe` program is called from your SIP server. The `getcvdata.exe` program gathers information from your Customer Views database (assuming that NNM Customer Views is configured and running on your NNM management station) and provides SIP with an XML file of `<Organization>`s and their associated `<NodeList>` and `<InterfaceList>` elements in the format required by SIP.

In the context of Customer Views, the term "organization" refers to the organizations you defined within NNM Customer Views and includes both "customers" and "providers."

If your Customer Views program is configured for your "customers" and "providers," go to "Configure SIP to Use getcvdata.exe" on page 149.

If you have not already configured Customer Views, perform the four tasks listed below. The commands for doing so are described in Table 6-1 on page 148. For detailed information, see the documentation that comes with Customer Views.

1. On the NNM management station, start Customer Views.
2. Create Organizations.
3. Associate Nodes with Organizations.
4. Associate Interfaces with Organizations.

The `ovcustomer` command can be run interactively or in batch mode. To run the `ovcustomer` command in interactive mode, run the `ovcustomer` command and then enter specific commands at the “`ovcustomer>`” prompt. Shown below are the relevant `ovcustomer` commands:

Table 6-1 **`ovcustomer` Commands**

Action	Command
Create a new organization.	`ovcustomer>`**`create_org`** ***`<organizationType>`*** ***`<organizationName>`*** “customer” and “provider” are supported values for *organizationType*. An organization name with spaces should be placed in quotes (for example, `"My Customer"`).
Print the list of organizations.	`ovcustomer>`**`print_org`**
Associate a node with an organization.	`ovcustomer>`**`add_associations_to_org`** ***`<organizationName>`*** ***`<Hostname>`***
Print the nodes associated with a specific organization.	`ovcustomer>`**`print_associated_node`** ***`<organizationName>`***
Associate an interface with an organization.	`ovcustomer>`**`add_associations_to_org <organizationName> <IPaddress>`**
Print the interfaces associated with a specific organization.	`ovcustomer>`**`print_associated_interface <organizationName>`**

Configure SIP to Use getcvdata.exe

1. On the SIP server, register `getcvdata.exe` as a SIP Customer Model source. Use the following string. If you add the `?Organization=orgName` string, you can query Customer Views for the data about one particular organization:

   - If your NNM management station is running on *Windows:*

     ```
     http://NNMHostname/OvCgi/getcvdata.exe
     http://NNMHostname/OvCgi/getcvdata.exe"?Organization=orgName
     ```

   - If your NNM management station is running on *UNIX:*

```
http://NNMHostname:8880/OvCgi/getcvdata.exe
http://NNMHostname/:8880/OvCgi/getcvdata.exe"?Organization=orgName
```

   See “Registering SIP Customer Model Sources” on page 162.

NOTE When collecting data from Customer Views that is running in a language other than English, set the locale using the `AcceptLang` and `&Developer` CGI parameters (both are required):

```
http://host/OvCgi/getcvdata.exe?AcceptLang=ja&Developer
```

The value of `AcceptLang` is a Web locale, NNM converts the web local to an operating system locale using the locale mapping table in: `NNM_install_directory/www/conf/locales.mapping`.

For example, the `AcceptLang` value “`ja`” translates into the locale “`ja_JP.SJIS`”

`getcvdata.exe` supports the following parameters:

Organization — Generates information for only one specified Customer Views organization.

`Organization="orgname"`

`OrgList` — Generates a list of all the organizations and their attributes: `name`, `type`, and `ExternalKey`. Not including child information (nodes, interfaces, and services).

`?AcceptLang` and `&Developer` — When Customer Views is running in a language other than English.

null	When no attributes are specified, `getcvdata` returns all information for all the customers in the Customer Views database.

2. If you configure SIP to call `getcvdata.exe` from multiple Customer Views servers, register each one as a customer model source.

3. To verify that `getcvdata.exe` is working as expected, in the `SIP Administration Pages`, on the `Customer Model` tab, select the data source that you just registered, and click `[Report]`.

4. You are now ready to associate these `<Organization>` elements in your Customer Model with Management Data filters for specific SIP Roles, as appropriate. See the “Configuring Users and Roles” section in the *SIP Deployment and Integration Guide*, (`SIP_Deployment_Integration.pdf`) for more information.

   See example on page 140.

Exporting Customer Views Organization Data to an XML File

Through use of a supplied CGI program, you can gather customer model data for SIP's use from one or more remote Customer Views servers.

The program—`getcvdata.exe`—generates an XML file of the data from each Customer Views database. Each XML file is a complete SIP customer model mapping that conforms to the `SimpleCustomerModel.dtd`.

On the NNM Management Station

NOTE Customer Views running on an NNM *6.1* server:

If you have not already done so, you must carefully follow the directions in "Establishing Communication Between NNM and SIP" on page 28 in order to install the `getcvdata.exe` program onto the NNM management server. This program is automatically installed with NNM 6.2 or greater.

The `getcvdata.exe` program gathers information from your Customer Views database (assuming that NNM Customer Views is configured and running on your NNM management station) and generates an XML file of organizations and their associated nodes and interfaces in the format required by SIP.

In the context of Customer Views, the term "organization" refers to the organizations you defined within NNM Customer Views and includes both "customers" and "providers."

If you have not already configured Customer Views, do so before proceeding (see "Configure NNM Customer Views" on page 147). For detailed information, see the documentation that comes with Customer Views.

To generate an XML file of the information in your Customer Views database, on the NNM management station:

1. At the command prompt, type:

   Windows NT/2000:

```
<NNM_install_dir>\www\cgi-bin\getcvdata.exe > C:\temp\uniqueFileName.xml
```

   UNIX:

```
/opt/OV/www/cgi-bin/getcvdata.exe > C:/temp/uniqueFileName.xml
```

   This returns all information for all the customers known by the Customer Views server.

NOTE When collecting data from Customer Views that is running in a language other than English, make sure that you save the new XML file in the UTF-8 codeset before placing the file on the SIP server. See "Running in Languages Other Than English" on page 47 for more information.

2. Now, move the *uniqueFileName*.xml file or files that you just created over to the SIP server. Place these XML files in the following location:

   Windows 2000: `%SIP_HOME%\conf\share\organizations\`
   UNIX: `/opt/OV/SIP/conf/share/organizations/`

NOTE You may create subdirectories to contain your XML file or files.

On the SIP Server

1. Open each *uniqueFileName*.xml file that you created in the previous section.

   The first line within each generated XML file is a reference to the location of the `SimpleCustomerModel.dtd` file. Your XML file is not valid unless the path to its governing DTD file is correctly listed at the top of the file. Enter the appropriate information as explained in the following examples:

- Example of DTD reference in XML file located in the `organizations` directory:

```
<!DOCTYPE SimpleCustomerModel SYSTEM "SimpleCustomerModel.dtd">
```

- Example of DTD reference in XML file located other than the `organizations` directory (change *`SIPserver.co.com`* to your SIP server's fully-qualified hostname):

```
<!DOCTYPE SimpleCustomerModel PUBLIC "SimpleCustomerModel"
"http://SIPserver.co.com/ovportal/servlet/DTDServer/conf/share/organizations/
SimpleCustomerModel.dtd\">
```

2. After you make modifications to XML files, validate the syntax. See "Validating XML Files" on page 225 for more information.

3. Register the XML file as a Customer Model Source with SIP. See "Registering SIP Customer Model Sources" on page 162.

4. You are now ready to associate these `<Organization>` elements in your Customer Model with Management Data filters for specific SIP Roles, as appropriate. See the "Configuring Users and Roles" section in the *SIP Deployment and Integration Guide*, (`SIP_Deployment_Integration.pdf`) for more information.

   See example on page 140.

Dynamically Gathering NNM Node and Interface Data

Through use of a supplied servlet (`NNMSimpleCustomerModel`), you can dynamically gather node and interface lists from the NNM object database (`ovwdb`) of one or more NNM management stations. The data generated by `NNMSimpleCustomerModel` is a partial customer model formatted according to the `SimpleCustomerModel.dtd`. Essentially, the generated XML content consists of `<NodeList>` and `<InterfaceList>` elements that can be mapped to the `<Organization>`s defined in your SIP customer model files.

These lists of nodes and interfaces are generated according to a schedule that you control, and are stored in memory on the SIP server. You simply place pointers to these lists (by name) in your `<Organization>` definitions.

Configure SIP to Use the NNMSimpleCustomerModel Servlet

On the SIP server:

1. Open the `NNMData.xml` file located in the following directory:

   Windows 2000: `%SIP_HOME%\conf\share\modules\NM\`
   UNIX: `/opt/OV/SIP/conf/share/modules/NM/`

   You must modify this XML code to meet your needs (for more information, see the `NNMData.dtd` in this same directory). Default contents are provided as a starting point:

```
<NNMSimpleCustomerModel>
  <NNMNodeList name="List1">
    <NNMStation hostname="localhost" ovwdbPort="9999" />
      <NodeSelection>
        <IPHostFilter>
          <IPHost hostname=".*" />
        </IPHostFilter>
      </NodeSelection>
  </NNMNodeList>
  <NNMInterfaceList name="List2">
    <NNMStation hostname="localhost" ovwdbPort="9999" />
      <InterfaceSelection>
         <IPInterfaceFilter>
           <IPInterface ipAddr=".*" />
         </IPInterfaceFilter>
      </InterfaceSelection>
    </NNMInterfaceList>
</NNMSimpleCustomerModel>
```

2. Enter a name for the `NNMNodeList` you wish to generate. The `NNMNodeList` element defines how nodes are grouped into specific `<NodeList>` elements for use in the SIP customer model. For example:

   ```
   <NNMNodeList name="AccountingDepartment">
   ```

3. Enter the NNM management station's `hostname` as entered into through SIP Configuration Editor.

   Enter the `ovwdbPort` that SIP is configured to use when communicating with this NNM management station (see "Establishing Communication Between NNM and SIP" on page 28 for more information). For example:

   ```
   <NNMStation hostname="mountain.rm.cnd.com"
   ovwdbPort="2447" />
   ```

4. Enter the desired query specifications into the `NNMData.xml` file.

   Each query is configured as XML and serves as a filter that is applied to the NNM object database (`ovwdb`). The filter specifications define which nodes and interfaces are returned.

   Decide how your filter is defined. You can use any combination of two kinds of filtering:

- **Perl regular expression-based filtering**. Use Perl5 regular expressions within <IPHostFilter> or <IPInterfaceFilter> elements.

NOTE When writing filters, use Perl5 regular expressions. For example: `.*\.eagle\.wingnuts\.com`

See `www.perl.com` or `www.perldoc.com` for information about Perl5 regular expressions.

- **Capability filtering**. The `CapabilityFilter` commonly refers to the NNM capability filters such as isRouter, isNode, etc. However, `CapabilityFilter` can utilize *any* NNM object database field within Network Node Manager's object database (ovwdb).

  Be aware that an empty `CapabilityFilter` yields the empty set which allows nothing to pass.

TIP On the NNM management station, to generate a complete list of the currently defined ovwdb *fields*, at the command prompt, type:

```
ovobjprint -f > filename
```

To identify the valid *values* for a particular field, at the command prompt, type:

```
ovobjprint -a "field_name" > filename
```

See the `ovobjprint` reference page in NNM's online help (or the UNIX manpage) for more information.

5. Input all the `<NodeSelection>` elements (that are output as `<NodeList>` elements) that you wish to refer to from your SIP customer model's `<Organization>` definitions. The returned `<NodeList>` elements are formatted according to the `CustomerModel.dtd` and ready to reference within the SIP customer model. You have two choices: `<IPHostFilter>` or `<CapabilityFilter>`. See the `NNMdata.dtd` file for more information.

   Example one `<IPHostFilter>`, filters upon hostname. Use Perl5 regular expressions:

```
<NodeSelection>
   <IPHostFilter op="OR">
     <IPHost hostname=".*\.customer1\.com"/>
     <IPHost hostname=".*\.customer2\.com"/>
   <IPHostFilter>
</NodeSelection>
```

Example two `<CapabilityFilter>` can utilize *any* field within Network Node Manager's object database (`ovwdb`):

```
<NodeSelection>
   <CapabilityFilter op="OR">
     <Capability field="IPStatus" value="Critical"/>
     <Capability value="isServer"/>
    <Capability field="vendor" value="Hewlett-Packard" />
</CapabilityFilter>
</NodeSelection>
```

6. Enter a name for the `NNMInterfaceList` you wish to generate. The `NNMInterfaceList` element defines how interfaces are grouped into specific `<InterfaceList>` elements for use in the SIP customer model. For example:

   ```
   <NNMInterfaceList name="AccountingDepartment">
   ```

7. Enter the NNM management station's `hostname` as entered into through SIP Configuration Editor.

   Enter the `ovwdbPort` that SIP is configured to use when communicating with this NNM management station (see "Establishing Communication Between NNM and SIP" on page 28 for more information). For example:

   ```
   <NNMStation hostname="mountain.rm.cnd.com"
   ovwdbPort="2447" />
   ```

8. Input all the `<InterfaceSelection>` elements (that are output as `<InterfaceList>` elements) that you wish to refer to from your SIP customer model's `<Organization>` definitions. The returned `<InterfaceList>` elements are formatted according to the `CustomerModel.dtd` and ready to reference within the SIP customer model. You have one choice: `<IPInterfaceFilter>`. See the `NNMdata.dtd` file for more information.

   Example `<IPInterfaceFilter>`, filters upon `IPAddress`. Use Perl5 regular expressions:

```
<InterfaceSelection>
   <IPInterfaceFilter op="OR">
     <IPInterface ipAddr=".*\.112\.*"/>
   <IPInterfaceFilter>
</InterfaceSelection>
```

NOTE The only way to add non-IP addresses to an <InterfaceList> is to add them manually to the XML file. “Manually Creating NodeList and InterfaceList Elements” on page 140.

9. To establish the dynamically generated <NodeList> elements and <InterfaceList> elements into the SIP customer model, continue to the next step.

   To export your <NodeList> elements and <InterfaceList> elements into an XML file for use in the SIP customer model, go to “Exporting NNM Node and Interface Data to an XML File” on page 159.

10. On the SIP server, register the new SIP Customer Model source. Use the following string:

    - If your NNM management station is running on *Windows:*

      `http://SIPhostname/ovportal/NNMSimpleCustomerModel`

    - If your NNM management station is running on *UNIX:*

      `http://SIPhostname:8880/ovportal/NNMSimpleCustomerModel`

    See “Registering SIP Customer Model Sources” on page 162.

11. To verify that `getcvdata.exe` is working as expected, in the `SIP Administration Pages`, on the `Customer Model` tab, select the data source that you just registered, and click `[Report]`.

12. You are now ready to associate these <NodeList> elements and <InterfaceList> elements with <Organization> elements in your Customer Model, as appropriate. See the “Mapping Organizations to Their Resources” section in the *SIP Deployment and Integration Guide*, (`SIP_Deployment_Integration.pdf`) for more information.

    See example on page 140.

Exporting NNM Node and Interface Data to an XML File

The supplied `NNMSimpleCustomerModel` servlet can extract data from the NNM object database (ovwdb) of one or more NNM management stations and generate `<NodeList>` and `<InterfaceList>` elements formatted according to the `SimpleCustomerModel.dtd`. Essentially, the generated XML content consists of `<NodeList>` and `<InterfaceList>` elements that can be mapped to the `<Organization>` elements defined in your SIP customer model.

Generate the XML File

To configure the `NNMSimpleCustomerModel` servlet, follow the directions in “Dynamically Gathering NNM Node and Interface Data” on page 154. Then, return to this section.

Follow these steps to generate and display the lists provided by the `NNMSimpleCustomerModel` servlet. The time required to generate the lists depends upon the size of the object database on the NNM management station. Once the list is generated, you will save it as an XML file on the SIP server.

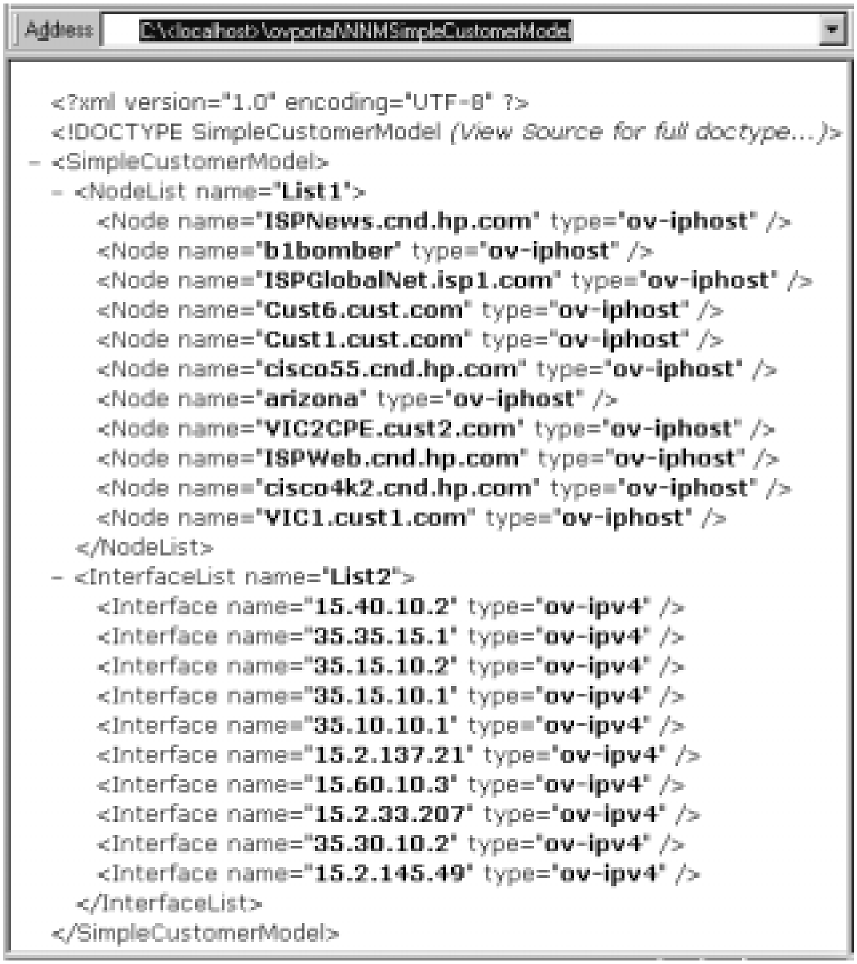
1. Open a browser window on the SIP server and type:

   http://*localhost*/ovportal

2. Type the following to generate and display the lists and verify that the results meet your expectations and requirements. For this tool to work, it must run locally, not remotely:

   http://*localhost*/ovportal/NNMSimpleCustomerModel

3. Examine the XML output and verify that the results meet your expectations and requirements. The output should look something like the following example:



Address C:\<localhost>\ovportal\NNMSimpleCustomerModel

```
  <?xml version="1.0" encoding="UTF-8" ?>
  <!DOCTYPE SimpleCustomerModel (View Source for full doctype...)>
- <SimpleCustomerModel>
  - <NodeList name="List1">
      <Node name="ISPNews.cnd.hp.com" type="ov-iphost" />
      <Node name="b1bomber" type="ov-iphost" />
      <Node name="ISPGlobalNet.isp1.com" type="ov-iphost" />
      <Node name="Cust6.cust.com" type="ov-iphost" />
      <Node name="Cust1.cust.com" type="ov-iphost" />
      <Node name="cisco55.cnd.hp.com" type="ov-iphost" />
      <Node name="arizona" type="ov-iphost" />
      <Node name="VIC2CPE.cust2.com" type="ov-iphost" />
      <Node name="ISPWeb.cnd.hp.com" type="ov-iphost" />
      <Node name="cisco4k2.cnd.hp.com" type="ov-iphost" />
      <Node name="VIC1.cust1.com" type="ov-iphost" />
    </NodeList>
  - <InterfaceList name="List2">
      <Interface name="15.40.10.2" type="ov-ipv4" />
      <Interface name="35.35.15.1" type="ov-ipv4" />
      <Interface name="35.15.10.2" type="ov-ipv4" />
      <Interface name="35.15.10.1" type="ov-ipv4" />
      <Interface name="35.10.10.1" type="ov-ipv4" />
      <Interface name="15.2.137.21" type="ov-ipv4" />
      <Interface name="15.60.10.3" type="ov-ipv4" />
      <Interface name="15.2.33.207" type="ov-ipv4" />
      <Interface name="35.30.10.2" type="ov-ipv4" />
      <Interface name="15.2.145.49" type="ov-ipv4" />
    </InterfaceList>
  </SimpleCustomerModel>
```

4. Now, save this XML file to the SIP server. You may place this XML file anywhere that you wish. However, it is recommended that you place the file in the following location:

   Windows 2000: `%SIP_HOME%\conf\share\organizations\`
   UNIX: `/opt/OV/SIP/conf/share/organizations/`

   When you save the output of the `NNMSimpleCustomerModel` program as an XML file, the applicable DTD is automatically inserted into the file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE SimpleCustomerModel [
<!ELEMENT SimpleCustomerModel
(NodeList|InterfaceList|Error| Warning)*>
<!ELEMENT NodeList (Node*)>
<!ATTLIST NodeList name CDATA #IMPLIED>
<!ELEMENT Node EMPTY>
<!ATTLIST Node type CDATA "ov-iphost" name CDATA #REQUIRED>
<!ELEMENT InterfaceList (Interface*)>
<!ATTLIST InterfaceList name CDATA #IMPLIED>
<!ELEMENT Interface EMPTY>
<!ATTLIST Interface type CDATA "ov-ipv4" name CDATA
#REQUIRED>
<!ELEMENT Error EMPTY>
<!ATTLIST Error msg CDATA #REQUIRED>
<!ELEMENT Warning EMPTY>
<!ATTLIST Warning msg CDATA #REQUIRED>
]>
```

5. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 225 for more information.

6. Register this file as a source for customer model data with SIP. For detailed instructions, see “Registering SIP Customer Model Sources” on page 162.

7. You are now ready to associate these `<NodeList>` elements and `<InterfaceList>` elements with `<Organization>` elements in your Customer Model, as appropriate. See the “Mapping Organizations to Their Resources” section in the *SIP Deployment and Integration Guide*, (`SIP_Deployment_Integration.pdf`) for more information.

   See example on page 140.

Registering SIP Customer Model Sources

You need to register with SIP the name and location of the sources of the customer model data. This is done through the `SIP Administration Pages` that are accessed in the SIP portal through a special `SIP Administrator` role.

Registering a Customer Model Source

Customer model sources can be defined in one or more files on the SIP server or can be programmatically generated by CGI programs and servlets. Most likely, the sources to your fully-integrated customer model are a mix of files and programs.

1. Go to the `SIP Administration Pages` by logging in as a user who can access the special `SIP Administrator` role. Switch to the `SIP Administrator` role.
2. Click the `Customer Model` tab.
3. In the `Customer Model Configuration` segment, go to `Customer Model Sources`.
4. In the `New customer model source URL` field, type a relative file name (relative to the `conf/share/organizations` directory), an absolute file name, or a URL. See examples below:

   - Examples of a Relative File Name

     ```
     CustomerModel.xml
     ../uniqueFilename.xml
     ```

     If you specify a relative file path (e.g., "`../CustomerModel.xml`"), it is interpreted relative to the `conf/share/organizations` directory. If the `/organizations` directory is actually remote (that is, the `SIP_CONF_SHARE_DIR` in `SIPPath.properties` is remote), the same syntax is used to specify location relative to the `organizations` directory.

   - Example of an Absolute File Name

     `c:/temp/`*`CustomerModel`*`.xml`

- Examples of URLs

  Calling Windows systems:

  ```
  http://othermachine/CustomerModel.xml
  http://NNMHostname/OvCgi/getcvdata.exe
  http://SIPhostname/ovportal/NNMSimpleCustomerModel
  ```

  Calling UNIX systems:

  ```
  http://NNMHostname:8880/OvCgi/getcvdata.exe
  ```

5. Click `[Add]` to add the name of the customer model source to the `Customer Model Sources` list. The customer model is refreshed automatically, so you do not need to force a refresh by clicking `[Refresh]`.
6. *Optional:* To periodically update the customer model according to a schedule (in addition to each time that SIP's `tomcat` service is restarted), set the `Refresh Rates`.
7. Click `[Help]` for more information.

NOTE After registering your customer model sources, you can create management data filters that reference one or more `<Organization>` elements in the customer model. See the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`) “Defining/Modifying Management Data Filters”.

After you have created the management data filters, you can associate a filter with each role. See the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “Assigning a Management Data Filter to a Role.” At that point you can display segmented management data through SIP modules.

Unregistering a Customer Model Source

1. Go to the `SIP Administration Pages` by logging in as a user who can access the special `Administrator` role. Switch to the `Administrator` role.
2. Click the `Customer Model` tab.
3. In the `Customer Model Configuration` segment, go to `Customer Model Sources`.
4. In the `Customer Model Sources` list, select a customer model source and click `[Delete]`.

7 Display Filtering for NNM Modules

Introduction to Display Filtering

Whereas the `Management Data` filter determines what is *possible* to see in the NNM modules (explained in Chapter 6), display filters control what is actually visible in a particular module instance. Display filters provide a method of restricting the data one last time before the module is visible. The NNM modules use two types of display filters.

To produce a list of nodes for a `NodeSelection` filter, three child filters are available: `IPHostFilter`, `CapabilityFilter`, and `OrganizationFilter`.

To produce a list of interfaces for an `InterfaceSelection` filter (such as specifying a list of certain interfaces within a router), two child filters are available: `IPInterfaceFilter` and `OrganizationFilter`.

- The **Alarms module** applies display filtering at an alarm category level through a `NodeSelection` filter. The Alarms module does not allow the `InterfaceSelection` filter (for example, you cannot request only those alarms from individual interfaces within a router). Because the display filter is applied at the alarm-category level, the display filter affects all Alarms module instances displaying that particular alarm category. If necessary, you can create multiple *sets* of alarm category files to meet different display filtering needs. To create a display filter for an alarm category, you must directly edit the alarm category's XML file. See page 167.
- The **Network Device Health module** requires display filtering as an element of each gauge's configuration. Either a `NodeSelection` or `InterfaceSelection` filter is used. To modify the display filter for a gauge, you must directly edit the XML file. Each display filter affects only one gauge. See page 176.
- For the **Topology module**, no display filtering is available. However, settings are provided to *bypass* the `Management Data` filter on a global basis (all submaps in all Topology modules) or *ignore* `Management Data` filters by individual submap. See page 184.

See also the following DTD:

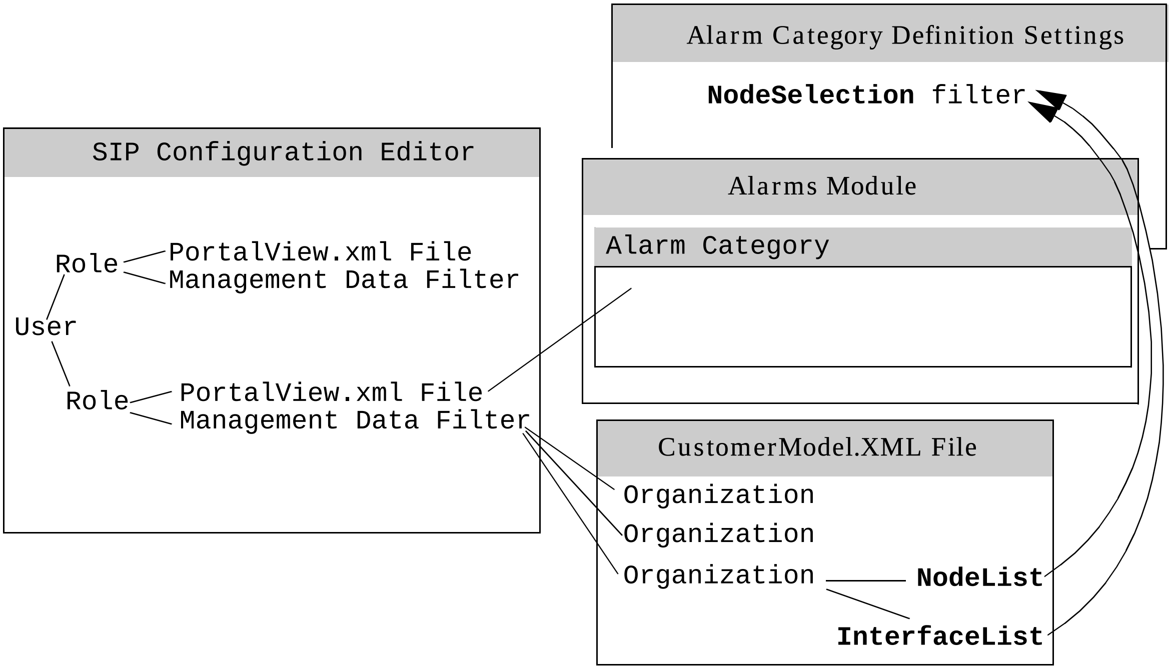
Windows 2000: `%SIP_HOME%\conf\share\views\filter.dtd`
UNIX: `/opt/OV/SIP/conf/share/views/filter.dtd`

Filtering Possibilities for the Alarms Module

Data displayed in the Alarms module must pass the `Management Data` filter assigned to the current SIP Role. See Chapter 6, “Segmenting the NNM Data for Your Customers,” on page 135, and the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “Creating Users and Roles” section, for information about the `Management Data` filter and the SIP User Role Model.

In addition to the filtering you applied through the `Management Data` filter, a finer, second level of filtering is available by applying a `NodeSelection` filter to each alarm category. A variety of other filtering attributes and elements are included in each alarm category definition (“Additional Filtering Attributes and Elements” on page 174).

Figure 7-1 Alarm Module’s “Display Filtering”



NodeFilters for Alarm Categories

Overview

The NodeSelection filter is defined for individual alarm categories to further restrict the set of nodes whose alarms are displayed. The NodeSelection filter has three child elements:

- IPHostFilter
  Passes only alarms from devices specified by hostname or IP address. For example:

  ```
  <NodeSelection>
     <IPHostFilter>
        <IPHost hostname="10\.2\.5\.125"/>
        <IPHost hostname="eagle\.wingnuts\.com"/>
  </NodeSelection>
  ```

  NOTE: when writing filters, using Perl5 regular expressions, only .* is allowed. This example allows all nodes ending in ".eagle.wingnuts.com" to pass:
  .*\.eagle\.wingnuts\.com

  See www.perl.com or www.perldoc.com for information about Perl5 regular expressions

- CapabilityFilter
  The CapabilityFilter commonly refers to the capability field within the NNM database, with values such as isRouter or isNode. However, this filter can search upon any field in the NNM object database. For example:

  ```
  <NodeSelection>
     <CapabilityFilter>
         <Capability field="IPStatus" value="Critical"/>
         <Capability value="isServer"/>
     </CapabilityFilter>
  </NodeSelection>
  ```

- OrganizationFilter
  Passes only alarms from devices included in the specified User Role Model's *organization* element. For example

  ```
  <NodeSelection>
     <OrganizationFilter>
         <OrganizationRef href="OrganizationOne" />
         <OrganizationRef href="OrganizationTwo" />
  ```

```
      </OrganizationFilter>
   </NodeSelection>
```

The actual nodes whose alarms are displayed result from the *intersection* of nodes between the Management Data filter and the NodeSelection filter. It is possible at runtime for the *intersection* of these lists to be the empty set. In this case, no nodes are selected and no output occurs for this alarm category.

Leaving the NodeSelection filter empty results in all nodes that pass the Management Data filter passing the NodeSelection filter when determining the intersection of candidate nodes. For example, the following NodeSelection filter allows all nodes to pass:

```
<NodeSelection>
</NodeSelection>
```

The following NodeSelection filter example passes all nodes that meet *all three* criteria:

- Any node specifically listed in the IPHostFilter,
- That has the isServer capability or is currently in a critical state (the CapabilityFilter commonly refers to the capability field within the NNM object database, ovwdb, with values such as isRouter or isNode. However, this filter can search upon any field in the NNM database),
- And node that is included in OrganizationOne or OrganizationTwo definition.

```
<NodeSelection>
   <IPHostFilter>
      <IPHost hostname="10\.2\.5\.125"/>
      <IPHost hostname="10\.2\.6\.254"/>
      <IPHost hostname="eagle\.wingnuts\.com"/>
      <IPHost hostname="hawk\.wingnuts\.com"/>
   </IPHostFilter>
   <CapabilityFilter>
       <Capability field="IPStatus" value="Critical"/>
       <Capability value="isServer"/>
   </CapabilityFilter>
   <OrganizationFilter>
       <OrganizationRef href="OrganizationOne" />
       <OrganizationRef href="OrganizationTwo" />
   </OrganizationFilter>
</NodeSelection>
```

NOTE When you specify more than one filter (`IPHostFilter`, `CapabilityFilter`, or `OrganizationFilter`), they are AND’d together. An empty `IPHostFilter`, `CapabilityFilter`, or `OrganizationFilter` filter yields the empty set, which allows nothing to pass.

Writing a NodeSelection Filter for an Alarm Category

To create a `<NodeSelection>` filter, you must edit the Alarm Category’s XML file:

1. Make a backup of XML configuration files before you customize them. If you edit the file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.
2. Before you start, access the following two files as a reference:
   - *Windows 2000* :`%SIP_HOME%\conf\share\view\filter.dtd`
     `%SIP_HOME%\conf\share\modules\alarms\NmAlarmCat.dtd`
   - *UNIX:*

     ```
     /opt/OV/SIP/conf/view/filter.dtd
     /opt/OV/SIP/conf/share/modules/alarms/NmAlarmCat.dtd
     ```
3. In an ASCII or XML editor, open the *`NmAlarmCat`*`.xml` file that defines the alarm category you wish to modify. These definition files must be stored in the following location:
   - *Windows 2000:* `%SIP_HOME%\conf\share\modules\alarms\`
   - *UNIX:* `/opt/OV/SIP/conf/share/modules/alarms/`

   The file will look something like the following:

```
<?xml version='1.0' encoding='utf-8' standalone='no' ?>
<!DOCTYPE AlarmCategoryDef SYSTEM "NmAlarmCat.dtd">

<AlarmCategoryDef
     DisplayTitle="Sample Alarms"
     NNMBaseCategory="Sample Alarms" >
  <Severities critical="1"
              major="1"
              minor="1"
              warning="1"
              normal="1"/>
  <Acknowledgement
              acknowledged="1"
              unacknowledged="1"/>
</AlarmCategoryDef>
```

4. On the line prior to `</AlarmCategoryDef>`, add the following:

   ```
   <NodeSelection>
   </NodeSelection>
   ```

5. The <`NodeSelection>` element can restrict alarms so that only alarms from specified devices are displayed in the Alarms module. Three child elements are available: `<IPHostFilter>` or `<CapabilityFilter>`, or can refer to an `<OrganizationFilter>`.

   For example, the following `NodeSelection` filter allows all nodes to pass:

   ```
   <NodeSelection >
   <NodeSelection/>
   ```

   - If you wish to filter nodes based upon IP addresses or fully-qualified hostnames, enter the following within the NodeSelection start/stop elements:

     ```
     <IPHostFilter>
          <IPHost hostname="fully-qualified-hostname"/>
          <IPHost hostname="IP address"/>
     </IPHostFilter>
     ```

     For each node, enter an `<IPHost>` line with the appropriate information. If you wish to use Perl5 regular expressions, only .* is allowed.

     This example allows all nodes ending in “.eagle.wingnuts.com” to pass:
     `.*\.eagle\.wingnuts\.com`

- If you wish to filter nodes based upon a field or fields in the NNM object database, ovwdb, enter the following within the NodeSelection start/stop elements:

  ```
  <CapabilityFilter>
     <Capability
         field="ovwdb field name" (if not
  present=capability)
         value="sortCriteria"/>
     <Capability
         field="ovwdb field name"
         value="sortCriteria"/>
  </CapabilityFilter>
  ```

  For each node, enter a <Capability> line with the appropriate information. If you enter more than one <Capability> line, devices having any of the capabilities pass the filter.

- If you wish to restrict a *set* of <Organization> elements in the Management Data filter assigned to the current SIP role (for NNM modules, only the <NodeList> and <InterfaceList> child elements within the<Organization> apply), enter the following within the NodeSelection start/stop elements:

  ```
  <OrganizationFilter>
     <OrganizationRef
         href="a-defined-organization"/>
     <OrganizationRef
         href="a-defined-organization"/>
  </OrganizationFilter>
  ```

  Specify the href = *name* of an <Organization> currently defined within your SIP Customer Model.

  NOTE: the <Organization> element must be the last child element in the list of filters.

6. Your filter should look something like the following:

```
<?xml version='1.0' encoding='utf-8' standalone='no' ?>
<!DOCTYPE AlarmCategoryDef SYSTEM "NmAlarmCat.dtd">

<AlarmCategoryDef
     DisplayTitle="Sample Alarms"
     NNMBaseCategory="Sample Alarms"
  <Severities critical="1"
              major="1"
              minor="1"
              warning="1"
              normal="1"/>
  <Acknowledgement
              acknowledged="1"
              unacknowledged="1"/>
   <NodeSelection>
          <IPHostFilter>
            <IPHost hostname="a-machine-from-which-to-collect-data"/>
        </IPHostFilter>
        <CapabilityFilter>
            <Capability field="ovwdb-field (if not present=capability)
                        value=”xx"   />
        </CapabilityFilter>
        <OrganizationFilter>
            <OrganizationRef href="a-reference-to-a-defined-organization"/>
        </OrganizationFilter>
  </NodeSelection>
</AlarmCategoryDef>
```

7. If you used more than one of the child filter elements, only alarms from nodes passing all of the filters are displayed in the Alarms module.

8. Close and save the *NmAlarmCat*.xml file.

9. After you make modifications to this XML file, validate the syntax. See “Validating XML Files” on page 225 for more information.

10. After you make modifications to this XML file, you must restart the Tomcat engine. See “Restarting the Servlet Engine” on page 220 for more information.

11. In a browser, log into the portal to verify that the alarms appear as desired.

Additional Filtering Attributes and Elements

All of the choices explained below are specified in the each alarm category definition file (`*NmAlarmCat*.xml`). See `NmAlarmConfig.dtd` and `SampleNmAlarmCat.xml` for more information.

Alarms that pass the `Management Data` filter and `NodeSelection` display filter are further filtered by the following criteria as an AND condition before being displayed in any SIP portal view:

- `NNMStation`
  (optional, subset of NNM management stations configured in the SIP Configuration Editor. Alarms for this category are gathered only from the specified NNM management stations. If empty, all stations specified in the SIP Configuration Editor pass.)

NOTE Verify that you are consistent in your usage of *either* hostname or IP address when specifying the NNM management station in the SIP Configuration Editor and any alarm category definition files (`*NmAlarmCat*.xml` files).

- `NNMBaseCategory`
  (*required*, the alarm category currently defined in NNM that you wish to access for this SIP alarm category)
- `MatchDescSubstring`
  (*optional*, display only alarms whose messages include the specified text. If empty, all descriptions pass.)
- `OlderThanXMinutes`
  (*optional*, wait the specified number of minutes before displaying any alarm. If empty, show alarms as soon as they happen.)
- `Severities`
  (*required*, the NNM-defined alarm severity levels that you wish to include. No alarms pass if you specify none.)
- `Acknowledgement`
  (*required*, include alarms that are acknowledged and/or unacknowledged within NNM. No alarms pass if you specify neither.)

For example, the alarm criteria may specify NodeSelection: `host.corp.com` and Severity: `critical`. In this example, only alarms with the source name matching `host.corp.com` AND having a severity of `critical` are displayed.

If any of the above support multiple values, each value is treated as an OR condition. To continue the example, with severities `critical` and `major`, alarms matching the hostname `host.corp.com` *AND* having either a severity of `critical` *OR* `major` are displayed. If multiple nodes are supplied in the node list (such as `hostA.corp.com;hostB.corp.com`) alarms matching EITHER `hostA` OR `hostB` are displayed.

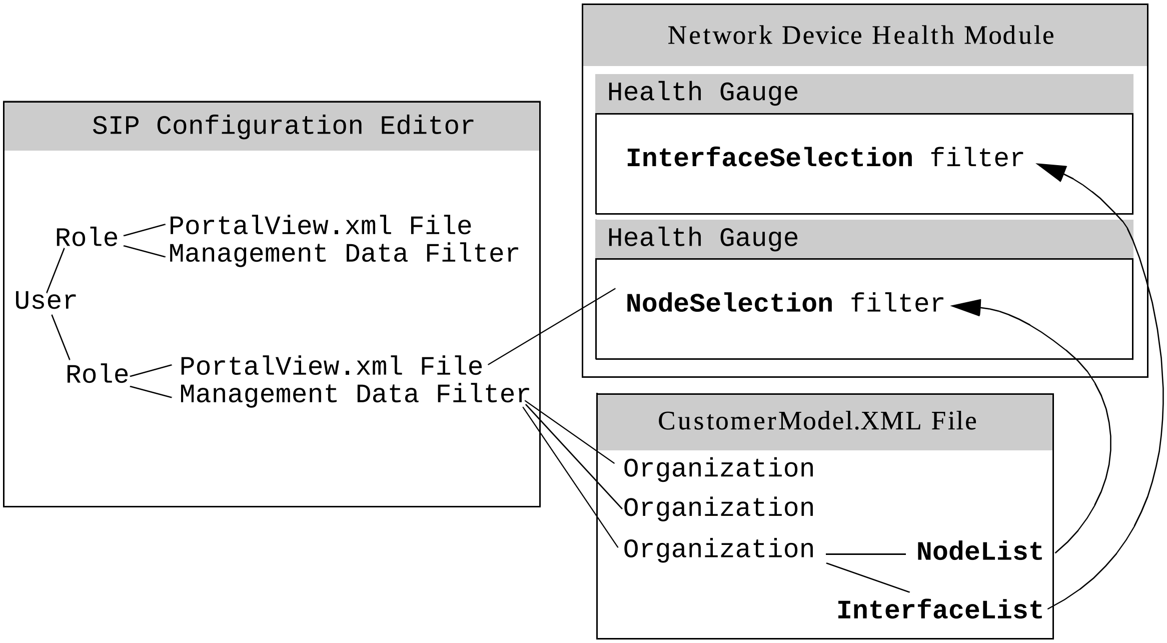
Filtering Possibilities for the Network Device Health Module

The Effect of Management Data Filter and Roles

Data displayed in the Network Device Health module must pass the `Management Data` filter assigned to the current SIP Role. See Chapter 6, “Segmenting the NNM Data for Your Customers,” on page 135, and the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “Creating Users and Roles” section, for information about the `Management Data` filter and the SIP User Role Model.

In addition to the filtering you applied through the `Management Data` filter, a finer, second level of filtering is applied with a `NodeSelection` filter and/or `InterfaceSelection` filter in each gauge.

Figure 7-2 Network Device Health Module’s “Display Filtering”



Node and Interface Filters for Specific Gauges

Whereas the `Management Data` filter defines what a module can *potentially* display, the `NodeSelection` filter and `InterfaceSelection` filter provide a finer level of control over what the module *actually* displays. Either a `NodeSelection` filter or `InterfaceSelection` filter must be specified in each gauge. These filters control the set of nodes or interfaces whose health the gauge is calculating.

When `NodeSelection` or `InterfaceSelection` filters are left empty, *all* nodes and interfaces pass (if they pass the `Management Data` filter). The `NodeSelection` filter has three potential child elements that can further restrict the nodes allowed to pass:

- `IPHostFilter`
  Passes only devices specified by hostname or IP address.
- `CapabilityFilter`
  Passes only devices having the specified `value` within the NNM object database, `ovwdb`; for example, `isRouter`.
- `OrganizationFilter`
  Passes only nodes included in `Management Data` filter of the currently assigned SIP Role (specified in the SIP Customer Model *organization*, `<NodeList>` elements).

The `InterfaceSelection` filter has two child elements that can further restrict the interfaces allowed to pass:

- `IPInterfaceFilter`
  Passes only interfaces specified by IP address.
- `OrganizationFilter`
  Passes only interfaces included in `Management Data` filter of the currently assigned SIP Role (specified in the SIP Customer Model *organization*, `<InterfaceList>` elements)

When `IPHostFilter`, `CapabilityFilter`, `OrganizationFilter`, and/or `IPInterfaceFilter` are used, they contribute to the *intersection* of the `Management Data` filter and each gauge filter. It is possible at runtime for the *intersection* of all filters to be the empty set. In this case, no nodes or interfaces are selected and the message “Managed Objects not found” is displayed for this gauge.

When `IPHostFilter`, `CapabilityFilter`, `OrganizationFilter`, and/or `IPInterfaceFilter` are included, yet left empty, they are considered an empty set, which allows nothing to pass.

Writing a NodeSelection Filter for a Gauge

To modify a `<NodeSelection>` filter, you must edit the gauge's XML file:

1. Make a backup of XML configuration files before you customize them. If you edit the file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.
2. Before you start, access the following two files as a reference:
   - *Windows 2000:*
     `%SIP_HOME%\conf\share\views\filter.dtd`
     `%SIP_HOME%\conf\share\views\OVNetworkHealth.dtd`
   - *UNIX:*
     `/opt/OV/SIP/conf/share/views/filter.dtd`
     `/opt/OV/SIP/conf/share/views/OVNetworkHealth.dtd`
3. In an ASCII or XML editor, open the gauge's XML file that you wish to modify, stored in one of the following locations:
   - *Windows 2000:*

     `%SIP_HOME%\conf\share\views\`*`portalView`*`.xml`
     `%SIP_HOME%\registration\defaults\OVDefaultNetHealth.xml`
   - *UNIX:*

     `/opt/OV/SIP/views/`*`portalView`*`.xml`
     `/opt/OV/SIP/registration/defaults/OVDefaultNetHealth.xml`
4. If you are modifying a module instance in a *`portalView`*`.xml` file, search for the following string to find your existing module to edit:

   `classid="com.hp.ov.portal.modules.health"`
5. Search for the `<Summary` string to locate the particular gauge you wish to edit. For example:

```
<Summary display="yes" displayDepth="3"
         id="ServerHealth" title="Server Health">
   <Component href="#IfHealth" vital="no" weight="1"/>
   <NodeSelection id="Servers" op="AND" title="Servers">
      <CapabilityFilter op="OR">
             <Capability field="isServer" value="true"/>
      </CapabilityFilter>
    </NodeSelection>
```

6. Locate the `<NodeSelection>` or `<InterfaceSelection>` element. Only one of these is allowed per gauge. If you find an `<InterfaceSelection>` element, you must delete it before proceeding.
7. Modify the <`NodeSelection>` element as desired. Only those devices that pass the `<NodeSelection>` filter are monitored for this gauge's health rating. Three child elements are available: `<IPHostFilter>` and/or `<CapabilityFilter>`, and/or `<OrganizationFilter>`.

   For example, the following `NodeSelection` filter allows all nodes to pass:

   ```
   <NodeSelection>
   </NodeSelection>
   ```

   - If you wish to filter nodes based upon IP addresses or fully-qualified hostnames, enter the following within the `<NodeSelection>` start/stop elements:

     ```
     <IPHostFilter>
          <IPHost hostname="fully-qualified-hostname"/>
          <IPHost hostname="IP address"/>
     </IPHostFilter>
     ```

     For each node, enter an `<IPHost>` line with the appropriate information. You may use any Perl5 regular expressions. See your Perl5 documentation for information about valid expressions. This example allows all nodes ending in ".eagle.wingnuts.com" to pass: .`*\.eagle\.wingnuts\.com`

     See also `www.perl.com` or `www.perldoc.com` for information about Perl5 regular expressions

   - If you wish to filter nodes based upon a field or fields in the NNM object database, `ovwdb`, enter the following within the `<NodeSelection>` start/stop elements:

     ```
     <CapabilityFilter>
        <Capability
            field="ovwdb-field" (if not present=capability)
            value="sortCriteria"/>
        <Capability
            field="ovwdb field name"
            value="sortCriteria"/>
     </CapabilityFilter>
     ```

For each `<Capability>`, enter a line with the appropriate information. If you enter more than one `<Capability>` line, devices having any of the capabilities pass the filter.

- If you wish to restrict a *set* of `<Organization>` elements in the `Management Data` filter assigned to the current SIP role (for NNM modules, only the `<NodeList>` and `<InterfaceList>` child elements within the `<Organization>` apply), enter the following within the `<NodeSelection>` start/stop elements:

  ```
  <OrganizationFilter>
     <OrganizationRef
         href="a-defined-organization"/>
     <OrganizationRef
         href="a-defined-organization"/>
  </OrganizationFilter>
  ```

  Specify the href = *name* of an `<Organization>` currently defined within your SIP Customer Model.

  NOTE: the `<Organization>` element must be the last child element in the list of filters.

8. Your filter should look something like the following:

```
<NetworkHealth showRawData="no" showUnknown="no">
   <Summary display="yes" displayDepth="3"
          id="ServerHealth" title="Server Health">
     <Component href="#IfHealth" vital="no" weight="1"/>
     <NodeSelection>
        <IPHostFilter>
            <IPHost hostname="a-machine-from-which-to-collect-data"/>
        </IPHostFilter>
        <CapabilityFilter>
            <Capability field="ovwdb-field” (if not present=capability)
                        value=”xx"/>
        </CapabilityFilter>
        <OrganizationFilter>
            <OrganizationRef href="a-reference-to-a-defined-organization"/>
        </OrganizationFilter>
     </NodeSelection>
 </Summary>
</NetworkHealth>
```

9. If you used more than one of the child filter elements, only nodes passing all of the filters are included in this gauge’s calculations.

10. Close and save the *`portalView`*`.xml` or module default XML file.

11. After you make modifications to this XML file, validate the syntax. See “Validating XML Files” on page 225 for more information.

12. If you made modifications to the `OVDefaultNetHealth.xml` file, you must restart the Tomcat engine. See “Restarting the Servlet Engine” on page 220 for more information.

13. In a browser, log into the portal to verify that the gauge appears as desired.

Writing an InterfaceSelection Filter for a Gauge

If you want to create a gauge that measures health for a set of interfaces, possibly on different nodes, create a `<InterfaceSelection>` filter, you must edit the gauge’s XML file:

1. Make a backup of XML configuration files before you customize them.

   If you edit the file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.

2. Before you start, access the following file as a reference:

   - *Windows 2000:*
     `%SIP_HOME%\conf\share\views\filter.dtd`
   - *UNIX:*
     `/opt/OV/SIP/conf/share/views/filter.dtd`

3. In an ASCII or XML editor, open the gauge’s XML file that you wish to modify, stored in one of the following locations:

   - *Windows 2000:*

     ```
     %SIP_HOME%\conf\share\views\portalView.xml
     %SIP_HOME%\registration\defaults\OVDefaultNetHealth.xml
     ```

   - *UNIX:*

     ```
     /opt/OV/SIP/conf/share/views/portalView.xml
     /opt/OV/SIP/registration/defaults/OVDefaultNetHealth.xml
     ```

4. If you are modifying a module instance in a *`portalView`*`.xml` file, search for the following string to find your existing module to edit:

   `classid="com.hp.ov.portal.modules.health"`

5. Search for the `<Summary` string to locate the particular gauge you wish to edit. For example:

```
<Summary id="InterfaceHealth" title="Interface Health"
         display="no" displayDepth="3">
 <Component weight="2" vital="yes" href="#IfStatus"/>
 <Component weight="1" href="#IfUtil"/>
 <Component weight="1" href="#IfInErrors"/>
 <Component weight="1" href="#IfOutErrors"/>
 <InterfaceSelection title="All Interfaces"
    id="AllInterfaces" op="AND">
 </InterfaceSelection>
</Summary>
```

6. Locate the `<NodeSelection>` or `<InterfaceSelection>` element. Only one of these is allowed per gauge. If you find an `<NodeSelection>` element, you must delete it before proceeding.

7. Add or modify the `<InterfaceSelection>` element as desired. Only those devices that pass the `<InterfaceSelection>` filter are monitored for this gauge's health reading. Two child elements are available: `<IPInterfaceFilter>` and `<OrganizationFilter>`.

   For example, the following `InterfaceSelection` filter allows all nodes to pass:

   ```
   <InterfaceSelection>
   </InterfaceSelection>
   ```

   - If you wish to filter interfaces based upon IP addresses, enter the following within the InterfaceSelection start/stop elements:

     ```
     <IPInterfaceFilter>
          <IPInterface ipAddr="10\.2\.5\.130"/>
          <IPInterface ipAddr="10\.2\.5\.130"/>
     </IPHostFilter>
     ```

     For each interface, enter an `<IPInterface>` line with the appropriate information. Use Perl5 regular expressions, if you wish. See your Perl5 documentation for information about valid expressions. This example allows all nodes ending in ".eagle.wingnuts.com" to pass:
     `.*\.eagle\.wingnuts\.com`

     See also `www.perl.com` or `www.perldoc.com` for information about Perl5 regular expressions

   - If you wish to restrict a *set* of `<Organization>` elements in the Management Data filter assigned to the current SIP role (for NNM modules, only the `<NodeList>` and `<InterfaceList>` child elements within the `<Organization>` apply), enter the following within the `<InterfaceSelection>` start/stop elements:

```
<OrganizationFilter>
   <OrganizationRef
       href="a-defined-organization"/>
   <OrganizationRef
       href="a-defined-organization"/>
</OrganizationFilter>
```

Specify the href = *name* of an `<Organization>` currently defined within your SIP Customer Model.

8. Your filter should look something like the following:

```
<Summary id="InterfaceHealth" title="Interface Health" display="no"
displayDepth="3">
        <Component weight="2" vital="yes" href="#IfStatus"/>
        <Component weight="1" href="#IfUtil"/>
        <Component weight="1" href="#IfInErrors"/>
        <Component weight="1" href="#IfOutErrors"/>
 <InterfaceSelection>
   <IPInterfaceFilter>
      <IPInterface ipAddr="10\.2\.5\.130"/>
      <IPInterface ipAddr="10\.2\.6\.54"/>
   </IPInterfaceFilter>
   <OrganizationFilter>
       <OrganizationRef href="OrganizationOne" />
       <OrganizationRef href="OrganizationTwo" />
   </OrganizationFilter>
 </InterfaceSelection> </Summary>
</NetworkHealth>
```

9. If you used more than one of the child filter elements, only nodes passing all of the filters are included in this gauge’s calculations.

10. Close and save the *portalView*.xml or module default XML file.

11. After you make modifications to this XML file, validate the syntax. See “Validating XML Files” on page 225 for more information.

12. If you made modifications to the `OVDefaultNetHealth.xml` file, you must restart the Tomcat engine. See “Restarting the Servlet Engine” on page 220 for more information.

13. In a browser, log into the portal to verify that the gauge appears as desired.

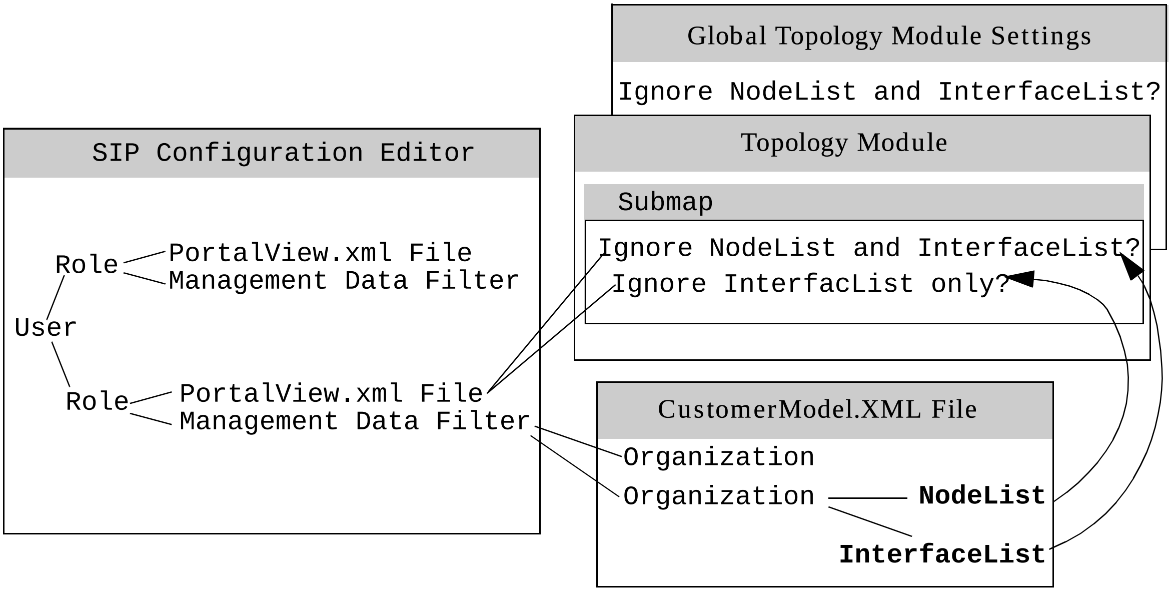
Filtering Possibilities for the Topology Module

The Effect of Management Data Filter and Roles

By default, the data displayed in Topology module must pass the `Management Data` filter assigned to the current SIP role. Submap symbols for objects that are an IP node, IP interface, or non-IP interface (objects that pass the `Management Data` filter) are displayed on the submaps within the Topology module in the SIP portal view. Other submap symbols, such as IP networks, segments, and container objects are only displayed in SIP if they are connected to an object that has passed the `Management Data` filter. See Chapter 6, “Segmenting the NNM Data for Your Customers,” on page 135, and the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`), “Creating Users and Roles” section, for information about the `Management Data` filter and the SIP User Role Model.

The Topology module gives you the option of ignoring the `Management Data` filter and displaying NNM submaps exactly as they appear on the NNM management station.

Figure 7-3 Topology Module’s “Display Filtering”



Global Filter Settings for Topology Modules

The global settings are in the `topologyConfig.xml` file. You must make changes directly in this file using an ASCII or XML editor (see "Establishing Global Settings for All Topology Modules" on page 129 for instructions). If you make any changes to the `topologyConfig.xml` file, you must follow the directions in:

- "Validating XML Files" on page 225.
- "Restarting the Servlet Engine" on page 220.

There are two attributes in the `topologyConfig.xml` file used to control filtering for all Topology modules:

- `defaultFilter`
  If set to *yes*, the NNM submaps are filtered according to the `<NodeList>` and `<InterfaceList>` elements in the `Management Data` filter assigned to the current SIP role.

  If set to *no*, the `Management Data` filter is ignored, and the SIP submaps include all symbols that appear on the NNM submap.

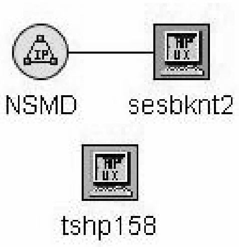
- `filterConSymbols`
  If set to *yes*, the NNM submaps are filtered according to the `<InterfaceList>` elements in the `Management Data` filter assigned to the current SIP role. Interfaces are displayed as connective lines in the SIP submap. Submap symbols, such as IP networks, segments, and container objects are only displayed in SIP if they are connected to a node that has passed the `Management Data` filter.

  See example submaps below showing the results from the following Management Data filter:

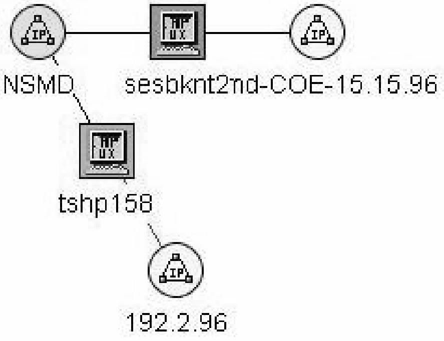
  ```
  <NodeList>
      <Node name="tshp108.div.co.com"/>
      <Node name="sesbknt2.div.co.com"/>
  </NodeList>
  <InterfaceList>
        <Interface name="10.2.115.2" type="ov-ipv4"/>
  </InterfaceList>
  ```

  If set to *no*, connective lines representing interfaces are displayed in SIP if the node to which they are connected passes the `Management Data` filter, irrespective of any limitations specified in the `InterfaceList`.

filterConSymbols set to “yes”



filterConSymbols set to “no”



Bypass All Filtering for Specific Topology Modules

Sometimes, a filtering strategy, that is desirable for the NNM Alarms modules and Network Device Health Gauge modules, produces awkward Topology layouts (for example, when the filters eliminate so many devices that the map shows only disjointed layouts). The following attribute setting enables you to optionally *bypass* filtering for individual submaps (see the `OVTopology.dtd` file for more information). This attribute is entered into a specific `Submap` element in a specific Topology module in a specific *`PortalView`*`.xml` file:

- `filter`
  If set to *yes*, (the default setting) this NNM submap is filtered according to the settings in the applicable `Management Data` filter for the particular SIP role.

  If set to *no*, the `Management Data` filter is ignored, and the SIP submap includes all symbols as they appear when this submap is displayed on your NNM management station.

See “Directly Editing the PortalView.XML File” on page 124 for instructions.

8 Troubleshooting

General

Determining the Amount of Time It Takes to Display Each Module

If you are having performance problems with a customer’s portal, you can determine the amount of time each module takes to load by using the Timer CGI token.

In your internet browser, run:

`http://<yourmachinename>/ovportal?Timer=yes` (or `true`)

This will show the time in milliseconds to display each module. To turn off timing, run:

`http://<yourmachinename>/ovportal?Timer=no` (or `false`)

Certain SIP modules require significant initial load times because a large amount of data is passed over the network when the module is first accessed. As a result, the first person displaying these modules experiences the longest delay. To enhance your customer's experience, log into the portal following any restart of SIP and open any portal view that contains the following modules (if used in your environment):

- A Network Device Health module
- One of the following modules:
  - Service Graph
  - Service Browser
  - Service Health
  - Service Card

If you open it first, the required information is already cached when your customers access their portals. Your customers won't experience the delay. It is not necessary to perform a log in for each portal user. A single log in, viewing the above listed modules, is sufficient.

The tab containing an NNM module remains blank, with the progress bar partially loaded.

One of the NNM management stations (that SIP gathers data from) may be in the early phase of an NNM backup procedure.

Wait for NNM’s backup to proceed beyond the `ovpause` state. The modules display when the NNM management station issues an `ovresume` command. If the browser timeout limit is exceeded while you are waiting, you must press `[Refresh]` to display the modules.

Alarms Module

The portal fails to display any alarms data

Possible Cause A:

Alarm categories can be configured to filter alarms in a number of ways. If you implement a `CapabilityFilter` within an alarm category's `NodeSelection` filter definition, the Alarms module has a dependency upon NNM's `ovwdb` process.

The portal may not be communicating with `ovwdb`.

Solution:

Restart `ovwdb` on the target system via `ovstart`.

Possible Cause B:

The portal may not be communicating with `ovalarmsrv`.

Solution:

Use the `ovstatus` command to validate the status of `ovalarmsrv`.

Restart `ovalarmsrv`, if necessary, by issuing the `ovstart` command.

Try to communicate directly with `ovalarmsrv`:

1. Enter: `telnet <NNMStationName> <ovAlarmsSrvPort>`.
2. In the telnet window, enter `0:0:CATEGORIES:TestUser`.

   If you get a response, it's up and running.
3. In the telnet window, enter `6` to end communications.

Possible Cause C:

Invalid port configured for `ovalarmsrv`.

Solution:

1. If the port number configured for an NNM system is invalid, the alarm module will not be able to obtain alarms for display from this system. Check which port each NNM station is communicating with. The port that the bits respond on depends on the entries in the `services` file.

On UNIX, the `services` file resides in `/etc/services`. `Ovalarmsrv` has two entries: `ovalarmsrv` and `ovalarmsrv_cmd`. The value that is set in the file determines which port `ovalarmsrv` runs on. (The same is true of `ovwdb.`)

On Windows NT/2000, the `services` file resides in `WINNT\system32\drivers\etc\services`.

2. Modify the entry or entries for Management Stations in the SIP Configuration Editor, as necessary, to match what you find.
3. Restart the servlet engine, see "Restarting the Servlet Engine" on page 220.
4. Changes to portal view files take effect when you display or refresh the portal view.
5. If you still do not see data, verify that you are communicating with `ovalarmsrv` on each NNM station listed in the SIP Configuration Editor by completing steps 1-3 in the solution to Possible Cause B.

Possible Cause D:

Specified NNM stations do not match those in the SIP Configuration Editor.

Solution:

Resolve the differences by editing the alarm category definition file and the Management Station configurations in the SIP Configuration Editor.

Possible Cause E:

Alarms categories defined in the `portalview.xml` files don't match alarm categories in the `NmAlarmCatsIndex.xml` file.

Solution:

Make sure they match.

Possible Cause F:

Base categories used in alarm category definition file not valid for the given NNM station.

Solution:

Make sure you are using valid NNM base categories for the stations you are connecting to.

Possible Cause G:

No data passed filters.

Solution:

Check the management data filters defined for this role.

Check any substring match, OlderThanXMins sevs, acks, or node selection filters defined for this category.

Possible Cause H:

No NNM stations configured in the SIP Configuration Editor.

Solution:

Make sure stations are listed.

Make sure the correct `ovAlarmSrvPort` is listed.

Make sure `alarmsDataSource` is set to yes.

Possible Cause I:

Timeout values are too short.

Solution I:

See "The portal fails to display a specific alarm category."

The portal fails to display a specific alarm category

Possible Cause A:

Timeout values are too short.

Solution A:

1. Modify the following attributes in the `NmAlarmConfig.xml` file. (Note: Each time you modify these attributes, you will need to complete steps 2 and 3 to see your changes.) More information about these attributes is available in the comments within the `NmAlarmConfig.dtd` file and in “Establishing Global Settings for Alarms Modules” on page 64:

   - `maxConnections`
     The maximum number of (socket) connections that a SIP server is allowed to establish with all the NNM management stations.
   - `connTimeOut` (zero or greater)
     The number of seconds to pause after each socket connection is opened.
   - `addSyncTime` (zero or greater)
     The number of seconds to add to `connTimeout` when making a synchronous call to get data from the `ovalarmsrv` on each NNM management station.
   - `socketTimeout` (zero or greater)
     The number of seconds to wait for a socket connection to be made.
   - `responseTimeout` (zero or greater)
     The number of seconds to wait each time for any response (protocol or data) from `ovalarmsrv`.
   - `maxWaitTime` (zero or greater)
     The maximum number of seconds to wait for a data response from `ovalarmsrv`.

2. Restart the servlet engine, see “Restarting the Servlet Engine” on page 220.

3. Changes to portal view files take effect when you display or refresh the portal view.

“Invalid XML” error message

Possible Cause:

Bad configuration file.

Solution:

Check the log file for parse errors, run the XML validator and make sure that you have a legitimate XML file. The problem could be in either the users configuration XML file or the configuration of the alarm display file.

The `portal_log` file is located in the following directory:

Windows 2000: `%SIP_HOME%\log\sip.log`

UNIX: `/opt/OV/SIP/log/sip.log`

SNMP Data Collection

Data collection configuration did not get updated to reflect changes in gauge definitions or Customer Model configurations

Symptom:

The expected data collection did not happen for devices added to a network device health gauge by expanding the filters or adding or creating a new gauge. Or, the expected data collection did not happen after changing or adding a configuration for a particular organization in the Customer Model.

Possible Cause:

- Automatic data collector configuration may not have been enabled on the NNM stations.
- The `ovcolautoconf` program may not have run since you made the change. NNM’s SNMP data collection configuration is updated by this program.
- The portal tab containing the Network Device Health module for this user/role has never been displayed.
- The portal tab containing the Network Device Health module for this user/role has not been displayed within the last 30 days. `ovcolautoconf` removes configuration entries for data that has not been requested within the last 30 days. (30 is the default and can be overwritten with the `-maxConfAge` option on `ovcolautoconf`.)
- SIP may not yet have sent its configuration requests to the NNM stations.
- Or, `ovcolautoconf` may be experiencing errors.

Solution:

1. Enable `autoDCConfig` if it is not already enabled. See “The Data Collection Process for the Network Device Health Module” on page 40 for information.
2. Display the portal tab containing Network Device Health.

3. Wait ten minutes.
4. Run `ovcolautoconf`. At the command prompt on the NNM systems, type: `ovcolautoconf -verbose.`(You may wish to run this command as a scheduled task.) Note that for UNIX `/opt/OV/bin` must be in your path.

   Collected data generally appears within a half hour of executing this command.

NOTE Any network device that you want to be included in the SNMP data collection process must be a *managed* device within the NNM topology database, and NNM must know the correct SNMP GET community name for that device before the device will be included in the SNMP data collection process.

For information about the steps required when the Service Information Portal software requests SNMP data from NNM, see the “The Data Collection Process for the Network Device Health Module” on page 40 and “Collecting Data for Network Device Health Gauges” on page 105. Verify that each step of the process is working correctly.

For other possible causes and solutions, see “Data not available” Error Message In Details Table under Network Device Health troubleshooting.

NNM Data Collector files of my SIP information are not being trimmed

Symptom:

NNM’s `snmpCollect` database is growing without bound.

Possible Cause:

No steps have been taken to trim the NNM `snmpCollect` database.

Solution:

See the NNM manual *Managing Your Network with NNM* or the SIP manual “Monitoring the Size of NNM’s snmpCollect Database” on page 42 for how to trim data in the `snmpCollect` database.

Extraneous data collections are being gathered for network device health gauges

Symptom:

When I check the file snmpRepPrev.conf, there are entries for devices for which I do not want to collect data.

Possible Cause A:

Are you collecting more data than you need? Check each XML file listed in the Customer Model Source definitions in the SIP Administrator Role's, Customer Model tab. Select All organizations, Show nodes, [View] summary. Check your network device health filter specifications within each PortalView.xml file.

Solution:

As necessary, modify the Customer Model Sources and/or the NodeSelection and InterfaceSelections within each PortalView.xml file.

Possible Cause B:

The unwanted entries may have been added at an earlier time, but due to SIP configuration changes, they are no longer needed.

Solution:

By default, ovcolautoconf removes configuration entries that have not been needed for 30 days. Run ovcolautoconf, using the -maxConfAge option if desired, to remove younger entries. See “ovcolautoconf.exe” on page 112 in the Network Device Health Module chapter.

Network Device Health Module

“Currently not configured” error message instead of gauge

Symptom:

No data is displayed in the gauge. The “Currently not configured” error message displays instead.

Possible Cause A:

No NNM stations are configured in the SIP Configuration Editor, or there are no NNM stations configured with the `snmpDataSource` attribute set to “yes.”

Solution:

Make sure there is at least one NNM station entry in the SIP Configuration Editor with the `snmpDataSource` attribute set to “yes.”

Possible Cause B:

The combination of the `MgmtData` filter and the gauge’s `NodeSelection` or `InterfaceSelection` results in no filtering. In other words, all nodes/interfaces pass the filters. Computation on *all* nodes/interfaces is not supported. See the `sip.log` file for specific error messages.

The `sip.log` file is located in the following directory:

Windows 2000: `%SIP_HOME%\log\sip.log`

UNIX: `/opt/OV/SIP/log/sip.log`

Solution:

Limit the number of devices that pass the gauge’s filters by doing one or more of the following (for more information about filters, see “Filtering Possibilities for the Network Device Health Module” on page 176):

- Narrow the `MgmtData` filter for this user role.
- In the `PortalView.xml` configuration file, narrow the gauge’s `NodeSelection` or `InterfaceSelection` filter. To avoid having the same problem the next time you create a new `PortalView.xml` file, modify the gauge definitions in the `OVDefaultNetHealth.xml` file.

When you insert the Network Device Health module into a new portal for the first time, all gauges defined within the `OVDefaultNetHealth.xml` file are copied into the `PortalView.xml` file.

"Managed objects not found" error message instead of gauge

Symptom:

No data is displayed in the gauge. The "Managed objects not found" error message is displayed instead.

Possible Cause A:

The combination of the customer's `MgmtData` filter of the current user role and the gauge's `NodeSelection` or `InterfaceSelection` filter settings are so restrictive that no network devices can pass. An entry will be logged to `sip.log`, such as "No Nodes found for health summary category <*category name*>" or "No Interfaces found for health summary category <*category name*>".

This is most likely to occur with Key Device Health, CPE Health and Server Health.

The `sip.log` file is located in the following directory:

Windows 2000: `%SIP_HOME%\SIP\log\sip.log`

UNIX: `/opt/OV/SIP/log/sip.log`

Solution:

In NNM, select `Edit->Find->Object By Attribute` to determine if there are any devices with the specified capability set to TRUE (`isKeyDevice`, `isCPE`, `isServer`). If such nodes exist, do they pass the customer's `MgmtData` filter? If the desired capability is not set for one or more nodes, see *Managing Your Network with NNM* for information about how to set NNM object capabilities for the various network devices.

If necessary, modify the `MgmtData` filter and/or locate the gauge's <*Summary*> (definition in `PortalView.xml`) and modify the `NodeSelection` or `InterfaceSelection` filter. For more information, see "Filtering Possibilities for the Network Device Health Module" on page 176.

Possible Cause B:

An NNM station entry in the SIP Configuration Editor is incorrectly specified.

Solution:

Verify that the `ovwdbPort` attributes specified in the SIP Configuration Editor are correct. For more information, see the online `[Help]` in the SIP Configuration Editor or "On the SIP Server" on page 28.

"Data currently unavailable" error message instead of gauges

Symptom:

No data is displayed in any gauge. The "Data unavailable" error message displays instead.

Possible Cause A:

The `sip.log` file contains a detailed message about the problem. There may be a syntax error in the `netHealthConfig.xml` file. (For example, the href syntax for a health Element may be invalid.)

Solution:

Check the `sip.log` file for a detailed message about the problem.

The `sip.log` file is located in the following directory:

Windows 2000: `%SIP_HOME%\SIP\log\sip.log`

UNIX: `/opt/OV/SIP/log/sip.log`

Restore the `netHealthConfig.xml` file to its last working state or fix the syntax error identified in the `sip.log` file.

See the comments in `netHealthConfig.dtd` for information about the correct syntax within this file.

Possible Cause B:

The CGI program `getnnmdata.exe` may not be on the NNM station.

Solution:

If the NNM station is running version 6.1 of NNM, see "On each NNM Management Station" on page 31 for instructions on how to install CGI programs needed by SIP.

Possible Cause C:

The hostname or webSrvPort attributes in the SIP Configuration Editor may be incorrectly specified.

Solution C:

Verify the hostname and webSrvPort attributes are correct. See the online [Help] in the SIP Configuration Editor or “On the SIP Server” on page 28 for more information.

Possible Cause D:

SIP 1.0 was uninstalled from one of your NNM management stations after configuring the NNM management station for SIP 2.0 or 3.0. Uninstalling SIP 1.0 uninstalls a library that is required by the SIP 2.0 and 3.0 CGIs. The library name is:

Windows NT/2000: std312d.dll

UNIX: libstd12d.sl

Solution D:

Check the sip.log for the following error message:

```
error   NetDevHealth    Thread-21       988216387619    Unable to successfully
invoke http://<hostname>:<port>/OvCgi/getnnmdata.exe -- 405:
java.io.FileNotFoundException: http://<hostname>:<port>/OvCgi/getnnmdata.exe
```

You can fix the problem by repeating the CGI installation process described in “On each NNM Management Station” on page 31. This will reinstall SIP 2.0 CGI executables and the library that was removed. It will NOT overwrite the configuration files, mibExprAuto.conf and snmpRepAuto.templ, so any customizations you may have made are preserved.

“Data unavailable” error message in details table for all scores except Interface Status

Symptom:

“Data unavailable” appears instead of data in the detail tables.

Possible Cause A:

The NNM SNMP Data Collector may not be configured to collect the data needed by SIP.

Solution:

See “Data collection configuration did not get updated to reflect changes in gauge definitions or Customer Model configurations” on page 195 and “The Data Collection Process for the Network Device Health Module” on page 40.

Possible Cause B:

NNM may not be able to contact the nodes in question via SNMP.

Solution:

In NNM, highlight the node, and select `Tools->SNMP MIB Browser` and walk the MIB2 interfaces group to see if the node is responding to SNMP requests. If it is not responding, there are several possibilities:

- The node may be down. Does the Interface Status column show “Down”? Does the node respond to ping?
- NNM may be using the wrong SNMP GET community string for the node. In NNM, select `Options->SNMP Configuration` to determine what community string NNM is using for the node. If you change one of these, while logged in as `root` or administrator, at the command prompt, type **`snmpCollect -C <nodename>`**
- The node’s SNMP agent is not up or not responding.

Possible Cause C:

Network Node Manager is having problems with the SNMP data collection process.

Solution:

Run **`ovstatus-c snmpCollect`** on NNM to verify `snmpCollect` is running. See NNM log file `../log/snmpCol.trace` on the NNM system.

Possible Cause D:

By default, Service Information Portal gauges only use data up to 1 hour old. Perhaps the data is too old to be considered “near real-time” by network device health.

Solution:

To increase the acceptable age for SNMP data, increase the `maxAge` attribute in the `netHealthConfig.xml` file (the value represents minutes). This setting affects all gauges in all defined customer portals.

Possible Cause E:

Did you enable automatic data collection configuration on the NNM station? If so, did you edit the `snmpRepAuto.templ` file directly? If there is a syntax error in this file, the data collection process will fail.

Solution:

If the problem arose after you edited the `snmpRepAuto.templ` file, restore the `snmpRepAuto.templ` file to its last working state and following the directions in `snmpRepAuto.templ` when making changes.

"Data unavailable" error message on one row of details table (for a particular node or interface)

Symptom:

"Data unavailable" appears in the detail tables.

Possible Cause A:

NNM may not be able to contact the node in question via SNMP.

Solution:

For single MIB values, in NNM highlight the node, and select `Tools->SNMP MIB Browser`. Walk the MIB2 interfaces group to see if the node is responding to SNMP requests. For collections on MIB expressions, in NNM go to `Options->Data Collections & Thresholds:SNMP`, highlight the collection in question, and choose `Actions->Test SNMP`.

If it is not responding, there are several possibilities:

- The node may be down. Does the `Interface Status` column show `Down`? Does the node respond to ping?
- NNM may be using the wrong SNMP GET community string for the node. In NNM, select `Options->SNMP Configuration` to determine what community string NNM is using for the node. If you change one of the community strings and want to immediately attempt to reinitialize for a particular node (instead of waiting until the scheduled data collection check), while logged in as `root` or administrator, at the command prompt, type **`snmpCollect -C <nodename>`**
- The node's SNMP agent is not up or not responding.

Possible Cause B:

An SNMP agent patch may be required on the node in question.

Solution:

If the node is an HP-UX node, the column is `Interface % Utilization`, and the raw utilization value is greater than 100%, this is due to a known SNMP agent defect on HP-UX. The 11.0 patch for the HP-UX SNMP Agent software that fixes this problem is PHNE_21673 from the following web site:

`http://www.hp.com`, then click “technical support”, “unix and mpe/ix servers”

Note: When this agent defect is encountered, a warning is logged in the `sip.log` file: “Data value XYZ does not fall into any of the specified XML ranges.”

Possible Cause C:

NNM may have incomplete network interface information.

Solution:

Check for valid IF Index values in NNM’s topology database. In NNM, highlight the node, drill down into the node’s Interface submap. Right click on an interface in question, and select `Interface Properties`. Examine the `Interface #` field. If this is blank or 0 (zero), Network Device Health is not able to retrieve SNMP data from this node. Such interface numbers values sometimes occur when NNM’s discovery has not been allowed to complete for a node. Verify that NNM’s `netmon` process is running. Is NNM’s auto-discovery enabled? Are the node and interface *managed* within NNM?

Possible Cause D:

The node in question may not support one of the MIB variables used in computing that column value.

Solution:

The most common case of this is the `CPU Utilization` column in the first level of node drill down. This uses the Cisco MIB variable `cisco.local.lsystem.avgBusy5`, hence non-Cisco nodes will display the “Data unavailable” string for this column.

One approach to determining which MIB variable is unsupported is to let `snmpCollect` tell you what is wrong:

1. Toggle on `snmpCollect` tracing: `snmpCollect -T`
2. Toggle on `snmpCollect` verbose tracing: `snmpCollect -V`
3. Force a collection check on the node in question: `snmpCollect -C <nodename>`
4. Toggle off `snmpCollect` verbose tracing: `snmpCollect -V`
5. Toggle off `snmpCollect` tracing: `snmpCollect -T`
6. Examine `$OV_LOG/snmpCol.trace` for messages indicating why `snmpCollect` couldn't set up the collections for that node.

In general, you can do the following to determine which MIB variables are used in computing health column values:

1. Go to `netHealthConfig.xml`. Find the `Metric` whose Title matches the column title in question (for example, CPU Utilization).
2. Look at the last part of the href attribute of the Element to determine the NNM MIB expression/variable used. For example, `href="snmp://%item%[0]/p_cisco5minavgbusy"` indicates that the MIB expression `p_cisco5minavgbusy` is being used.
3. If a MIB expression (not a simple MIB variable) is being used, to determine which MIB variables are requested in the mathematical formula, open NNM and select `Options->Data Collection & Thresholds: SNMP`. Find the expression in the `MIB Objects Configured for Collection` list. Double-click on the entry. This will bring up a dialog box. Click on `[Describe]`:
   - **Direct NNM MIB expression**: shows the mathematical formula of MIB variables that the direct MIB expression is using.
   - **Indirect NNM MIB expression**: shows a list of possible direct MIB expressions in use. The actual direct MIB expression used will depend upon the attributes of the interface. To determine which direct MIB expression is being requested from a specific node, exit the `Description` dialog box, and in the `MIB Object Collection Summary` list click on the node in question and select `Actions->Test SNMP`. Note which direct MIB expression is being requested for each interface (for example, `IfHDplxUtilization`). Exit the `Test SNMP` dialog box.

Unfortunately, there aren't many options when a node does not support an SNMP variable used to compute health. You can do one of the following:

- Remove the Health Component altogether from the gauge’s <*Summary*> entry in the `PortalView.xml` file. In this way, the associated SNMP variable/expression will not be used in computing health.
- Configure the `MgmtData` filter or the Portal View file’s `NodeSelection` or `InterfaceSelection` such that only nodes supporting that MIB variable pass.

“Data unavailable” error message in one column of details table (for all nodes or interfaces)

Symptom:

“Data unavailable” in detail tables for all nodes.

Possible Cause A:

The node in question may not support one of the MIB variables used in computing that column value.

Solution:

See the Solution under Possible Cause D on page 204.

Possible Cause B:

There may be an error in the specification of the requested MIB variable or MIB expression, preventing NNM’s `snmpCollect` process from performing any collections on this metric.

Solution:

Check for MIB variable/expression validity. See “Data unavailable” error message on one row of details table (for a particular node or interface) on the previous page.

For information about the steps required when the Service Information Portal software requests SNMP data from NNM, see “Collecting Data for Network Device Health Gauges” on page 105. Verify that each step of the process is working correctly.

Nodes or interfaces missing from details table

Symptom:

You expected more nodes to pass the filters than are displayed within the Detailed Network Health table.

Possible Cause A:

By default, only 20 rows are displayed within any Detailed Network Health table, although more than 20 may be included in the health calculation.

Solution:

By default, only the 20 least healthy nodes/interfaces are shown. To increase this value, increase the `maxDetail` attribute in the `netHealthConfig.xml` file. This setting affects all gauges within all defined portals.

If you increase the number of rows displayed and still have nodes missing, verify that the `MgmtData` filter, `NodeSelection` and/or `InterfaceSelection` filters are correctly defined.

Possible Cause B:

The missing nodes or interfaces may currently have status of “unknown” in the NNM object database. This happens when the device is unreachable from the NNM management station due to some connection device being down (such as a router).

Solution:

By default, devices with an “unknown” status are excluded from the details table. If you wish to include “unknown” devices, change the `showUnknown` attribute setting to “yes” in the `netHealthConfig.xml` file. This setting affects all gauges within all defined portals. The unknown devices, if any, will be placed at the bottom of the table, following any “known” devices.

Reading on the gauge does not match the values in the details table

Symptom:

The values displayed in the Detailed Network Health table do not seem to support the final value displayed on the gauge.

Possible Cause:

By default, only 20 rows are displayed within any Detailed Network Health table, although more than 20 may be included in the health calculation.

Soluton:

By default, only the 20 least healthy nodes/interfaces are shown. To increase this value, increase the `maxDetail` attribute in the `netHealthConfig.xml` file. This setting affects all gauges within all defined portals.

Score for a node does not match the values given for its interfaces in the next lower level of details table

Symptom:

The values displayed for a node's interfaces in the Detailed Network Health table don't seem to support the value displayed for the overall node.

Possible Cause:

By default, only 20 rows are displayed within any Detailed Network Health table, although more than 20 may be included in the health calculation.

Solution:

By default, only the 20 least healthy nodes/interfaces are shown. To increase this value, increase the `maxDetail` attribute in the `netHealthConfig.xml` file. This setting affects all gauges within all defined portals.

Gauges are not available for me to add from the list of available Network Device Health Gauges

Symptom:

When displaying the list of available gauges, some are missing.

Possible Cause:

When you insert the Network Device Health module into a portal for the first time, all gauges defined within the `OVDefaultNetHealth.xml` file are copied into the `PortalView.xml` file. From that point on, only gauges physically defined within the current `PortalView.xml` file are displayed on the list.

Soltion:

If a gauge that you wish to use is missing from the selection list, open a Portal View XML configuration file that contains the gauge and copy the gauge's definition (<*Summary*>) into the `PortalView.xml` file that you want to add it to.

What does the 100% health score mean? How do I display more information about how health scores are calculated?

Symptom:

I want to display more information about the health score calculation in the details table.

Possible Cause:

The `showRawData` attribute in the `PortalView.xml` file may be set to `NO`.

Solution:

To display the maximum amount of information about how health scores are calculated, set the `showRawData` attribute in the `PortalView.xml` file may be set to `YES`.

Note that `showRawData` applies to all *Summary* sections in that Network Device Health module instance.

The data collected seems to switch from one router interface to another

Symptom:

The data collected for a particular interface in a router is questionable.

Possible Cause:

Each time a router reboots, the SNMP interface index mapping is reconfigured. The ifIndex numbers assigned may drift from one interface to another. The NNM Data Collector is using `ifIndex` to identify interface instances. The collected data may be coming from a different interface after each router reboot.

Solution:

The drift of `ifIndex` numbers will stabilize when the Service Information Portal's data collection configuration is updated by executing the `ovcolautoconf` command on the NNM system. The problem only appears after router reboots.

Topology Module

“Data currently unavailable” message appears below a submap’s title bar

Possible Cause A:

The NNM `ovwdb` process on an NNM management station is not running. The Topology module is dependent upon this process to supply information.

Symptom A: Log message: ERROR: Connection to OVW lost.

The `portal_log` file is located in the following directory:

Windows 2000: `%SIP_HOME%\log\sip.log`

UNIX: `/opt/OV/SIP/log/sip.log`

Solution A:

Run the `ovstart` command on the NNM management station.

Possible Cause B:

ovw is not running on the server with the map open.

Symptom B:

Log message reads:

```
error Topology:Ovw    Thread-19    985636515095    An ovw
serving the map default was not found on the host
jorma.cnd.hp.com. Tried the following port(s):3700 3701
```

The `sip.log` file is located in the following directory:

Windows 2000: `%SIP_HOME%\log\sip.log`

UNIX: `/opt/OV/SIP/log/sip.log`

Solution B:

Start ovw on the server with the map open.

Possible Cause C:

The map is running, but the map is running with a session number greater than 0 and there is a gap of *numMapRetries* in the sequence of ovw session ports.

Symptom C:

See Symptom B.

Solution C:

Exit the ovw session and restart it so that it uses the lowest available session number.

Possible Cause D:

OVW authorization not configured on the remote server.

Symptom D:

Log message reads:

```
Permission denied. The map default was not found on the host
nganesan.cnd.hp.com
```

The `sip.log` file is located in the following directory:

Windows 2000: `%SIP_HOME%\log\sip.log`

UNIX: `/opt/OV/SIP/log/sip.log`

Solution D:

Modify `ovw.auth` and `ovwdb.auth` on remote server.

Possible Cause E:

`Wrong ovwDbPort` Specified in the SIP Configuration Editor.
`Data Currently Unavailable`

Symptom E:

```
error Topology:Ovw    Thread-21    985710826476   Database
not available
```

Solution E:

Fix `ovwDBPort` setting in the SIP Configuration Editor.

Possible Cause F:

Submap is not persistent.

Symptom F:

Log file reads: errorTopology: OvwThread-19988141835579 The submap name may be incorrect or if Customer Views is installed the submap name is not unique. If this is the case, specify the whole path.

Solution F:

Make the submap persistent.

Possible Cause G:

Submap name is misspelled, or if customer views is installed, the submap name may not be unique.

Symptom G:

See symptom F.

Solution G:

Specify the whole path. Correct spelling.

Possible Cause H:

Submap has been deleted and no longer exists

Symptom H:

See symptom F.

Solution H:

Remove submap name from configuration file or re-create submap.

Topology Map module hangs when trying to display a submap

Possible Cause A:

The NNM management station is in the pause state, for example for a backup procedure.

Solution A:

Wait until the backup complete (or run the `ovresume` command).

Possible Cause B:

There is an ovw running on the server that is hung. This may or may not be the ovw for the map having the submap to be displayed. (For example, a hung ovw can occur if you exit out of a Reflection X session without closing ovw.)

Solution B:

Check to make sure all ovw processes that are running are responding. If any of the ovw processes are hung, manually stop the process.

Possible Cause C:

It may not be hung but may just be taking a long time to find the map. This could occur if *numMapRetries* is high or there are many session number gaps between the running ovw sessions. Timeouts will generally only be a problem on Windows NT.

Solution C:

One solution is to exit and restart all the ovw sessions. This will restart the ovw sessions with contiguous session numbers.

Possible Cause D:

Another process on port 3600 or 3601. Check log file. Calling `OvwInitSession` on port.

Solution D:

Determine port configuration on the NNM management station.

“Managed objects not found” message is displayed in the submap area

Possible Cause A:

It may be that there are no symbols in the submap.

Possible Cause B:

It may be that a filter has been applied that results in no objects passing the filter for that particular submap.

Solution:

Change the filter that you are applying.

None of the icon symbols are displayed correctly

Possible Cause A:

No topology symbol images (GIF files) have been gathered from your NNM management station and copied to the SIP server. Therefore, SIP can only display the background shape (circle, square, etc.) for map symbols.

Solution A:

Open the SIP Configuration Editor and verify that the `symbolRegSource` attribute is set to “`yes`” for at least one NNM management station.

Possible Cause B:

The wrong `webSrvPort` is specified in the SIP Configuration Editor. Therefore, no topology data can be gathered from your NNM management station.

Solution B:

Check the `sip.log` error log for the following:

```
error Topology:SymbolRefreshCache   Thread-21   985709926656
You might want to check the webSrvPort for
<NNMstationHostName>:8880 java.net.ConnectException:
Connection refused: no further information
```

Open the SIP Configuration Editor and verify that the web server port actually in use by the NNM management station is specified in the `webSrvPort` attribute. To determine which port is configured for the NNM web server, ask your NNM administrator. On an NNM management station running in *UNIX*, see `/opt/OV/apache/conf/httpd.conf`

Some of the icon symbols are not displayed correctly

Possible Cause A:

If you add or change NNM’s topology symbols, SIP gathers the new symbol information from your NNM management station and copies the GIF files to the SIP server according to the following two settings:

- the SIP Configuration Editor’s `symbolRegSource` attribute
- `topologyConfig.xml` file’s `symbolFetchRateInMin` attribute

Solution A:

Open the SIP Configuration Editor. Verify that at least one NNM management station has the `symbolRegSource` attribute set to `"yes"` (see the online `[Help]` in the SIP Configuration Editor for more information).

To force SIP to update the NNM topology symbol information, open the SIPConfiguration Editor and select `Use As OVw Symbol Source` for this management station. Now you must follow the steps in “Restarting the Servlet Engine” on page 220.

Background graphic for a submap is not displayed

Note: Only the default background graphics in the `<OpenView directory>/backgrounds` directory are certain to work across servers.

Possible Cause:

The background graphics file was not found in the expected location on the portal server.

Solution:

Install the background graphics on the portal server in the same location as the remote server.

“Currently not configured” message appears below Topology module title bar and no submap displays.

Possible Cause A:

Hostname in `PortalView.xml` file does not match host name in the SIP Configuration Editor.

The `sip.log` file contains:

```
error Topology:Ovw    Thread-21   985710426012   The host
nganesan.cnd.hp.com is not specified in mgmtStations.xml
```

Possible Cause B:

No stations configured in the SIP Configuration Editor

The `sip.log` file contains:

```
error mgmtStations Thread-21 985710619802 There are no NNM
stations configured in mgmtStations.xml. At least one station
must be configured for NNM modules to operate.”
```

The Topology module opens slowly

Possible Cause:

You are getting map data from multiple NNM stations, and they all have the same applications installed (i.e., the same symbol information).

Solution:

Open the SIP Configuration Editor on the SIP server. Improve the startup performance by specifying `symbolRegSource=yes` for just one NNM station and `symbolRegSource=no` for all the rest.

A Restarting Tomcat

Restarting the Servlet Engine

After making certain configuration changes, you must restart the servlet engine before changes take effect:

- After adding or changing a module registration file.
- After making changes to the authentication provider configuration.
- In other situations where you are specifically instructed to do so.

To Restart the Servlet Engine from the SIP Administration Pages

Be aware that you and all other SIP users will be logged out when you restart the servlet engine.

1. Log in as a user who has access to a special `SIP Administrator` role. For more information, see “Understanding Special SIP Administrator Roles” on page 103 of the *SIP Deployment and Integration Guide* (`SIP_Deployment_Integration.pdf`).
2. Switch to the `SIP Administrator` role, if it is not already displayed.
3. Click the `SIP General Admin` tab.
4. In the `Servlet Engine Control` segment, click `[Restart]`.

To Restart the Servlet Engine from Outside of SIP

Windows 2000:
From the `Control Panel`, select `Services`. Stop and then restart `Tomcat`. Alternatively, you can use the command line: **`net stop tomcat`** and **`net start tomcat`**.

UNIX:
As `root`, stop and restart the web server and servlet engine by running the following. (The DISPLAY variable must be configured prior to restarting the web server and servlet engine, unless DISPLAY is set in `/etc/rc.config.d/ovsip`.)

Stop on HP-UX: `/sbin/init.d/ovsip stop`
Start on HP-UX: `/sbin/init.d/ovsip start`

Stop on Solaris: `/etc/init.d/ovsip stop`
Start on Solaris: `/etc/init.d/ovsip start`

B Working with XML

Rules for Direct Editing of XML Files

- Make a backup before modifying XML files.
- Understand editing permissions on XML files.
- Validate the XML after you modify it.
- Be careful not to lose changes made through the GUI. This can happen when you edit through the XML file and edit through the GUI at the same time.

Backing Up XML Files

Make a backup of XML configuration files before you customize them. If you edit the file and get incorrect XML syntax, you may want the ability to revert to the previous version of the file.

Understanding Editing Permission on XML Files

When using the editing windows within the SIP portal, the web server needs to have read/write permissions to the underlying files in order to save your changes. The apache web server and SIP run as:

Solaris: user `"nobody"`

HP-UX: user `"www"`

At runtime, `umask` is set by the `tomcat.sh` script to 022, so files are created mode 0644 and directories created mode 0755.

Therefore, at install time, SIP sets permissions and ownership for files to `mode 0644` and directories to `mode 0755`. If you add or change anything, make sure directories are owned by the appropriate user specified above, files set to `mode 0644`, and directories set to `mode 0755`.

For `tomcat` to operate properly, the following directories and all files underneath them need to have the correct permissions set (user as specified above, files set to mode 0644, and directories are set to mode 0755):

- `/opt/OV/SIP/tomcat`
  (directory only, so tomcat can create the work directory when needed)

- `/opt/OV/SIP/tomcat/conf`
  (directory only)
- `/opt/OV/SIP/tomcat/logs`
  (directory, all subdirectories, and all files)
- `/opt/OV/SIP/tomcat/webapps`
  (directory, all subdirectories, and all files)
- `/opt/OV/SIP/tomcat/work`
  (directory, all subdirectories, and all files)

For SIP to operate properly, these directories and all `.xml` files (not `.dtd` files) underneath them need to have the correct permissions set (`user` set to anyone with editing permissions, files set to mode 0644, and directories are set to mode 0755):

- `/opt/OV/SIP/conf/share/organizations`
  (directory, all subdirectories, and all .xml files)
- `/opt/OV/SIP/conf/share/users`
  (directory, all subdirectories, and all .xml files)
- `/opt/OV/SIP/conf/share/modules`
  (directory, all subdirectories, and all .xml files)
- `/opt/OV/SIP/conf/share/roles`
  (directory, all subdirectories, and all .xml files)
- `/opt/OV/SIP/conf/share/views`
  (directory, all subdirectories, and all .xml files)

Validating XML Files

The Service Information Portal will detect and report an invalid XML configuration file. However, after you make modifications to XML files, you may want to validate your XML syntax.

Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. This command uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP.

For the command to work from outside the `bin` directory, add the following to your PATH variable:

Windows 2000: `%SIP_HOME\bin`
UNIX: `/opt/OV/SIP/bin`

The correct usage of the `xmlvalidate` command is:

```
xmlvalidate -v <xml filename>
```

An XML file is “well-formed” if it conforms to a minimal set of rules defined for all XML documents. It is “valid” if it conforms to the DTD listed at the beginning of the XML file.

Sometimes an error reported by `xmlvalidate` may not clearly indicate how to fix the problem. For example, a message like “Attribute ‘name’ must be declared for element type ‘XYZ’, is an indication that the attribute ‘name’ may have been misspelled.

As an alternative to `xmlvalidate`, you can find an XML validation tool for Windows NT at `www.xmlspy.com`.

Avoiding Loss of Changes

If you are using the portal interface to change a configuration and directly editing the XML configuration file at the same time, be careful not to lose the changes made through the interface by writing out the file over the interface changes.

Index