

HP OpenView Operations and HP OpenView Service Navigator Integration with Service Information Portal

Version: 3.2

Windows®, HP-UX, and Solaris



Manufacturing Part Number: None

September 2004

© Copyright 2004 Hewlett-Packard Development Company, L.P.

Legal Notices

Warranty.

Hewlett-Packard makes no warranty of any kind with regard to this document, 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.

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.

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 in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices.

©Copyright 1999-2004 Hewlett-Packard Development Company, L.P.

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

Trademark Notices.

Adobe® is a trademark of Adobe Systems Incorporated.

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® 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.

Support**1. How OVO/OVSN Works with SIP**

The OpenView Operations Module for SIP	12
The Service Navigator Modules for SIP	13
Communication Paths Between OVO/OVSN and SIP	14
Installation of the Modules	17

2. Configuration Steps

Establishing Communication Between OVO/OVSN and SIP	20
On each OVO management server	20
On each OVSN management server	22
On the SIP Server	22
SIP Distribution Model	28
Running in Languages Other Than English	29
Configuring SIP to Access UTF-8 Data From OVO/OVSN	29
Getting Japanese Node Lists and Service Lists from OVO for Windows	30
Secure Socket Layer (SSL) Support	31
Running the OVO/OVSN Modules in a Wireless Environment	32

3. OVO Messages Module

Using the OVO Messages Module	36
Adding the OVO Messages Module	36
Editing the OVO Messages Module	38
Using the OVO Messages - Edit page	38
Directly editing the PortalView.xml file	39
Establishing Global Settings for OVO Messages Modules	42
OVOConfig.xml/dtd	42
SIP Configuration Editor	44
Editing OVO Message Categories	45
Modifying Existing Message Categories	45
Creating New Message Categories	47
Eliminating a Message Category	48
Relevant Files	50

4. The Service Browser Module

Using the Service Browser Module	54
Adding the Service Browser Module	55

Contents

Editing the Service Browser Module	57
Using the Service Browser - Edit Page	57
Directly Editing the PortalView.xml File	58
Relevant Files.	61
5. The Service Cards Module	
Using the Service Cards Module	64
Adding the Service Cards Module	65
Editing the Service Cards Module	67
Using the Service Cards - Edit Page	67
Directly Editing the PortalView.xml File	68
Relevant Files.	71
6. The Service Graph Module	
Using the Service Graph Module	74
Adding the Service Graph Module.	75
Editing the Service Graph Module	77
Using the Service Graph - Edit Page.	77
Directly Editing the PortalView.xml File	78
Relevant Files.	82
7. The Service Health Module	
Using the Service Health Module	86
Adding the Service Health Module	87
Editing the Service Health Module.	89
Using the Service Health - Edit Page	89
Directly Editing the PortalView.xml File	90
Relevant Files.	95
8. Creating Your Own Service Modules	
Understanding the Possibilities	98
Creating Your Module's Registration File	100
Developing Your Own XSL Stylesheet	101
Sample XSL Stylesheets.	101
Rules for Creating Your Own Stylesheet.	102
Creating Your Module's Default Instance.	104

Inserting Your Custom Service Module into a Portal	107
Writing Your Own Online Help	108
The Default Help Topic for the Module	108
One Module Instance's Help Topic	109
Relevant Files	111
9. Segmenting the OVO/OVSN Data by Customer	
Integrating OVO/OVSN Data into Your Customer Model	116
Manually Creating NodeList and ServiceList Elements	120
Customer-to-Node Mappings	121
Customer-to-Service Mappings	125
Registering SIP Customer Model Sources	132
Registering a Customer Model Source	132
Unregistering a Customer Model Source	133
10. Display Filtering for OVO/OVSN	
Introduction to Display Filtering	136
Filtering Possibilities for the OVO Messages Module	137
Filters for Specific Message Category Definitions	137
Filtering at the Module Instance Level	140
Display Filtering the OVSN Modules	142
Filtering Possibilities for the Service Browser Module	142
Filtering Possibilities for the Service Cards Module	143
Filtering Possibilities for the Service Graphs Module	144
Filtering Possibilities for the Service Health Module	145
A. Service Status	
Service Status	148
B. Restarting Tomcat	
Restarting the Servlet Engine	150
To Restart the Servlet Engine from the SIP Administration Pages	150
To Restart the Servlet Engine from Outside of SIP	150
C. Working with XML	
Rules for Direct Editing of XML Files	154
Backing Up XML Files	154
Understanding Editing Permission on XML Files	154

Contents

Validating XML Files	155
Avoiding Loss of Changes.....	156
Index	157

Support

Please visit the HP OpenView web site at:

<http://www.managementsoftware.hp.com/>

This web site provides contact information and details about the products, services, and support that HP OpenView offers.


You can also go directly to the support web site at:


<http://support.openview.hp.com/>


HP OpenView online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valuable support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and log in. Throughout the site, access levels are indicated by the following icons:

 HP Passport

 Active contract

 Premium contract

To find more information about access levels, go to the following URL:

http://support.openview.hp.com/access_level.jsp

To register for an HP Passport ID, go to the following URL:

<https://passport.hp.com/hpp2/newuser.do>

1 How OVO/OVSN Works with SIP

The OpenView Operations Module for SIP

HP OpenView Operations (OVO) is a central operations and problem-management product for distributed multi-vendor environments. OVO provides:

- Central message-management for consolidation, simplification and automation of message processing.
- Central monitoring for proactive problem resolution.
- Central problem-management for problem notification, resolution, and tracking.
- Central control for efficient management.

If OVO is configured and running within your management environment, Service Information Portal (SIP) can display information from the OVO message database through the OVO Messages module.

For information about configuring OpenView Operations, itself, see the documentation that comes with OVO (or access the web site:

http://ovweb.external.hp.com/lpe/doc_serv).

In SIP, you control access to three possible layers of information within the OVO Messages module:

- **Summary**
This is a high-level list of the type and number of messages received within one or more message categories. These categories correspond to the standard OVO database's message attributes: Nodes, Services, Objects, Applications, and Message Groups. If you have configured *custom* OVO message attributes, they cannot be used as SIP message categories (custom attributes are displayed on the Details Page). The messages are filtered according to your SIP configuration.
- **Drill-Down to List Page**
Accessed through links on the Summary page, the summary is a complete list of messages received within a specific category.
- **Drill-Down to Details Page**
Accessed through links on the List page, the list presents all known information about a specific message, including *custom* attributes.

For more information, see "OVO Messages Module" on page 35.

The Service Navigator Modules for SIP

HP OpenView Service Navigator (OVSN) -- an add-on product for OVO -- enables you to manage an IT environment while focusing on the IT services. The integration of OVSN into Service Information Portal (SIP) provides up-to-date information to your customers about their managed services. The services must be defined and configured within OVSN before they can be displayed in the SIP modules.

For more information about configuring services within OpenView Service Navigator, see the *HP OpenView Service Navigator Concepts and Configuration Guide* (or access the web site: http://ovweb.external.hp.com/lpe/doc_serv).

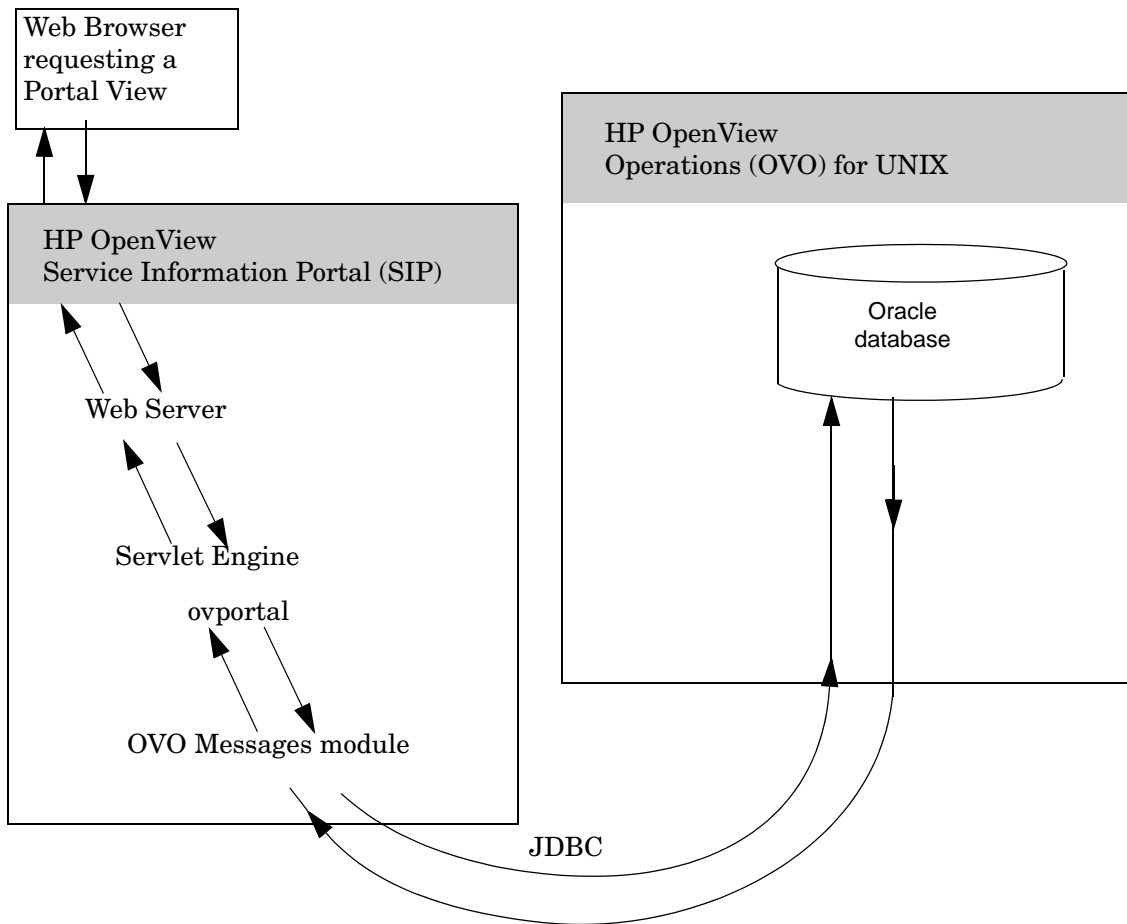
The following OVSN modules are included in SIP:

- **Service Browser module**
A table-like view of services, providing links to details about specific services. See Chapter 4, “The Service Browser Module,” on page 53 for more information.
- **Service Cards module**
A business-card view of services, providing links to details about specific services. See Chapter 5, “The Service Cards Module,” on page 63 for more information.
- **Service Graph module**
Hierarchical view of the OVSN configuration settings for a particular service. See Chapter 6, “The Service Graph Module,” on page 73 for more information.
- **Service Health module**
A gauge-like view that shows aggregated service status and provides links to details about specific services. See Chapter 7, “The Service Health Module,” on page 85 for more information.
- **Custom Service modules**
Create your own presentation of OVSN data. This requires that you write your own XSL style sheet. See Chapter 8, “Creating Your Own Service Modules,” on page 97 for more information.

Communication Paths Between OVO/OVSN and SIP

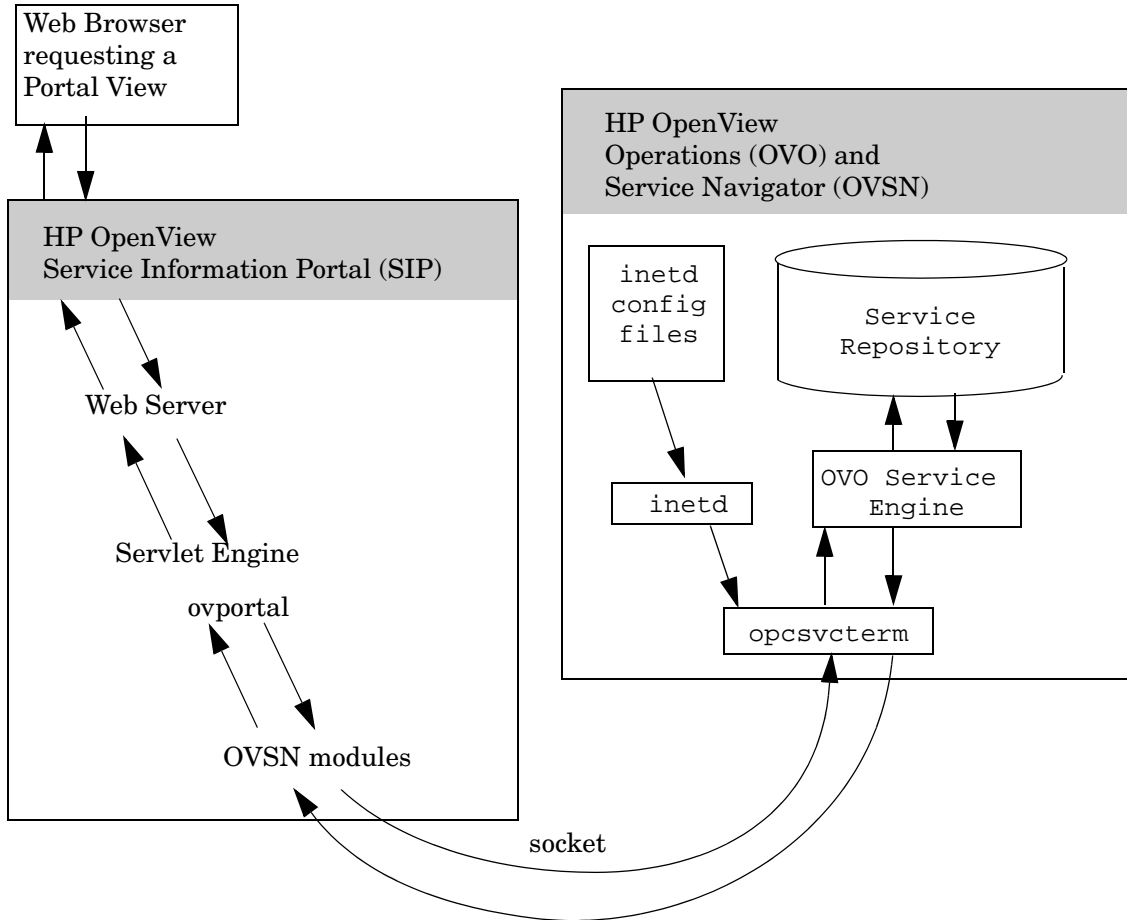
The following diagram illustrates the processes involved in communicating data to SIP from OpenView Operations for UNIX:

Figure 1-1 **Communication Process for the OVO Messages Module Displaying Data from OVO for UNIX**



The following diagram illustrates the processes involved in communicating data to SIP from OpenView Service Navigator:

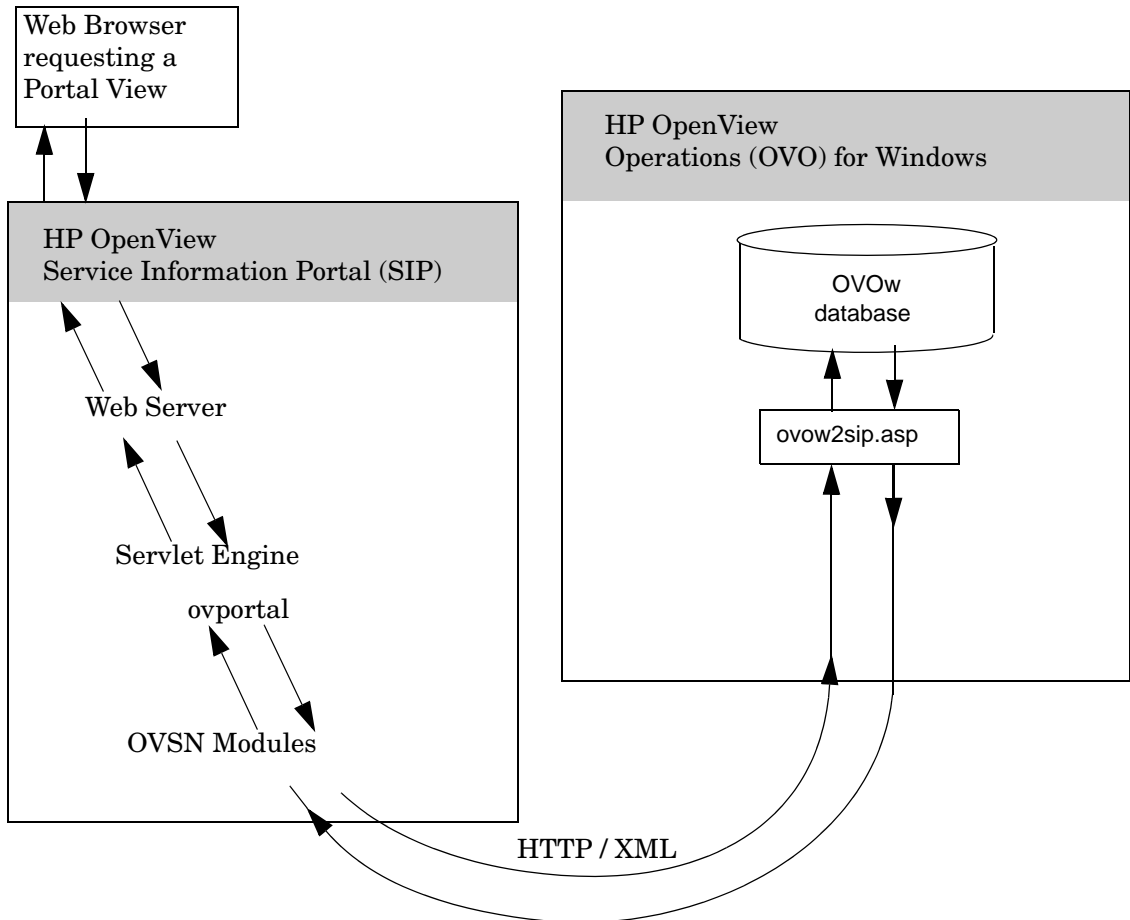
Figure 1-2 **Communication Process for the OVSN Modules**



Communication Paths Between OVO/OVSN and SIP

The following diagram illustrates the process involved in communicating data to SIP from OpenView Operations for Windows, including data for the OVSN modules:

Figure 1-3 **Communication Process for the OVO Messages Module Displaying Data from OVO for Windows**



Installation of the Modules

SIP can run on Windows, HP-UX, or Solaris and can communicate with multiple OVO management servers running on any combination of Windows, HP-UX, and Solaris. You configure each SIP module to communicate with one of your OVO management servers.

The OVO Messages module and OVSN modules are automatically installed with the Service Information Portal software. If you have not already installed SIP, see the *SIP Installation Guide* ([SIP_Install_Guide.pdf](#)) for SIP installation instructions, and OVO/OVSN version and patch requirements.

Before using the modules, you must configure SIP and OVO to communicate with each other. See “Establishing Communication Between OVO/OVSN and SIP” on page 20.

2 Configuration Steps

Establishing Communication Between OVO/OVSN and SIP

To establish communication between SIP and your OVO management servers, (and optionally, to establish communication between SIP and your OpenView Service Navigator program running on your OVO management server) you need to take the following steps.

SIP can run on Windows, HP-UX, or Solaris and can communicate with multiple OVO management servers running on any combination of Windows, HP-UX and/or Solaris.

On each OVO management server

NOTE

Verify that you are using a version of HP OpenView Operations that is supported by SIP, see the *SIP Installation Guide* ([SIP_Install_Guide.pdf](#)) for the list of supported product versions.

On UNIX

Service Information Portal communicates with OVO Unix through the listener service and with Service Navigator through the `opcsvcterm` service.

IMPORTANT

Because the `opcsvcterm` service provides read-write access to the OVO/SN database, it is recommended that access to this service be limited to trusted hosts with the need to access it, such as the SIP server.

1. In an ASCII or XML editor, open the `/etc/services` file on each OVO management server.
2. Verify that the following lines are present:
 - *For OVO:*
`listener 1521/tcp #Oracle Net8 listener`

- *For OVSN:*

```
opcsvcterm 7278/tcp # Service engine remote access
```

If not, consult the documentation that comes with OVO and OVSN to determine how to configure remote access settings. Note that the port numbers specified in the SIP Configuration Editor (“On the SIP Server” on page 22) and in this `/etc/services` file must be identical.

3. In an ASCII or XML editor, on each OVO/SN management server, open the `/etc/inetd.conf` file.
4. Verify that the following line is present, and if not then add it:

- *For OVSN:*

```
opcsvcterm stream tcp nowait root /opt/OV/bin/OpC/opcsvcterm opcsvcterm
```

5. Trigger the `inetd` service to re-read the configuration files.

- *HP-UX:* As superuser, run: `/etc/inetd -c`
- *Solaris:* Send the hangup signal to the currently running `inetd` process.
Run: `kill -1 current_inetd_PID`

6. If you are running OVO 6.0.x, install the following patch:

- *HP-UX:* PHSS_24264 or later.
- *Solaris:* ITOSOL_00101 or later.

On Windows

SIP comes with a script that enables OVO Windows to respond to SIP’s requests for data. The script must be placed in a virtual directory on the web server where OVO Windows is running.

1. Provide to the OVO Administrator the following script.

```
Windows: %SIP_HOME%\cgi-bin\WindowsNT\ovow2sip.asp
```

2. *To the OVO Administrator:* Place the script into a virtual directory on the web server where OVO Windows is running. You can put the script in the OVO Web directory or create a new virtual directory and place it there.
3. *To the OVO Administrator:* Provide the SIP Administrator with the following information:

- The path to the virtual directory into which the `ovow2sip.asp` script has been placed. For example, if the script is located in the OVO Web virtual directory, indicate that the path is `OVOWeb/ovow2sip.asp`.
- The port on which the web server is running on the OVO Windows machine, and an indication of whether the web server is a secure web server.
- The user and password, if the virtual directory requires authentication.

On each OVSN management server

NOTE

Verify that you are using a version of HP OpenView Service Navigator that is supported by SIP, see the *SIP Installation Guide* (`SIP_Install_Guide.pdf`) for the list of supported product versions.

If OpenView Service Navigator is installed and configured on your OVO/OVSN management server:

1. Log on as `root` on the system where Service Navigator is installed.
2. If you are running OVO 6.0x, install the following patch:
 - *Solaris*: `ITOSOL_00117` or later.
 - *HP-UX*: `PHSS_24741` or later.
3. Ensure that the OVSN processes are running, at the command prompt, type:

```
/opt/OV/bin/OpC/opcsv -start
```

On the SIP Server

Three configuration steps must be completed on the SIP server:

- “Management Station Settings” on page 23.
- “SIP Customer Model Definitions” on page 25.
- “Role Configuration Settings” on page 26.

Management Station Settings

To establish communication between SIP and OVO/OVSN, you need to define the OVO management server configuration settings:

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

Windows: 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 OVO management server, right-click the title Management Stations, select New, and type in the fully-qualified host name of the OVO management server.

To add OVO/OVSN settings to an existing management station, right-click the name of the management station, and select Properties.

4. Navigate to the OVO tab.
5. Select OVO Is Installed On This System.

NOTE

OpenView operations for Windows supports Service Navigator capabilities. Therefore, this can always be enabled on OpenView Operations for Windows if these capabilities are desired.

6. If the same is true for Service Navigator, select SN Is Installed On This System.
7. Default values for the remaining fields are entered automatically. If you have changed any of the default settings on your OVO management server, enter the current settings.

For OVO UNIX:

- OVO DB Port: The Oracle TCP port to connect to for remote access to the OVO database. The default port is 1521.
- OVO DB User: The Oracle user name for SIP to use when logging into the Oracle database.

- **OVO DB Password:** The password for SIP to use when logging into the Oracle database.
- **OVO DB Name:** The Oracle database name (SID) that is assigned during installation of OVO. The default is `openview`.
- **OVO DB Host:** The host on which the Oracle database resides. Leaving this field blank indicates that the OVO Management Server is the Oracle database host. Fill this in only if OVO was configured for remote database access.
- **Min Connections:** Minimum number of connections to store in a pool shared by all OVO Messages module instances when connecting to Oracle.
- **Max Connections:** Maximum number of connections to store in a pool shared by all OVO Messages module instances when connecting to Oracle.
- **Max Cached Statements:** Maximum number of statements to cache in the OVO Messages module.
- **Service Navigator Port:** The TCP port to use for communication with Service Navigator. The default port is 7278.

For OVO Windows:

- **OVO Language:** The language used by the OVO Windows server. As of OVO/W 7.21, only English and Japanese are supported.
- **OVO URL:** The path to the virtual directory into which the `ovow2sip.asp` script has been placed by the OVO Administrator. For example if the script is located in the OVO Web virtual directory, the OVO URL entry would be `OVOWeb/ovow2sip.asp`.
- **Web Server Port:** The port on which the web server is running on the OVO Windows machine.
- **Https Enabled:** Indicates that you are running a secure web server. The default value of the Web Server Port changes to 443 when this option is selected.
- **User:** A user name required to access a virtual directory that requires authentication. If the virtual directory entered in the OVO URL field is protected, the User and Password entries are required.
- **Password:** The password required for the above user.

8. Click [OK] when you are finished making changes.
9. Repeat from step 2 for each OVO management server with which SIP should communicate.
10. Click [OK] to return to the main SIP Configuration Editor window. You must save your configuration changes by clicking [Save] on the main window.

TIP

If you are configuring SIP to communicate with multiple OpenView Operations systems, the system name appearing closest to the top of the Management Stations list in the SIP Configuration Editor window becomes the *default* OpenView Operations system.

SIP Customer Model Definitions

You *must* populate the SIP Customer Model with <Organization> elements that contain one or more <NodeList> and <ServiceList> elements to enable the OVO/OVSN modules. OVO/OVSN modules display data about all <Organization> elements assigned to the current SIP Role.

Viewing the Current SIP Customer Model

To view your current SIP Customer Model configuration:

1. Go to the SIP Administration Pages by logging into the SIP portal 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 Summary Report segment.
4. Click [View] to see a list of all nodes and services currently configured in your SIP Customer Model.

Review the Organizations and their assigned nodes and services to ensure that the data sources you need for your OVO/OVSN modules are listed:

- <NodeList> elements must contain all nodes whose messages you wish to display in the OVO Message module. Only those messages from nodes that pass through the currently assigned Management Data Filter are displayed.

- <ServiceList> elements must contain every top-level OVSN service whose data you wish to include in the OVSN modules.

5. Click [Help] if you need more information.

If within each SIP <Organization> you already have the appropriate <NodeList> and <ServiceList> elements, continue with “Role Configuration Settings” on page 26.

If you need to create or modify <NodeList> and <ServiceList> elements within your SIP <Organization> definitions, continue with “Creating NodeList and ServiceList Elements” on page 26.

Creating NodeList and ServiceList Elements

If you need to add nodes or services, do so now:

- For detailed instructions about developing the required <NodeList> and <ServiceList> elements for the OVO/OVSN modules, turn to Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115.
- For background information about setting up the SIP Customer Model, see the *SIP Deployment and Integration Guide*, (SIP_Deployment_Integration.pdf), “Segmenting Data by Customer Organization.”

DO NOT CONTINUE until your SIP Customer Model <Organization> elements contain the required <NodeList> and <ServiceList> elements.

Role Configuration Settings

To enable SIP roles to display data in the OVO/OVSN modules, you need to establish the Management Data filter settings for each role:

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

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

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

2. Navigate to your Role definitions (other than SIP Administrator).

3. For each role that needs access to the OVO Messages module or OVSN modules, right-click the role name and select Properties.

4. Navigate to the `Management Data` tab and make the appropriate selection. The `Management Data` filter controls the flow of data from your OVO management servers into the OVO/OVSN modules. See Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 if you need more information.
5. Repeat from step 3 for each role that needs access to OVO Messages module and/or OVSN modules.
6. Save your changes and exit the SIP Configuration Editor.

SIP Distribution Model

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

- Web Browser Tier
- Web Server Tier
- 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 OVO management server communication can also go through a firewall, if desired. The ports that need to be opened through the firewall to gather data for the OVO Messages module and OVSN modules are specified through the SIP Configuration Editor program, Management Station configuration settings, on the OVO tab (see “On the SIP Server” on page 22).

Table 2-1 OVO Messages Module Port Requirements

Server Product	Protocol	Default Port	Configuration Location
OVO for UNIX	OVO DB	1521	On the SIP server, use the SIP Configuration Editor to configure protocol and port (OVO tab for management stations).
OVO for Windows	http	80	
	https	443	

Table 2-2 OVSN Modules Port Requirements

Protocol	Default Port	Configuration Location
opsvcterm	7278	On the SIP server, use the SIP Configuration Editor to configure protocol and port (OVO tab for management stations).

Running in Languages Other Than English

Any language that can be displayed within the UTF-8 codeset can be displayed through SIP.

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 UTF-8 Data From OVO/OVSN

For OVO UNIX, no additional configuration tasks are required to allow SIP to access non-ASCII data from OVO/OVSN (provided that you have completed the steps in the previous sections of this chapter).

For OVO for Windows

If an OVO/W management station is running in a language that uses non-ASCII characters, you need to change the OVO Language attribute for the OVO/W management station configured in SIP.

After the OVO Language attribute is set to Japanese, OVO message data is automatically translated into Shift JIS characters by the portal.

1. On the SIP server, open the SIP Configuration Editor.
2. Select your OVO for Windows management station and navigate to the OVO tab. Enter the appropriate OVO Language attribute for your OVO management station.
3. Click [OK] to return to the main SIP Configuration Editor window. You must save your configuration changes by clicking [Save] on the main window.

Getting Japanese Node Lists and Service Lists from OVO for Windows

When setting up a SIP customer model that segments the OVO/W data by customer, you may need to determine which nodes and services to add to the SIP customer model (explained in “Manually Creating NodeList and ServiceList Elements” on page 120). To generate NodeList elements and ServiceList elements in Japanese, do the following:

- From a web browser on the OVO/W management server, run the `ovow2sip.asp` script with the `lang=japanese` parameter, as shown below:
 - `ovow2sip.asp?lang=japanese&query=ListNodes`
 - `ovow2sip.asp?lang=japanese&query=ListServices`
 - `ovow2sip.asp?lang=japanese&query=ListServices&includeDisplay=true`

Secure Socket Layer (SSL) Support

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

Running the OVO/OVSN Modules in a Wireless Environment

If you configure SIP for portals that are accessed through wireless devices, the OVO Messages module and two of the OVSN modules are available and displayed as shown in Figure 2-1 through Figure 2-3. 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 OVO Messages Module

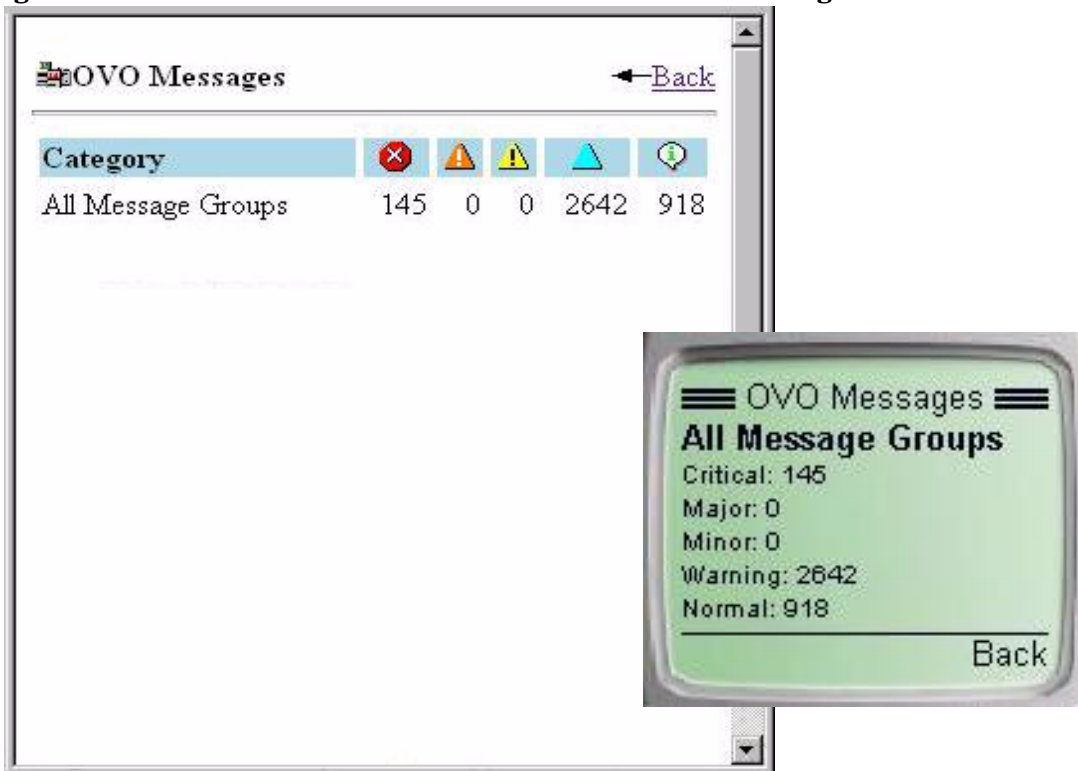


Figure 2-2 PDA and WML Format for the Service Cards Module

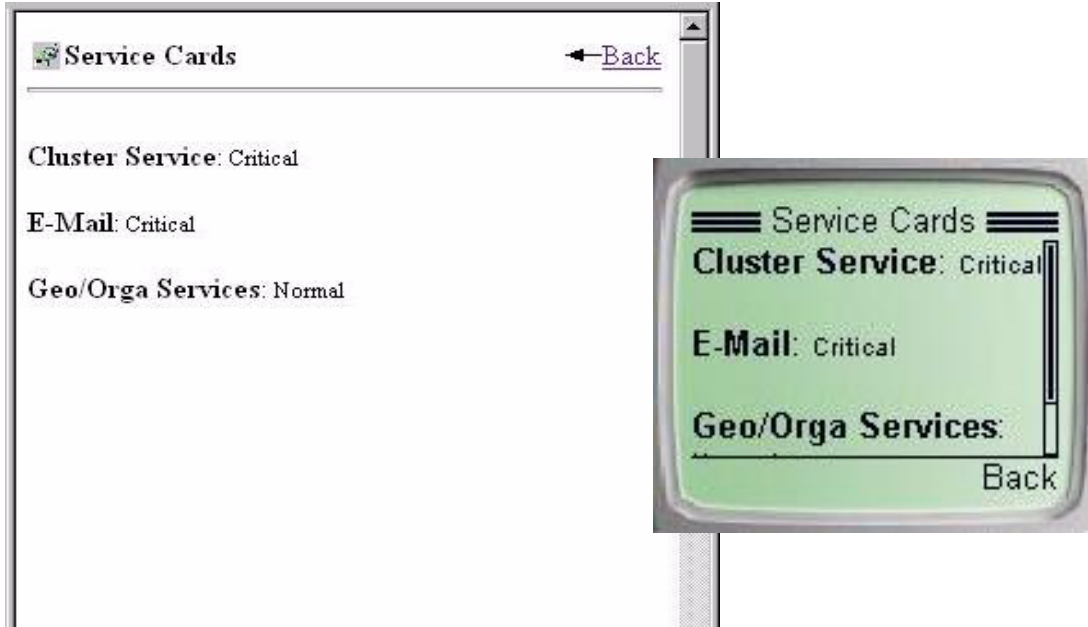
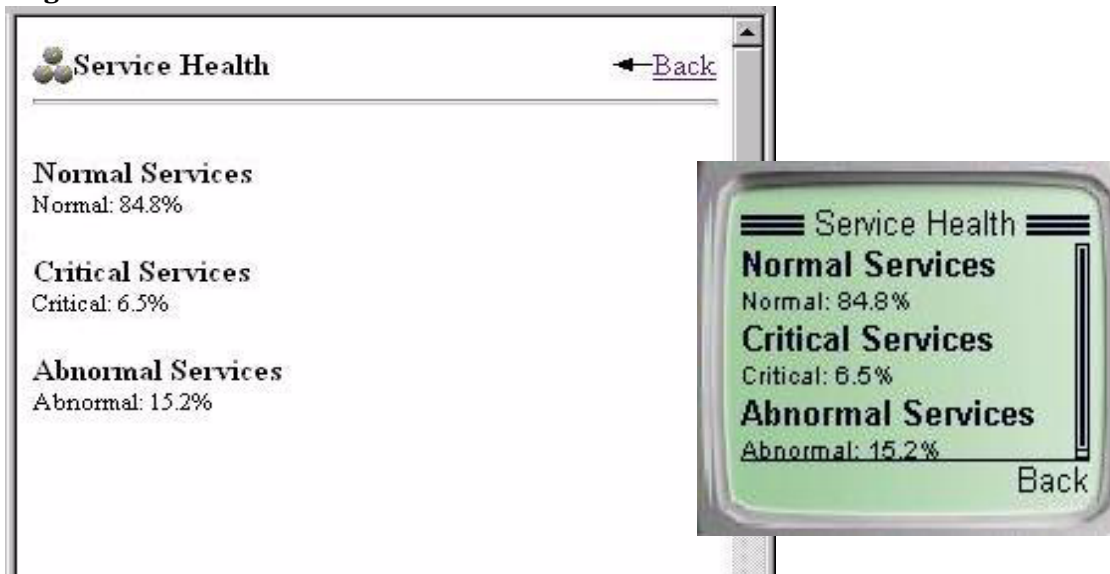


Figure 2-3 PDA and WML Format for the Service Health Module



Configuration Steps

Running the OVO/OVSN Modules in a Wireless Environment

3 OVO Messages Module

Using the OVO Messages Module

Use the OVO Messages module to display OVO messages through the SIP portal. Each message within the OVO message database has five standard OVO message attributes. SIP's OVO Messages module can be configured to group and display messages based upon any or all of these five standard attributes:

- Applications
- Message Groups
- Nodes
- Objects
- Services

Each message category displays a list of the OVO database elements in that attribute category, along with the number of messages received within each severity level for that element.

Click on any link to display details about the received messages, such as: date, time, source of the message, message description, and any custom message attributes you configured.

NOTE

You can further narrow down which messages are displayed in the OVO Messages module by writing filters (see Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 and Chapter 10, “Display Filtering for OVO/OVSN,” on page 135 for more information).

Adding the OVO Messages Module

To insert the OVO Messages 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 OVO Messages from the Select Module to Add list box, and click [Add], or
 - Click [Edit] to access the Edit Column page. Insert the OVO Messages 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 OVO Messages module is inserted into your *PortalView.xml* file.

- If you want to modify this module instance, turn to “Editing the OVO Messages Module” on page 38.

If you have more than one OVO/OVSN management server configured to communicate with SIP, the first OVO/OVSN management server listed in the SIP Configuration Editor is accessed (“Establishing Communication Between OVO/OVSN and SIP” on page 20). You can change this selection through Service Browser - Edit window.
- If you want to change the default module, see “Relevant Files” on page 50.

TIP

If you want to add a module to the list of available modules, see “Relevant Files” on page 50. You can create and add another instance of any module.

Editing the OVO Messages Module

Using the OVO Messages - Edit page

You can easily modify the OVO Messages 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 OVO Messages module, click the edit button:



4. Make any desired changes. Click `[Help]` for specific instructions:

- Only one OVO management server communicates with any OVO Messages module. This module instance gathers messages from the first OVO management server listed in the SIP Configuration Editor, *unless* you specify from which OVO management server to gather the messages. Choose from the OVO management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20 if you need to add one).
- Choose the message categories to be displayed in this module (Applications, Message Groups, Nodes, Objects, and/or Services). The messages displayed in the OVO Messages module are only those messages received from *either* (union of):
 - Nodes that pass the `<NodeList>` of the Management Data filter assigned to the current SIP Role, or
 - Services that pass the `<ServiceList>` of the Management Data filter assigned to the current SIP Role.
- Remove message categories from this module.
- Change the display order of message categories.

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.

Directly editing the PortalView.xml file

TIP

For the following adjustments, you must edit the XML file. It is recommended that you use the *OVO Messages - 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 modify message categories or create new message categories, see “Editing OVO Message Categories” on page 45.

To directly modify the XML code for an OVO Messages 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: %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.ovo"

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.ovo"
  display="yes"
  help="/OvSipDocs/C/help/OVO/OVOView.html"
  id="module13"
  rollupState="down"
  title="OVO Messages">
</ModuleInstance>
```

4. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags, if the `<OVO>` element does not already exist. Alternately, copy the contents from the OVO Messages module's default XML file (see "Relevant Files" on page 50):

```
<OVO>
  <Server hostname="your_OVO_server"/>
  <MessageCategoryRef href="applications"/>
  <MessageCategoryRef href="messageGroups"/>
  <MessageCategoryRef href="nodes"/>
  <MessageCategoryRef href="objects"/>
  <MessageCategoryRef href="services"/>
</OVO>
```

Each `<MessageCategoryRef>` element represents one OVO message category definition. See "Editing OVO Message Categories" on page 45 for more information about message categories.

You can make the following changes by directly editing the OVO Messages module's XML code:

5. To change the title of this OVO Messages module instance, change the `title` attribute (`<ModuleInstance title="new title">`).
- To change the title of all OVO Messages modules, change the `title` attribute in the registration file, see "Relevant Files" on page 50.
6. To assign a different HTML file to the help [?] button for this OVO Messages module instance, change the `help` attribute (`<ModuleInstance help=" " >`). See "Writing Your Own Online Help" on page 108.
7. Only one OVO management server communicates with any OVO Messages module. This module instance gathers messages from the first OVO management server listed in the SIP Configuration Editor, *unless* you specify from which OVO management server to gather the

messages. Choose from the OVO management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20).

To specify an OVO management server, add the `hostname` attribute (`<Server hostname="your_OVO_server" />`) with the fully-qualified hostname of your OVO management server.

8. Add or remove the message categories to be displayed in this OVO Messages module instance (`<MessageCategoryRef href=" " />` lines). The `href` values must match the `id` attribute values in your `OVOMessageCategory.xml` files.

The messages displayed in the OVO Messages module are only those messages received from *either* (union of):

- Nodes that pass the `<NodeList>` of the Management Data filter assigned to the current SIP Role, or
 - Services that pass the `<ServiceList>` of the Management Data filter assigned to the current SIP Role.
9. Change the display order of message categories by moving `<MessageCategoryRef href=" " />` lines.
 10. After making the desired changes, save the file.
 11. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
 12. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Establishing Global Settings for OVO Messages Modules

There are global settings that affect all message categories in every instance of an OVO Messages module in all *PortalView.xml* files.

OVConfig.xml/dtd

The following is an example of the possible settings in the XML code of the *OVConfig.xml* file:

```
<OVModuleConfig
    labelCacheRefresh="60"
    messageDisplayCount="50"

    editStyle="ovoEdit"
    summaryStyle="ovo"
    listStyle="ovo"
    detailStyle="ovo"
/>
```

The following two attributes control the behavior of all OVO Messages modules:

- `labelCacheRefresh` (zero or greater)
The number of minutes to wait before retrieving `label` values for `MessageGroups`, `Nodes`, and `Services` from the OVO database. Labels are retrieved for display within the OVO Messages module. If no `label` can be found, the value in the `id` field is used until the next refresh cycle.
- `messageDisplayCount` (zero or greater)
The default number of messages to show in the message list page. This can be overridden for a particular category by setting the `displayCount` attribute within a message category definition file.

The four style-related attributes are provided in case you wish to write and provide your own XSL style sheet. Currently, only the `listStyle` attribute has more than one setting to choose from:

- `summaryStyle`
The simplified stylesheet name to use to format the summary page.

- `listStyle`
The simplified stylesheet name to use to format the message list page:
`ovo` = the background color is white for the entire message table
`ovoColored` = the background color of each row matches the severity status (for example, critical messages have a red background)
- `detailStyle`
The simplified stylesheet name to use to format the message detail page.
- `editStyle`
The simplified stylesheet name to use to format the edit page.

See the comments in the `OVOConfig.dtd` and `OVOConfig.xml` files for more information.

To modify the `OVOConfig.xml` file:

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 file as a reference:
 - *Windows:*
`%SIP_HOME%\conf\share\modules\ovo\OVOConfig.dtd`
 - *UNIX:*
`/opt/OV/SIP/conf/share/modules/ovo/OVOConfig.dtd`
3. In an ASCII or XML editor, open the `OVOConfig.xml` file, stored in the same location as the `OVOConfig.dtd` file.
4. Make your changes and save the file.
5. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
6. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

SIP Configuration Editor

Configuration settings for each OVO management server are entered into the SIP Configuration Editor, Management Stations, OVO tab. A few settings for the OVO for UNIX management servers have a global affect on all instances of the OVO Messages module communicating with an OVO for UNIX management server. These settings are useful when troubleshooting issues and tuning performance:

- **Min Connections**
The minimum number of connections to store in a pool shared by all OVO module instances when connecting to Oracle.
- **Max Connections**
The maximum number of connections to store in a pool shared by all OVO module instances when connecting to Oracle.
- **Max Cached Statements**
The maximum number of statements to cache in the OVO module.

See the online [Help] within the SIP Configuration Editor (“Management Station Settings” on page 23) or the `conf\share\stations\ovoConfig.dtd` and `mgmtStations.dtd` files for more information.

Editing OVO Message Categories

Each message category for the OVO Messages module must be based upon *one* of the standard five OVO message attributes:

- Applications
- Message Groups
- Nodes
- Objects
- Services

SIP provides a simple message category definition file for each of the five standard OVO message attributes. You can modify the provided files or add your own. Each SIP message category must be defined in an *OVOMessageCategory.xml* file and be listed in the *OVOCategoryIndex.xml* file before being displayed in OVO Messages modules. For more information, see:

- “Modifying Existing Message Categories” on page 45
- “Creating New Message Categories” on page 47
- “Eliminating a Message Category” on page 48

Modifying Existing Message Categories

To modify an existing OVO message 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 file as a reference:

- *Windows:*

```
%SIP_HOME%\conf\share\modules\ovo\OVOMessageCategory.dtd
```

- *UNIX:*

```
/opt/OV/SIP/conf/share/modules/ovo/OVOMessageCategory.dtd
```

3. In an ASCII or XML editor, open the *OVOMessageCategory.xml* file that defines the message category you wish to modify. These definition files are stored in the following location:

- *Windows:*
%SIP_HOME%\conf\share\modules\ovo\
- *UNIX:*
/opt/OV/SIP/conf/share/modules/ovo/

4. Following the rules as explained in the *OVOMessageCategory.dtd* file, make the desired modifications.

The following is an example of the possible configuration settings in the XML code for an OVO message category file:

```
<OVOMessageCategory
  id="applications"
  title="All Applications"
  olderThanXMinutes=" "
  newerThanXMinutes=" "
  showDetails="(yes|no)"
  displayCount=" " >

  <attributeSelection />
</OVOMessageCategory >
```

- *id*= (required) the name of this message category. This name must be referenced in the OVO Messages module to display the category.
- *title*= (required) the text that appears in the title bar of the message category when displayed within any instance of the OVO Messages module and displayed in the OVO Categories selection list on the Edit page.
- *olderThanXMinutes*= (optional) filters out the newer messages.
- *newerThanXMinutes*= (optional) filters out the older messages.
- *showDetails*= (optional) specifies whether or not “detail” links are displayed for each message in the List page of the OVO Messages module. This link provides access to additional information about the message.
- *displayCount*= (optional) specifies the maximum number of messages that are displayed in the List page.

- *attributeSelection=* (required) specifies which one of the five OVO database attributes is used for grouping and displaying messages in this category: *ServiceSelection*, *NodeSelection*, *MessageGroupSelection*, *ObjectSelection*, or *ApplicationSelection*. This element can optionally be used to filter messages within the category, see Chapter 10, “Display Filtering for OVO/OVSN,” on page 135.
5. Close and save the *OVOMessageCategory.xml* file.
 6. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
 7. Log into the SIP portal as the appropriate user to ensure that you have the desired results.
 8. Locate or add the OVO Message module and examine the category that you changed.

Creating New Message Categories

Each OVO message category is defined in a separate XML file. SIP ships with predefined message category files for the five standard OVO message attributes. To create multiple versions of the same message category, each with unique configuration settings, do the following:

1. Before you start, navigate to the following directory, and open the following file as a reference:
 - *Windows:*
`%SIP_HOME%\conf\share\modules\ovo\OVOMessageCategory.dtd`
 - *UNIX:*
`/opt/OV/SIP/conf/share/modules/ovo/OVOMessageCategory.dtd`
2. In the same directory, copy and rename one of the following as a starting point for your new message category definition file:
 - `applications.xml`
 - `messageGroups.xml`
 - `nodes.xml`
 - `objects.xml`

- `services.xml`
3. In an ASCII or XML editor, open the new file.
You *must* provide a unique `id` attribute value.
Follow the directions in “Modifying Existing Message Categories” on page 45.
 4. Following the rules as explained in the `OVOMessageCategory.dtd` file, make the desired modifications.
 5. Close and save the new `OVOMessageCategory.xml` file.
 6. In an ASCII or XML editor, open the `OVOCategoryIndex.xml` file that contains the list of all valid message categories available for display in your portal view. This file must be stored in the following location:
 - *Windows:*
`%SIP_HOME%\conf\share\modules\ovo\`
 - *UNIX:*
`/opt/OV/SIP/conf/share/modules/ovo/`
 7. Add a `<MessageCategory>` element with the `href` attribute pointing to the newly created `OVOMessageCategory.xml` file. If you are including a path statement, SIP interprets the path as relative to the location of the `OVOCategoryIndex.xml` file.

```
<MessageCategory href="nodes.xml" />
```
 8. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
 9. Log into the SIP portal as the appropriate user to ensure that you have the desired results.
 10. Edit or add the OVO Message module and examine the category that you added.

Eliminating a Message Category

1. In an ASCII or XML editor, open the `OVOCategoryIndex.xml` file:
 - *Windows:*
`%SIP_HOME%\conf\share\modules\ovo\`

- *UNIX:*
`/opt/OV/SIP/conf/share/modules/ovo/`
2. Delete the `MessageCategory` line referring to the SIP message category that you wish to eliminate.
 3. Close and save the `OVOCategoryIndex.xml` file.

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

4. In a browser, log into the portal to verify that the messages appear as desired.

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

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

- *Windows:*

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

- *UNIX:*

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

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

- *Windows:*

```
%SIP_HOME%\conf\share\modules\ovo\
```

- *UNIX:*

```
/opt/OV/SIP/conf/share/modules/ovo/
```

Relevant Files

The OVO Messages 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` & `ovoConfig.dtd` & `mgmtStations.xml`

This XML file configures communication between OVO management servers and SIP servers. See the “Establishing Communication Between OVO/OVSN and SIP” on page 20.

- `OVModuleRegistraton.dtd` & `OVRRegOVO.xml`

This XML file grants access to the OVO Messages module through the SIP framework so that it is available for your use. To add another instance of the OVO Messages module to the SIP module selection list, copy and rename the `OVRRegOVO.xml` and the `OVDDefaultOVO.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 150.

- `OVO.dtd` & `OVDDefaultOVO.xml`

This DTD defines the rules for configuring the OVO Messages module. The XML file contains the *default* OVO Messages module. This XML file determines the list of OVO message categories that appear in the OVO Messages module when using the [Add] or [Edit] button in the SIP user interface. The contents of the default file are inserted into your portal view file each time you use the [Add] button to insert the OVO Messages module.

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

- Directly edit the XML code in the `OVDDefaultOVO.xml` file, or
- Insert an OVO Messages 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 `OVDDefaultOVO.xml` file.

See “Editing the OVO Messages Module” on page 38 for more information.

- `OVOCategoryIndex.dtd/OVOCategoryIndex.xml`
This XML file specifies the list of message category definitions that are available through SIP. Any message categories requested within OVO Messages module instances, but not listed in this file, are ignored. (See the `OVOMessageCategory.dtd` file for more information.)

`OVMessageCategory.dtd/OVMessageCat.xml`

There is one XML file for each OVO message category that you wish to display in SIP portals. (See “Editing OVO Message Categories” on page 45.)

- `OVOConfig.dtd & OVOConfig.xml`
This DTD provides the rules for establishing the global settings that control the OVO Message module. These settings affect all message categories in every instance of the OVO Messages module in all *PortalView.xml* files. You can edit the XML file to indicate your choices. See “Establishing Global Settings for OVO Messages Modules” on page 42.
- `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/OVO/*.html`
This directory contains the Help topics for this module. The Help topics are accessed by clicking the [?] button on the module title bar. If you want to supply your own customized help files, see “Writing Your Own Online Help” on page 108.

Table 3-1 OVO Messages Module Files on the SIP Server

File Name	Windows Location <code>%SIP_HOME%\...</code>	UNIX Location <code>/opt/OV/SIP/...</code>
<code>mgmtStations.dtd</code>	<code>conf\share\stations\</code>	<code>conf/share/stations/</code>
<code>mgmtStations.xml</code>	<code>conf\share\stations\</code>	<code>conf/share/stations/</code>

Table 3-1 OVO Messages Module Files on the SIP Server (Continued)

File Name	Windows Location %SIP_HOME%\....	UNIX Location /opt/OV/SIP/....
ovoConfig.dtd	conf\share\stations\	conf/share/stations/
OVMModuleRegistration.dtd	registration\	registration/
OVRRegOVO.xml	registration\	registration/
OVO.dtd	conf\share\views\	conf/share/views/
OVDDefaultOVO.xml	registration\defaults\	registration/defaults/
OVOCategoryIndex.dtd	conf\share\modules\ovo\	conf/share/modules/ovo/
OVOCategoryIndex.xml	conf\share\modules\ovo\	conf/share/modules/ovo/
OVMMessageCategory.dtd	conf\share\modules\ovo\	conf/share/modules/ovo/
<i>OVMMessageCat.xml</i> *	conf\share\modules\ovo\	conf/share/modules/ovo/
OVOConfig.dtd	conf\share\modules\ovo\	conf/share/modules/ovo/
OVOConfig.xml	conf\share\modules\ovo\	conf/share/modules/ovo/
PortalView.dtd	conf\share\views\	conf/share/views/
<i>PortalView.xml</i>	conf\share\views\	conf/share/views/
*.html	htdocs\C\help\OVO\	htdocs/C/help/OVO/

* The files can be stored anywhere, but all references to those files must use path statements relative to this directory.

4 The Service Browser Module

Using the Service Browser Module

The Service Browser module displays a list of Service Navigator's currently defined top level services or selected services. To quickly locate and display a particular service or child service, use the Search feature. The order of services can be easily changed by clicking the column title:

- **Sort by Status column**

The first column on the left tells you how healthy your service currently is by displaying the current status. The status ranges from normal to warning, minor, major, and critical. Table A-1, "Service Status," on page 148 describes the meaning of each color and gives a detailed explanation of each status. Click on this column title to sort the table by status.

Clicking on the status icon in this column may or may not display the defined Service Properties (such as actions, propagation rules, and calculation rules as defined within OVSN). Access to Service Properties is configurable. By default, access to service details is disabled.

- **Sort by Label column**

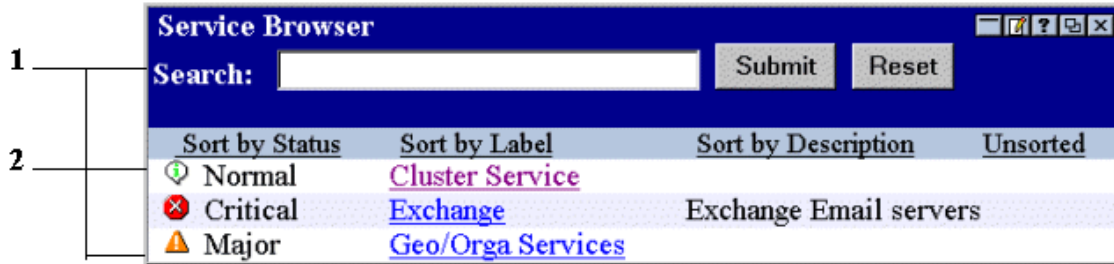
The second column from the left displays the label of the service. Click on this column title to sort the table by label. Clicking on an underlined service label displays the Service Browser with a list of the child services of this service.

- **Sort by Description column**

The third column from the left displays a description of the service. Click on this column title to sort the services by description.

Figure 4-1 shows an example of the Service Browser module.

Figure 4-1 **The Service Browser Module**



1. This section dedicated to searching and sorting services.

To use the Search feature, simply type in a text string and click [Submit].

The search tool searches all service labels. Text strings in lowercase matches words in any case. Search terms can be complete words, syllables, or single letters.

The results of the Search are presented in a Service Browser-like window with a list of all services whose label or name matches the search term.

2. This section displays a list of services.

The actual number and selection of services displayed in the Service Browser module depends upon the configuration settings of the Service Browser module (see “Editing the Service Browser Module” on page 57) and the configuration of filtering for the current SIP role (see Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 for information about filtering).

Adding the Service Browser Module

To insert the Service Browser 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 Service Browser from the Select Module to Add list box, and click [Add], or
 - Click [Edit] to access the Edit Column window. Insert the Service Browser 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 Service Browser module is inserted into your *PortalView.xml* file.

- If you want to modify this module instance, turn to “Editing the Service Browser Module” on page 57.

If you have more than one OVO/OVSN management server configured to communicate with SIP, the first OVO/OVSN management server listed in the SIP Configuration Editor is accessed (“Establishing Communication Between OVO/OVSN and SIP” on page 20). You can change this selection through Service Browser - Edit window.

- If you want to change the default module, see “Relevant Files” on page 61.

TIP

If you want to add a module to the list of available modules, see “Relevant Files” on page 61. You can create and add another instance of any module.

Editing the Service Browser Module

Using the Service Browser - Edit Page

You can easily modify the Service Browser 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 Service Browser module, click the edit button:



4. Make any desired changes. Click `[Help]` for specific instructions:
 - Only one OVO management server communicates with any OVSN module. This module instance gathers data from the first OVO management server listed in the SIP Configuration Editor, *unless* you specify from which OVO management server to gather the data. Choose from the OVO management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20 if you need to add one).
 - If you want to allow access to Service Properties (such as actions, propagation rules, and calculation rules as defined within OVSN), check `Allow details view drill-down`.
 - Choose the services to be displayed within the Service Browser module (either all top-level services or selected services).

If you want to select specific services, use shift-click or control-click to move groups of services from side to side. Only those services that passed the `ServiceList` of the Management Data filter assigned to the current SIP Role are displayed (see “SIP Customer Model Definitions” on page 25 and “Role Configuration Settings” on page 26).

See “Filtering Possibilities for the Service Browser Module” on page 142 for more information.

- Remove a service from the Service Browser module’s displayed list.
 - Change the display order of selected services.
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.

Directly Editing the PortalView.xml File

TIP

For the following adjustments, you must edit the XML file. It is recommended that you use the *Service Browser - 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 Service Browser 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: %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.service.SvcListServlet  
"
```

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.service.SvcListServlet"  
  display="yes"  
  help="/OvSipDocs/C/help/VPNavigator/browserView.html"  
  id="module8"  
  rollupState="down"  
  title="Service Browser">  
</ModuleInstance>
```

4. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags, if the `<ServiceBrowser>` element does not already exist. Alternately, copy the contents from the Service Browser module’s default XML file (see “Relevant Files” on page 61):

```
<ServiceBrowser details="no" server="your_OVO_server">  
  <ServiceRef name="email"/>  
  <ServiceRef name="email_exch"/>  
  <ServiceRef name="geo_orga"/>  
</ServiceBrowser>
```

Each `<ServiceRef>` element represents one line in the Service Browser. See the comments in the `OVSvcModules.dtd` file for more information about the correct XML syntax (see “Relevant Files” on page 61).

5. To change the title of this Service Browser module instance, change the `title` attribute (`<ModuleInstance title="new title">`).

To change the title of all Service Browser modules, change the `title` attribute in the registration file, see “Relevant Files” on page 61.

6. To assign a different HTML file to the help [?] button for this Service Browser module instance, change the `help` attribute (`<ModuleInstance help=" " >`). See “Writing Your Own Online Help” on page 108.

7. Only one OVO management server communicates with any OVSN module. This module instance gathers data from the first OVO management server listed in the SIP Configuration Editor, *unless* you specify from which OVO management server to gather the data. Choose from the OVO management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20).

To specify an OVO management server, add the `hostname` attribute (`<Server hostname="your_OVO_server" />`) with the fully-qualified hostname of your OVO management server.

8. To display specific services, replace the values of the `name` attributes in the `<ServiceRef name=" " />` lines with your service names as defined in your Service Navigator database. For example:

```
<ServiceRef name="email" />
```

Each service that you list must pass the `<ServiceList>` of the Management Data filter assigned to the current SIP Role before that service’s data is displayed in the Service Browser module. To display *all* top-level services that pass the Management Data filter assigned to the current SIP Role, delete all `<ServiceRef name="xx" />` lines. See Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 for more information about Management Data filters.

9. Place the `<ServiceRef name="xx" />` lines in the order in which they should be displayed in the browser.
10. Save the XML file.
11. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
12. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Relevant Files

The Service Browser 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` & `ovoConfig.dtd` & `mgmtStations.xml`

This XML file configures communication between OVO/OVSN management servers and SIP servers. See the “Establishing Communication Between OVO/OVSN and SIP” on page 20.

- `OVModuleRegistraton.dtd` & `OVRRegSvcBrowser.xml`

This XML file grants access to the Service Browser module through the SIP framework so that it is available for your use. To add another instance of the Service Browser module to the SIP module selection list, copy and rename the `OVRRegSvcBrowser.xml` and the `OVDDefaultSvcBrowser.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 150.

- `OVSvcModules.dtd` & `OVDDefaultSvcBrowser.xml`

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

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

- Directly edit the XML code in the `OVDDefaultSvcBrowser.xml` file, or
- Insert a Service Browser 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 `OVDDefaultSvcBrowser.xml` file.

See “Directly Editing the PortalView.xml File” on page 58 for more information

Relevant Files

- OVServiceDoc.dtd

This DTD is documented in the *HP OpenView VantagePoint Navigator Concepts and Configuration Guide* that is part of the OpenView Service Navigator documentation set. This DTD controls the XML formatting of data when it is sent from the OVO/OVSN management server to the SIP server.

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/VPNavigator/*.html

This directory contains the Help topics for OVSN modules. The Help topics are accessed by clicking the [?] button on the module’s title bar. If you want to supply your own customized help files, see “Writing Your Own Online Help” on page 108.

Table 4-1 OVSN Service Browser Module Files on the SIP Server

File Name	Windows Location %SIP_HOME%\....	UNIX Location /opt/OV/SIP/....
mgmtStations.dtd	conf\share\stations\	conf/share/stations/
mgmtStations.xml	conf\share\stations\	conf/share/stations/
ovoConfig.dtd	conf\share\stations\	conf/share/stations/
OVMModuleRegistration.dtd	registration\	registration/
OVRegSvcBrowser.xml	registration\	registration/
OVServiceDoc.dtd	conf\share\modules\service\	conf/share/modules/service/
OVSvcModules.dtd	conf\share\views\	conf/share/views/
OVDDefaultSvcBrowser.xml	registration\defaults\	registration/defaults/
PortalView.dtd	conf\share\views\	conf/share/views/
<i>PortalView.xml</i>	conf\share\views\	conf/share/views/
*.html	htdocs\C\help\VPNavigator\	htdocs/C/help/VPNavigator/

5 The Service Cards Module

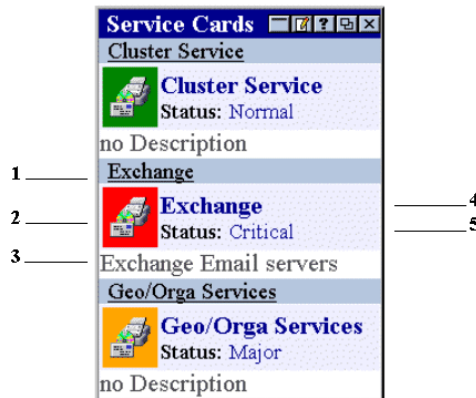
Using the Service Cards Module

Service Cards show at a glance the most important information about a service. You can drill down to details about each service or to any of the child services. To quickly locate and display a particular child service, use the search feature on the details window.

The number and selection of Service Cards depends upon the configuration settings of the Service Cards module (see “Editing the Service Cards Module” on page 67) and the configuration of filtering for the current SIP role (see Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 for information about filtering).

Figure 5-1 shows an example of the Service Cards module.

Figure 5-1 **The Service Cards Module**



1. Underlined label of the service.

Above the service icon, the underlined label of the service is displayed. If you click the icon, a Service Browser opens, displaying all children of the service. See “Using the Service Browser Module” on page 54.

2. Icon of the service.

The color of the icon indicates the current status of the service. See Table A-1 on page 148 for a description of the meaning of each color.

If you click the icon, the details of the Service Properties (such as actions, propagation rules, and calculation rules as defined within OVSN) are displayed.

Access to service details can be configured in the *PortalView.xml* file. By default, user access to service details is disabled.

3. Description of the service.

Below the icon, a description of the service is displayed. If no description is available, this is indicated as well.

4. Label of the service.

To the right of the icon, the label of the service is displayed.

5. Status of the service.

To the right of the icon, below the label of the service, the current status of the service is indicated.

Adding the Service Cards Module

To insert the Service Cards 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 `Service Cards` from the `Select Module to Add` list box, and click `[Add]`, or
- Click `[Edit]` to access the `Edit Column` window. Insert the `Service Cards` 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 Service Cards module is inserted into your *PortalView.xml* file.

- If you want to modify this module instance, turn to “Editing the Service Cards Module” on page 67.

If you have more than one OVO/OVSN management server configured to communicate with SIP, the first OVO/OVSN management server listed in the SIP Configuration Editor is accessed (“Establishing Communication Between OVO/OVSN and SIP” on page 20). You can change this selection through Service Cards - Edit window.

- If you want to change the default module, see “Relevant Files” on page 71.

TIP

If you want to add a module to the list of available modules, see “Relevant Files” on page 71. You can create and add another instance of any module.

Editing the Service Cards Module

The Service Cards module lets you add individual services in a business card format.

Using the Service Cards - Edit Page

You can easily modify the Service Cards 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 Service Cards module, click the edit button:



4. Make any desired changes. Click `[Help]` for specific instructions:

- Only one OVO management server communicates with any OVSN module. This module instance gathers data from the first OVO management server listed in the SIP Configuration Editor, *unless* you specify from which OVO management server to gather the data. Choose from the OVO management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20 if you need to add one).
- If you want to allow access to Service Properties (such as actions, propagation rules, and calculation rules as defined within OVSN), check `Allow details view drill-down`.
- Choose the services to be displayed as business card graphics within the module (either all top-level services or selected services).

If you want to select specific services, use shift-click or control-click to move groups of services from side to side. Only those services that passed the `<ServiceList>` of the

Management Data filter assigned to the current SIP Role are displayed (see “SIP Customer Model Definitions” on page 25 and “Role Configuration Settings” on page 26).

See “Filtering Possibilities for the Service Browser Module” on page 142 for more information.

- Remove a service from the Service Cards module’s displayed list.
 - Change the display order of selected services.
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.

Directly Editing the PortalView.xml File

TIP

For the following adjustments, you must edit the XML file. It is recommended that you use the *Service Cards - 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 Service Cards 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: %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.service.SvcCardServlet  
"
```

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.service.SvcCardServlet"  
  display="yes"  
  help="/OvSipDocs/C/help/VPNavigator/cardView.html"  
  id="module6"  
  rollupState="down"  
  title="Service Cards">  
</ModuleInstance>
```

4. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags, if the `<ServiceCard>` element does not already exist. Alternately, copy the contents from the Service Cards module’s default XML file (see “Relevant Files” on page 71):

```
<ServiceCard details="no" server="your_OVO_Server">  
  <ServiceRef name="email" />  
  <ServiceRef name="geo_orga" />  
  <ServiceRef name="cluster" />  
</ServiceCard>
```

Each `<ServiceRef>` element represents one card in the Service Cards module. See the comments in the `OVSvcModules.dtd` file for more information about the correct XML syntax (see “Relevant Files” on page 71).

5. To change the title of this Service Cards module instance, change the `title` attribute (`<ModuleInstance title="new title">`).

To change the title of all Service Cards modules, change the `title` attribute in the registration file, see “Relevant Files” on page 71.

6. To assign a different HTML file to the help [?] button for this Service Cards module instance, change the help attribute (`<ModuleInstance help=" " >`). See “Writing Your Own Online Help” on page 108.
7. Only one OVO management server communicates with any OVSN module. This module instance gathers data from the first OVO management server listed in the SIP Configuration Editor, *unless* you specify from which OVO management server to gather the data. Choose from the OVO management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20).

To specify an OVO management server, add the hostname attribute (`<Server hostname="your_OVO_server" />`) with the fully-qualified hostname of your OVO management server.

8. To display specific services, replace the values of the name attributes in the `<ServiceRef name=" " />` lines with your service names as defined in your Service Navigator database. For example:

```
<ServiceRef name="email" />
```

Each service that you list must pass the `<ServiceList>` of the Management Data filter assigned to the current SIP Role before that service’s data is displayed in the Service Cards module. To display *all* top-level services that pass the Management Data filter assigned to the current SIP Role, delete all `<ServiceRef name="xx" />` lines. See Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 for more information about Management Data filters.

9. Place the `<ServiceRef name="xx" />` lines in the order in which the cards should be displayed.
10. Save the XML file.
11. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
12. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Relevant Files

The Service Cards 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` & `ovoConfig.dtd` & `mgmtStations.xml`

This XML file configures communication between OVO/OVSN management servers and SIP servers. See the “Establishing Communication Between OVO/OVSN and SIP” on page 20.

- `OVModuleRegistraton.dtd` & `OVRegSvcCard.xml`

This XML file grants access to the Service Cards module through the SIP framework so that it is available for your use. To add another instance of the Service Cards module to the SIP module selection list, copy and rename the `OVRegSvcCard.xml` and the `OVDefaultSvcCard.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 150.

- `OVSvcModules.dtd` & `OVDefaultSvcCard.xml`

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

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

- Directly edit the XML code in the `OVDefaultSvcCard.xml` file, or
- Insert a Service Cards 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 `OVDefaultSvcCard.xml` file.

See “Directly Editing the PortalView.xml File” on page 68 for more information

- `OVServiceDoc.dtd`

This DTD is documented in the *HP OpenView VantagePoint Navigator Concepts and Configuration Guide* that is part of the OpenView Service Navigator documentation set. This DTD controls the XML formatting of data when it is sent from the OVO/OVSN management server to the SIP server.

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/VPNavigator/*.html

This directory contains the Help topics for OVSN modules. The Help topics are accessed by clicking the [?] button in the module’s title bar. If you want to supply your own customized help files, see “Writing Your Own Online Help” on page 108.

Table 5-1 OVSN Service Cards Module Files on the SIP Server

File Name	Windows Location %SIP_HOME%\....	UNIX Location /opt/OV/SIP/....
mgmtStations.dtd	conf\share\stations\	conf/share/stations/
mgmtStations.xml	conf\share\stations\	conf/share/stations/
ovoConfig.dtd	conf\share\stations\	conf/share/stations/
OVMModuleRegistration.dtd	registration\	registration/
OVRegSvcCard.xml	registration\	registration/
OVServiceDoc.dtd	conf\share\modules\service\	conf/share/modules/service/
OVSvcModules.dtd	conf\share\views\	conf/share/views/
OVDefaultSvcCard.xml	registration\defaults\	registration/defaults/
PortalView.dtd	conf\share\views\	conf/share/views/
PortalView.xml	conf\share\views\	conf/share/views/
*.html	htdocs\C\help\VPNavigator\	htdocs/C/help/VPNavigator/

6 **The Service Graph Module**

Using the Service Graph Module

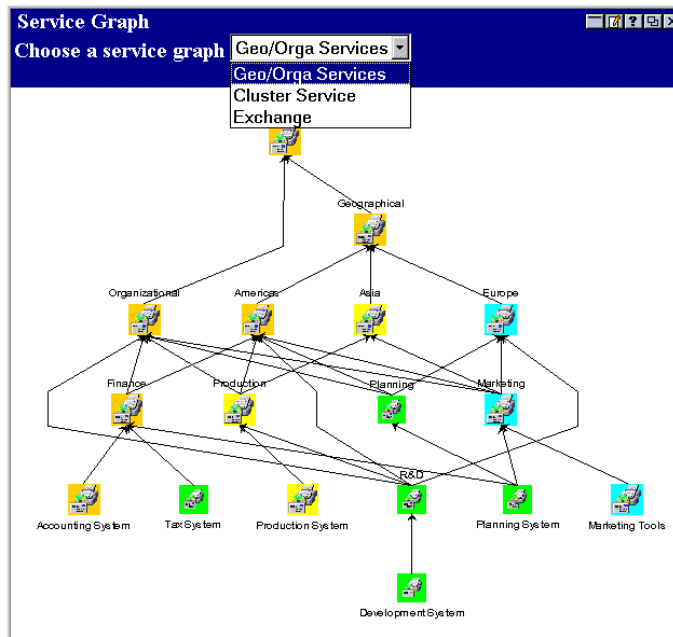
The Service Graph module displays the OVSN service hierarchy in a graphical view. Service Graph also provides drill-down capability. You can click on any service to get a summary view of child services. To select a Service Graph, select a graph from the Choose a service graph list box at the top of the Service Graph module.

The number and selection of available graphs depends upon the configuration settings of the Service Graph module (see “Editing the Service Graph Module” on page 77) and the configuration of filtering for the current SIP role (see Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 for information about filtering).

See Figure 6-1 for an example of the Service Graph module.

Figure 6-1

The Service Graph Module



NOTE

If you have over 200 nodes in the selected service's hierarchy, the graph is automatically limited to a depth of three child-service layers.

Adding the Service Graph Module

To insert the Service Graph 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 `Service Graph` from the `Select Module to Add` list box, and click `[Add]`, or
 - Click `[Edit]` to access the `Edit Column` window. Insert the `Service Graph` 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 Service Graph module is inserted into your `PortalView.xml` file.

- If you want to modify this module instance, turn to “Editing the Service Graph Module” on page 77.

If you have more than one OVO/OVSN management server configured to communicate with SIP, the first OVO/OVSN management server listed in the SIP Configuration Editor is accessed (“Establishing Communication Between OVO/OVSN and SIP” on page 20). You can change this selection through `Service Graph - Edit` window.

- If you want to change the default module, see “Relevant Files” on page 82.

TIP

If you want to add a module to the list of available modules, see “Relevant Files” on page 82. You can create and add another instance of any module.

Editing the Service Graph Module

The Service Graph module provides a graphical view of the service hierarchy.

Using the Service Graph - Edit Page

You can easily modify the Service Graph 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 Service Graph module, click the edit button:



4. Make any desired changes. Click `[Help]` for specific instructions:

- Only one OVO management server communicates with any OVSN module. This module instance gathers data from the first OVO management server listed in the SIP Configuration Editor, *unless* you specify from which OVO management server to gather the data. Choose from the OVO management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20 if you need to add one).
- Choose the services to be displayed within the Service Graph module (either all top-level services or selected services).

If you want to select specific services, use shift-click or control-click to move groups of services from side to side. Only those services that passed the `<ServiceList>` of the Management Data filter assigned to the current SIP Role are displayed (see “SIP Customer Model Definitions” on page 25 and “Role Configuration Settings” on page 26).

See “Filtering Possibilities for the Service Graphs Module” on page 144 for more information.

- Remove a service from the Service Graph module's displayed list.
 - Change the display order of selected services.
 - Set display depth for services.
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.

Directly Editing the PortalView.xml File

TIP

For the following adjustments, you must edit the XML file. It is recommended that you use the *Service Graph - 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 how many of child-service levels (`depth`) to display in the graph.
- Change the size of the graph using the `pixel width` and `height` attributes.

To directly modify the XML code for a Service Graph 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: %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.service.SvcGraphServlet"
```

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.service.SvcGraphServlet"  
  display="yes"  
  help="/OvSipDocs/C/help/VPNavigator/graphView.html"  
  id="module9"  
  rollupState="down"  
  title="Service Graph">  
</ModuleInstance>
```

4. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags, if the `<ServiceGraph>` element does not already exist. Alternately, copy the contents from the Service Graph module’s default XML file (see “Relevant Files” on page 82):

```
<ServiceGraph server="your_OVO_server">  
  <GraphRef  
    name="email"  
    depth="3"  
    width="600"  
    height="500" />  
</ServiceGraph>
```

Each `<GraphRef>` element represents one graph in the Service Graph module. See the comments in the `OVSvcModules.dtd` file for more information about the correct XML syntax:

- *Windows*: %SIP_HOME%\conf\share\views\OVSvcModules.dtd
 - *UNIX*: /opt/OV/SIP/conf/share/views/OVSvcModules.dtd
5. To change the title of this Service Graph module instance, change the title attribute (`<ModuleInstance title="new title">`).

To change the title of all Service Graph modules, change the `title` attribute in the registration file, see “Relevant Files” on page 82.

6. To assign a different HTML file to the help [?] button for this Service Graph module instance, change the `help` attribute (`<ModuleInstance help=" " >`). See “Writing Your Own Online Help” on page 108.
7. Only one OVO management server communicates with any OVSN module. This module instance gathers data from the first OVO management server listed in the SIP Configuration Editor, *unless* you specify from which OVO management server to gather the data. Choose from the OVO management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20).

To specify an OVO management server, add the `hostname` attribute (`<Server hostname="your_OVO_server" />`) with the fully-qualified hostname of your OVO management server.

8. To display specific services, replace the values of the `name` attributes in the `<GraphRef name=" " />` elements with your service names as defined in your Service Navigator database. For example:

```
<GraphRef name="email" />
```

Each service that you list must pass the `<ServiceList>` of the Management Data filter assigned to the current SIP Role before that service’s data is displayed in the Service Graph module. To display *all* top-level services that pass the Management Data filter assigned to the current SIP Role, delete all `<ServiceRef name="xx" />` lines. See Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 for more information about Management Data filters.

9. *Optional:* Specify how many of child-service levels (`depth`) to display in the graph, for example:

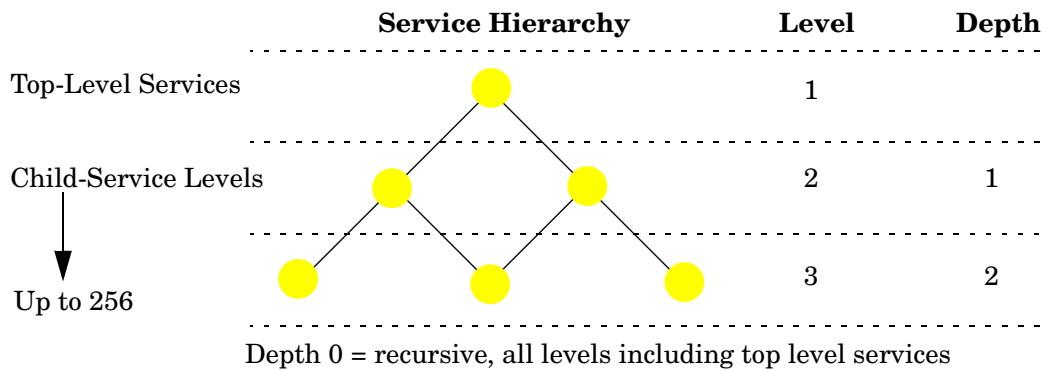
```
<GraphRef name="email" depth="3" />
```

The `depth` is the number of child-service levels within the OpenView Service Navigator hierarchy that are allowed to display in your SIP module. The `depth` value always indicates `depth` starting from level 2. For example:

- (*Default*) `depth="0"` stands for *recursive* and means that your SIP modules are allowed to display all service information.

- depth="1" means that your SIP modules are allowed to display service information from levels 1 and 2.
- depth="7" means your SIP modules are allowed to display service information from levels 1, 2, 3, 4, 5, 6, 7, and 8.

Figure 6-2 Deciding Upon the Child-Service Depth Setting



If you have over 200 nodes in the selected service’s hierarchy, the graph is automatically set at depth=3, three child-service layers.

10. *Optional:* Set the width and height (in pixels) for each graph.

```
<GraphRef
  name="email"
  depth="3"
  width="600"
  height="500" />
```

11. Place the <GraphRef name="xx" /> elements in the order in which the graphs should be displayed in the Choose a service graph list.
12. Save the XML file.
13. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
14. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Relevant Files

The Service Graph 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` & `ovoConfig.dtd` & `mgmtStations.xml`

This XML file configures communication between OVO/OVSN management servers and SIP servers. See the “Establishing Communication Between OVO/OVSN and SIP” on page 20.

- `OVModuleRegistraton.dtd` & `OVRegSvcGraph.xml`

This XML file grants access to the Service Graph module through the SIP framework so that it is available for your use. To add another instance of the Service Graph module to the SIP module selection list, copy and rename the `OVRegSvcGraph.xml` and the `OVDefaultSvcGraph.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 150.

- `OVSvcModules.dtd` & `OVDefaultSvcGraph.xml`

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

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

- Directly edit the XML code in the `OVDefaultSvcGraph.xml` file, or
- Insert a Service Graph 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 `OVDefaultSvcGraph.xml` file.

See “Directly Editing the PortalView.xml File” on page 78 for more information

- OVServiceDoc.dtd

This DTD is documented in the *HP OpenView Service Navigator Concepts and Configuration Guide* that is part of the OpenView Service Navigator documentation set. This DTD controls the XML formatting of data when it is sent from the OVO/OVSN management server to the SIP server.

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/VPNavigator/*.html

This directory contains the Help topics for OVSN modules. The Help topics are accessed by clicking the [?] button in the module’s title bar. If you want to supply your own customized help files, see “Writing Your Own Online Help” on page 108.

Table 6-1 OVSN Service Graph Module Files on the SIP Server

File Name	Windows Location %SIP_HOME%\....	UNIX Location /opt/OV/SIP/....
mgmtStations.dtd	conf\share\stations\	conf/share/stations/
mgmtStations.xml	conf\share\stations\	conf/share/stations/
ovoConfig.dtd	conf\share\stations\	conf/share/stations/
OVModuleRegistration.dtd	registration\	registration/
OVRegSvcGraph.xml	registration\	registration/
OVServiceDoc.dtd	conf\share\modules\service\	conf/share/modules/service/
OVSvcModules.dtd	conf\share\views\	conf/share/views/
OVDDefaultSvcGraph.xml	registration\defaults\	registration/defaults/
PortalView.dtd	conf\share\views\	conf/share/views/
<i>PortalView.xml</i>	conf\share\views\	conf/share/views/
*.html	htdocs\C\help\VPNavigator\	htdocs/C/help/VPNavigator/

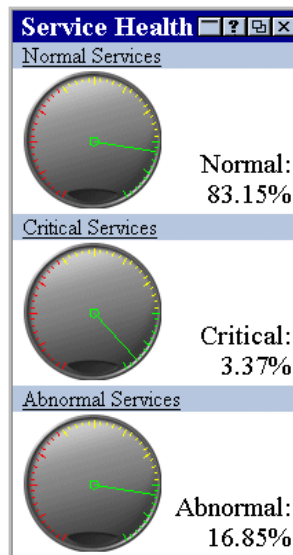
7 **The Service Health Module**

Using the Service Health Module

The Service Health module displays gauges measuring the percentage of services in a particular state (health status) or set of states. Service health is scored as a percentage from 0-100.

Figure 7-1 shows an example of the Service Health module. Depending upon how the Service Health module is configured, you may see one or more gauges, with each gauge presenting different data.

Figure 7-1 **The Service Health Gauges**



The Service Health module provides access to the following:

- The gauges indicate the percentage of all services being monitored that fall into the configured *status* category or categories. (See “Service Status” on page 148 for an explanation of the status categories.)

For example:

- The first gauge in Figure 7-1 is measuring the percentage of all services that currently report *normal* status (operating as expected). The high reading of 83.15% means that your service environment is healthy; a high reading here is good. The scale used by the gauge is:
 - 0-40% = red/bad
 - 41-80% = yellow/warning
 - 81-100% = green/good
- The second gauge in Figure 7-1 is measuring the percentage of all services that currently report *critical* status (immediate corrective action required). The low reading of 3.37% means that a small number of services are probably not available at this time; a low reading here is good. The scale used by the gauge is:
 - 100-60% = red/bad
 - 61-20% = yellow/warning
 - 19-0% = green/good
- The gauge title is a link. Click the title to display details about what is being measured. For example, if the gauge is monitoring all services currently reporting a *critical* status, a list of all *critical* services is displayed.
- The gauge icon is a link. Click the gauge icon to display details about each possible status and how many services are currently reporting each status. (See “Service Status” on page 148 for an explanation of the status categories.)

Adding the Service Health Module

To insert the Service 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 `Service Health` from the `Select Module to Add` list box, and click `[Add]`, or

- Click [Edit] to access the Edit Column window. Insert the Service 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 Service Health module is inserted into your *PortalView.xml* file.

- If you want to modify this module instance, turn to “Editing the Service Health Module” on page 89.

If you have more than one OVO/OVSN management server configured to communicate with SIP, the first OVO/OVSN management server listed in the SIP Configuration Editor is accessed (“Establishing Communication Between OVO/OVSN and SIP” on page 20). You can change this selection through Service Health - Edit window.

- If you want to change the default module, see “Relevant Files” on page 95.

TIP

If you want to add a module to the list of available modules, see “Relevant Files” on page 95. You can create and add another instance of any module.

Editing the Service Health Module

The Service Health module shows calculated service health in a gauge-like view.

Using the Service Health - Edit Page

You can easily modify the Service 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 Service Health module, click the edit button:



4. Make any desired changes. Click `[Help]` for specific instructions:
 - Only one OVO management server communicates with any OVSN module. This module instance gathers data from the first OVO management server listed in the SIP Configuration Editor, *unless* you specify from which OVO management server to gather the data. Choose from the OVO management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20 if you need to add one).
 - If you want to allow access to Service Properties (such as actions, propagation rules, and calculation rules as defined within OVSN), check `Allow details view drill-down`.
 - Create new health gauges to be displayed within the Service Health module. Click `[Add]` to advance to the Service Health Gauge Configuration page. Click the `[Help]` button for specific instructions.

- Modify an existing health gauge. Select the gauge title and click [Modify] to advance to the Service Health Gauge Configuration page. Click the [Help] button for specific instructions.
 - Remove a gauge from the Service Health module's list.
 - Change the display order of displayed gauges.
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.

Directly Editing the PortalView.xml File

TIP

For the following adjustments, you must edit the XML file. It is recommended that you use the Service 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.

To directly modify the XML code for a Service 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: %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.service.SvcHealthServlet"
```

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.service.SvcHealthServlet  
"  
    display="yes"  
    help="/OvSipDocs/C/help/VPNavigator/healthView.html "  
    id="module7"  
    rollupState="down"  
    title="Service Health">  
</ModuleInstance>
```

4. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags, if the `<ServiceHealth>` element does not already exist. Alternately, copy the contents from the Service Health module’s default XML file (see “Relevant Files” on page 95):

```
<ServiceHealth details="no" server="your_OVO_Server">  
    <ServiceSummary  
        id="uniqueString"  
        title="Your Gauge's Name"/>  
        critical="yes/no"  
        major="yes/no"  
        minor="yes/no"  
        warning="yes/no">  
        normal="yes/no"  
        inverse="yes/no"  
        recursive="yes/no"  
        scoreName="text to left of score"  
        <ServiceRef name="email"/>  
        <ServiceRef name="email_exch"/>  
        <ServiceRef name="geo_orga"/>  
    </ServiceSummary>  
</ServiceHealth>
```

Each `<ServiceSummary>` element represents one gauge in the Service Health module. See the comments in the `OVSvcModules.dtd` file for more information about the correct XML syntax (see “Relevant Files” on page 95).

5. To change the title of this Service Health module instance, change the `title` attribute (`<ModuleInstance title="new title">`).

To change the title of all Service Health modules, change the `title` attribute in the registration file, see “Relevant Files” on page 95.

6. To assign a different HTML file to the help [?] button for this Service Health module instance, change the `help` attribute (`<ModuleInstance help=" " >`). See “Writing Your Own Online Help” on page 108.
7. Only one OVO management server communicates with any OVSN module. This module instance gathers data from the first OVO management server listed in the SIP Configuration Editor, *unless* you specify from which OVO management server to gather the data. Choose from the OVO management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20).

To specify an OVO management server, add the `hostname` attribute (`<Server hostname="your_OVO_server" />`) with the fully-qualified hostname of your OVO management server.

8. If you want to include more Service Health Gauges, copy and paste the `<ServiceSummary>` block, one XML block for each gauge.

The order in which the `<ServiceSummary>` blocks are listed, determines the order in which the gauges are displayed in the Service Health module.

9. Modify each gauge’s `<ServiceSummary>` block:

- Replace the `id` value with a string that uniquely identifies this `ServiceSummary` within this particular Service Health module instance.
- Replace the value of the `title` attribute with the name you wish to use for your gauge.
- Indicate the status settings (as received from Service Navigator) that should contribute toward this gauge’s health calculation:
`critical="yes/no"`
`major="yes/no"`

```
minor="yes/no"  
warning="yes/no"  
normal="yes/no"
```

For example, if you select Critical and Major, a resulting score of 5% means that 5% of your services are either in a critical or major state.

- Indicate `inverse="yes"` or `"no"`. This setting controls which way the needle moves across the gauge. If a high score is a bad thing, set `inverse="yes"`.

If yes (high reading is a bad thing), the “inverted” scale is:

```
100-60% = red/bad  
61-20% = yellow/warning  
19-0% = green/good
```

If no (high reading is a good thing), the not inverted scale is:

```
0-40% = red/bad  
41-80% = yellow/warning  
81-100% = green/good
```

- Indicate `recursive="yes"` or `"no"`. A tree configuration of services and child services is configured in your HP OpenView Service Navigator database. The recursive attribute controls how many levels of the tree are included in this gauge's calculation.

If yes, all child services (all levels of the tree) under any specified service are included in the calculation.

If no, only the specified services and their immediate child services are included in the gauge's calculation (two layers deep).

NOTE: Each service must pass the `<ServiceList>` of the Management Data filter assigned to the current SIP Role. See “SIP Customer Model Definitions” on page 25 and Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 for information about Management Data filters.

- Replace the value of the `scoreName` attribute with a description of the score. This is the text that appears to the left of the percent (%) result at the bottom of the gauge. For example, if you type in "Normal", the results might be `Normal: 83.7%`
- Modify the list of `<ServiceRef>` elements to indicate which OVSN services should be included in this gauge's health calculation.

To indicate specific services, replace the values of the name attributes in the `<ServiceRef name=" " />` lines with your service names as defined in your Service Navigator database. For example:

```
<ServiceRef name="email" />
```

Each service that you list, within the `<ServiceSummary>` elements, must pass the `<ServiceList>` of the Management Data filter assigned to the current SIP Role before that service's data is included in the calculations for the gauge. To include *all* top-level services in the calculation, delete all `<ServiceRef name="xx" />` lines. See Chapter 9, "Segmenting the OVO/OVSN Data by Customer," on page 115 for more information about Management Data filters.

10. Save the XML file.
11. After you make modifications to XML files, validate the syntax. See "Validating XML Files" on page 155 for more information.
12. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Relevant Files

The Service 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` & `ovoConfig.dtd` & `mgmtStations.xml`

This XML file configures communication between OVO/OVSN management servers and SIP servers. See the “Establishing Communication Between OVO/OVSN and SIP” on page 20.

- `OVModuleRegistraton.dtd` & `OVRegSvcHealth.xml`

This XML file grants access to the Service Health module through the SIP framework so that it is available for your use. To add another instance of the Service Health module to the SIP module selection list, copy and rename the `OVRegSvcHealth.xml` and the `OVDefaultSvcHealth.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 150.

- `OVSvcModules.dtd` & `OVDefaultSvcHealth.xml`

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

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

- Directly edit the XML code in the `OVDefaultSvcHealth.xml` file, or
- Insert a Service 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 `OVDefaultSvcHealth.xml` file.

See “Directly Editing the PortalView.xml File” on page 90 for more information.

- OVServiceDoc.dtd

This DTD is documented in the *HP OpenView VantagePoint Navigator Concepts and Configuration Guide* that is part of the OpenView Service Navigator documentation set. This DTD controls the XML formatting of data when it is sent from the OVO/OVSN management server to the SIP server.

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/VPNavigator/*.html

This directory contains the Help topics for OVSN modules. The Help topics are accessed by clicking the [?] button in the module’s title bar. If you want to supply your own customized help files, see “Writing Your Own Online Help” on page 108.

Table 7-1 OVSN Service Health Module Files on the SIP Server

File Name	Windows Location %SIP_HOME%\....	UNIX Location /opt/OV/SIP/....
mgmtStations.dtd	conf\share\stations\	conf/share/stations/
mgmtStations.xml	conf\share\stations\	conf/share/stations/
ovoConfig.dtd	conf\share\stations\	conf/share/stations/
OVMModuleRegistration.dtd	registration\	registration/
OVRegSvcHealth.xml	registration\	registration/
OVServiceDoc.dtd	conf\share\modules\service\	conf/share/modules/service/
OVSvcModules.dtd	conf\share\views\	conf/share/views/
OVDDefaultSvcHealth.xml	registration\defaults\	registration/defaults/
PortalView.dtd	conf\share\views\	conf/share/views/
<i>PortalView.xml</i>	conf\share\views\	conf/share/views/
*.html	htdocs\C\help\VPNavigator\	htdocs/C/help/VPNavigator/

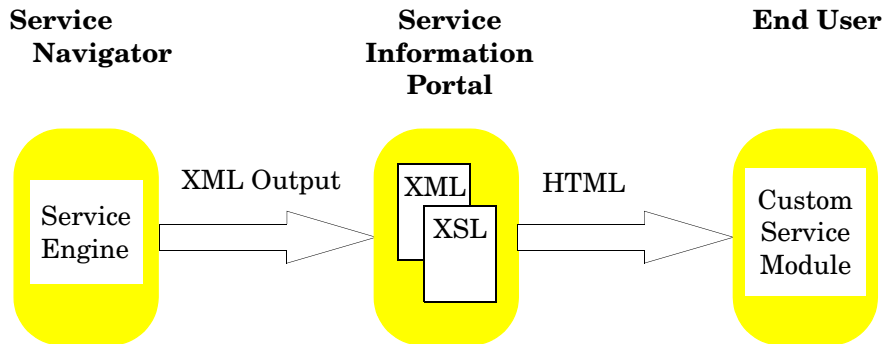
**8 Creating Your Own Service
 Modules**

Understanding the Possibilities

If none of the supplied OVSN modules meet your needs, you can develop your own format, layout, and style, without having to do any programming. For example, you could create a module that groups services by geography, groups services by function, or uses your own images.

SIP ships an example custom Service module that you can use as a starting point for creating your own custom Service module. This is just one example of how the custom Service module might work. (See the `README.html` file, page 111 for more information about the sample files. The `README.html` file contains a screen capture of the sample module as it appears in a SIP portal.)

Figure 8-1 **Displaying XML Data Through a Custom Service Module**



The remaining sections in this chapter guide you through the steps of creating your own custom Service module:

- “Creating Your Module’s Registration File” on page 100.
- “Developing Your Own XSL Stylesheet” on page 101.
- “Creating Your Module’s Default Instance” on page 104.
- “Inserting Your Custom Service Module into a Portal” on page 107.
- “Writing Your Own Online Help” on page 108.

The most difficult step is developing your own XSL stylesheet. Expertise in XSL is required.

TIP

Choose a name for your custom Service module's XSL stylesheet. This name *must* be used in the following two places because SIP automatically pairs stylesheets with custom Service modules based upon this name:

- The first part of the name of your XSL stylesheet, ***NAME***_html.xml
 - The value of the name attribute of the <ServiceCustom> element in your default custom Service module file.
-

Creating Your Module's Registration File

Register your new custom Service module with SIP to provide access to the new Service module through the SIP portal editing tools:

1. Copy and rename the following sample file. The new file can have any name that makes sense to you and *must* be placed in the same directory as the original file:
 - *Windows:* %SIP_HOME%/registration/OVRegSvcCustom.xml
 - *UNIX:* /opt/OV/SIP/registration/OVRegSvcCustom.xml
2. Change the description, title, classid, add, help, and defaultConfigXML attributes. The names you specify in help and defaultConfigXML are used in later steps:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE ModuleRegistration SYSTEM "OVModuleRegistration.dtd">

<ModuleRegistration
  vendorName="Hewlett-Packard Company"
  vendorURL="http://www.openview.hp.com"
  description="Enter a description of the module here"
  category="hp OpenView service navigator"
  title="Enter the Name of Your Custom Service Module Here"
  classid="uniqueString"
  implementation=
    "/servlet/com.hp.ov.portal.modules.service.SvcCustomServlet"
  outputType="any"
  add="yes" <!-- to add this module to the [Add] modules list -->
  edit="no" <!-- you cannot add an editing window (not supported) -->
  help="/OvSipDocs/C/help/VPNavigator/customView.html"
    <!--enter the name of the HTML help topic you wish to provide -->
  configDTD="OVSvcModules.dtd"
  defaultConfigXML="defaults/OVDefaultYourModuleName.xml"
    <!-- the name of your default module instance's XML filename -->
/>
```

3. Save the XML file.
4. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.

Developing Your Own XSL Stylesheet

You must develop an XSL (eXtensible Stylesheet Language) template that generates HTML from the XML code that SIP generates. Developing a new stylesheet requires expertise in using XSL.

The XML output produced by the Service Navigator service engine is based upon the `OVSvcCustomDoc.dtd` which is a system entity of `OVSvcCustomDoc.dtd`. You must develop an XSL stylesheet according to the rules of the `OVSvcCustomDoc.dtd` to format the data you wish to present in your custom Service module. The `OVSvcCustomDoc.dtd` defines the `<ServiceDocument>` element that is the root XML element from which formatting begins. Your new stylesheet must provide formatting rules for the `<Services>` child element. See “Relevant Files” on page 111 for more information about the governing DTD files.

Sample XSL Stylesheets

The stylesheets that are currently in use for the OVSN modules might be a good place to start looking for ideas. Three stylesheets exist for some of the OVSN modules: `*_html.xml` (for display in the SIP portal), `*_pda.xml` (for display on hand-held devices), `*_wml.xml` (for display on wireless devices, such as cell phone displays). Explore the following files. The `svccard_html.xml` is the simplest example.

All active SIP stylesheets *must* be stored in the `\conf\styles\` directory.

DO NOT MODIFY THESE FILES:

```
Windows: %SIP_HOME%\conf\styles\  
svcbrowser_html.xml  
svccard_html.xml  
svccard_pda.xml  
svccard_wml.xml  
svcdetail_html.xml  
svcgraph_html.xml  
svchealthdetail_html.xml  
svchealth_html.xml  
svchealth_pda.xml  
svchealth_wml.xml
```

```
UNIX: /opt/OV/SIP/conf/styles/  
svcbrowser_html.xsl  
svccard_html.xsl  
svccard_pda.xsl  
svccard_wml.xsl  
svcdetail_html.xsl  
svcgraph_html.xsl  
svchealthdetail_html.xsl  
svchealth_html.xsl  
svchealth_pda.xsl  
svchealth_wml.xsl
```

A sample custom XSL template is provided in the following directory. Examine this example, as well (see “Relevant Files” on page 111 for more information about the `exchmail_html.xsl`, `README.html`, and `exchmail.xml` sample files):

```
Windows: %SIP_HOME%\samples\service\exchmail_html.xsl  
UNIX: /opt/OV/SIP/samples/service/exchmail_html.xsl
```

Rules for Creating Your Own Stylesheet

Develop an XSL (eXtensible Stylesheet Language) template that generates HTML from the XML that SIP provides to the stylesheet.

1. The HTML output of your stylesheet must only contain HTML tags that are valid between the starting and closing tags of `<body>`. SIP automatically provides the framing `<html>`, `<head>`, and `<body>` tags.
2. You must save your new stylesheet file according to the following naming schema:

```
name_html.xsl
```

Where *name* is the same name as defined in the `name` attribute of the `ServiceCustom` element in the default custom Service module file (“Creating Your Module’s Default Instance” on page 104).

3. You must save your XSL style sheet into the following directory:

```
Windows: %SIP_HOME%\conf\styles\  
UNIX: /opt/OV/SIP/conf/styles/
```

To obtain the XML that SIP is providing to your XSL stylesheet, use the following process:

1. Log into SIP using a SIP role with `ViewAdmin` editing permissions.
2. Access the SIP Administrator role's portal and set the `Tracing Level for This Session` to `verbose`.
3. Switch to a SIP role where one of the OVSN modules is displayed.
4. Return to the SIP Administrator role's portal and click the `View Trace File: [View]` button to display the trace file.
5. Scroll through the trace file and look for a large entry starting with:

```
Processing XML: <ServiceDocument>
```

This entry represents the XML that needs to be formatted using your XSL stylesheet. Study this XML, refer to the DTDs and develop an XSL stylesheet that provides the format your desire.

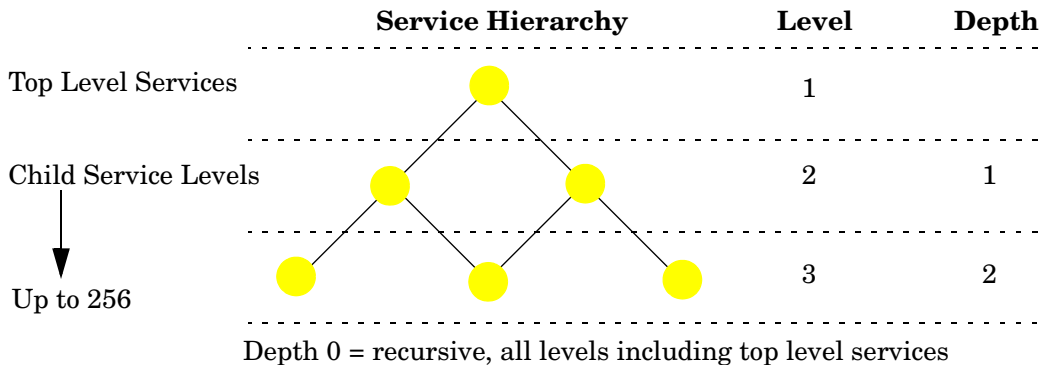
6. Turn off verbose tracing.

Creating Your Module's Default Instance

The `ServiceQuery` element defines the requests for information from the Service Navigator service engine. A custom Service module requires that you specify the `level` and `depth` attributes for your `ServiceQuery`:

- `level` = the starting point within your defined OpenView Service Navigator hierarchy from which services are visible through your custom Service module. For example, if `level="3"`, then service information starting at the third tier of your OpenView Service Navigator hierarchy is visible in your custom Service module.
- `depth` = the number of child-service levels within the OpenView Service Navigator hierarchy that are allowed to display in your custom Service module. The `depth` value always indicates depth starting from `level 2`. For example:
 - `depth="0"` stands for recursive and means that your custom Service module displays all service information below the specified hierarchy `level`.
 - `depth="1"` and `level="1"`, your custom Service module displays service information from levels 1 and 2. Level 1 is not visible.
 - `depth="7"` and `level="5"`, your custom Service module displays service information from levels 5, 6, 7, and 8. Levels 1, 2, 3, and 4 are not visible.

Figure 8-2 ServiceQuery Level and Depth



NOTE

The service information that appears in your custom Service module is further controlled by the service filter and depth settings you define in the SIP customer model data source. See also Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115.

To create a default version of your new module:

1. Copy and rename the sample `OVDDefaultSvcCustom.xml` file. The new file *must* have the same name that you specified in the registration file's `defaultConfigXML` attribute (see “Creating Your Module's Registration File” on page 100), and *must* be placed in the same directory as the original file:

Windows: %SIP_HOME%\conf\registration\defaults

UNIX: /opt/OV/SIP/conf/registration/defaults/

2. The sample `OVDDefaultSvcCustom.xml` file contains the following XML code:

```
<?xml version="1.0" encoding="UTF-8"?>
  <ServiceCustom name="" server="">
    <!-- <ServiceQuery depth="0" level="1" name="" /> -->
  </ServiceCustom>
```

3. Modify the code to create your new default custom Service module, or copy and modify the following into the XML code:

```
<ServiceCustom name="XSLfileName" server="your_OVO/OVSN_server">
  <ServiceQuery name="finance" depth="0" level="16" />
  <ServiceQuery name="email" depth="16" level="2" />
</ServiceCustom>
```

4. Replace the value of the `<ServiceCustom>` attributes with your own values:

name	The name you enter here matches the first part of the XSL stylesheet file name. See also “Developing Your Own XSL Stylesheet” on page 101.
server	<i>Optional:</i> Only one OVO/OVSN management server communicates with any OVSN module. This module instance gathers data from the first OVO/OVSN management server listed in the SIP Configuration Editor, <i>unless</i> you specify from which

OVO/OVSN management server to gather the data. Choose from the OVO/OVSN management servers currently configured for communication with SIP (see “Establishing Communication Between OVO/OVSN and SIP” on page 20), or delete this attribute.

5. Replace the attribute values of one or more `<ServiceQuery>` elements with your own values:

name	Enter a service name as defined in the OpenView Service Navigator database. Each service must pass the <code><ServiceList></code> of the Management Data filter assigned to the current SIP Role. See “SIP Customer Model Definitions” on page 25 and Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 for information about Management Data filters.
level	Specify the service hierarchy level (within the OpenView Service Navigator database) that is the starting point for gathering data for your custom Service module.
depth	Specify the child-service levels in the OpenView Service Navigator database hierarchy that are allowed to display in your custom Service module. (0-256, 0= <i>recursive</i>)

See also the `OVSvcModules.dtd` file for more information about the correct XML syntax.

6. Save your new default Custom module XML file.
7. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
8. See “Inserting Your Custom Service Module into a Portal” on page 107.

Your custom Service module is ready to be inserted into portal views when the module is selected from SIP's module selection drop-down list.

Inserting Your Custom Service Module into a Portal

To insert your custom Service module into a portal view:

1. Log into the *PortalView.xml* file to which you wish to add your custom Service module.

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

2. Navigate to the appropriate tab.
3. At the bottom of the column, either:
 - Select *Your Module Name* from the `Select Module to Add` list box, and click `[Add]`, or
 - Click `[Edit]` to access the `Edit Column` window. Insert your custom Service 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 instance of your custom Service module is inserted into your *PortalView.xml* file.

- If you want to modify this module instance, no `Edit` windows can be configured for custom Service modules. You must make all changes directly in the your *PortalView.xml* file.
- If you want to change your default custom Service module, see “Creating Your Module’s Default Instance” on page 104.

Writing Your Own Online Help

SIP ships a default HTML help file that is displayed when your customer clicks the [?] button in the title bar of your OVSN module. You can replace this default file with your own HTML help file:

1. Copy and rename any SIP help topic file. The new file can have any name that makes sense to you. Your new file can be located anywhere in the `htdocs/C/help/` directory. However, it is recommended that you store it in the `VPNavigator` directory with the other OVSN help topics:
 - *Windows:* `%SIP_HOME%\htdocs\C\help\VPNavigator\`
 - *UNIX:* `/opt/OV/SIP/htdocs/C/help/VPNavigator/`
2. The help file is in HTML format and should only use HTML syntax that is supported by *all web browsers*. Write your help file content. Follow the HTML format that you find in the other files in the `help` directory.

NOTE

Be careful to use only use ASCII or UTF-8 characters.

3. Save the new HTML help topic file.
4. Follow the directions for changing:
 - “The Default Help Topic for the Module” on page 108, and/or
 - “One Module Instance’s Help Topic” on page 109.

The Default Help Topic for the Module

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. Enter the new help file path and name into the `help` attribute of your module’s registration file. For more information about module registration files:

- For a custom Service module, see “Creating Your Module’s Registration File” on page 100.
 - For any other OVSN module, see the “Relevant Files” section of the chapter containing information about the module you wish to modify.
2. Save the file.
 3. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
 4. In a browser, log on as the appropriate SIP user.
 5. If you have not already done so, insert the module into the portal.
 6. Click [?] on the module’s title bar to ensure that the desired behavior is established.

One Module Instance’s Help Topic

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. In an ASCII or XML editor, open the *PortalView.xml* file that contains the module instance to which you wish to assign a new help file.

Portal view files are stored in the following directory:

Windows: %SIP_HOME%\conf\share\views

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

2. Search the following string until you locate the module with which you want to associate the new help file.
 - custom Service module: **uniqueString** (your new classid)
 - OVO Messages module: **modules.ovo**
 - Service Browser module: **modules.service.SvcListServlet**
 - Service Cards: **modules.service.SvcCardServlet**
 - Service Graph: **modules.service.SvcGraphServlet**
 - Service Health: **modules.service.SvcHealthServlet**

3. Insert the help attribute into the <ModuleInstance> element, for example:

```
<ModuleInstance
  classid="uniqueString"
  display="yes"
  help="/OvSipDocs/C/help/VPNavigator/helpTopic.html"
  id="module8"
  rollupState="down"
  title="Service Browser">
</ModuleInstance>
```

Add the help attribute as shown above and replace *helpTopic.html* with the name of your help file. Update the path, if necessary. The help attribute allows you to override the *default* help URL defined in the module's registration file. To change the default help topic, see "The Default Help Topic for the Module" on page 108.

4. Save the file.
5. After you make modifications to XML files, validate the syntax. See "Validating XML Files" on page 155 for more information.
6. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Relevant Files

The custom Service 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` & `ovoConfig.dtd` & `mgmtStations.xml`

This XML file configures communication between OVO/OVSN management servers and SIP servers. See the “Establishing Communication Between OVO/OVSN and SIP” on page 20.

- `exchmail_html.xsl` & `*.xsl`

This is a sample XML stylesheet for formatting the data for your custom Service module. This sample formats services into rows and columns of different colors. You can copy and rename this file or any of SIP's other XSL stylesheets as a starting point when creating your new custom Service module.

See “Developing Your Own XSL Stylesheet” on page 101 for more information.

- `README.html` & `exchmail.xml`

Use these files only if you want to load some dummy data into a test system running OpenView Service Navigator data base. This dummy data allows you to apply the sample XSL stylesheet provided with SIP (`exchmail_html.xsl`). Open the `README.html` to see a screen capture of an OVSN module that is using the `exchmail_html.xsl` stylesheet.

- `OVModuleRegistraton.dtd` & `OVRegSvcCustom.xml`

This sample XML file grants access to the custom Service module through the SIP framework so that it is available for your use. Copy and rename this file to add your new custom Service module to the SIP module selection list.

See “Creating Your Module’s Registration File” on page 100 for more information.

- `OVSvcModules.dtd` & `OVDdefaultSvcCustom.xml`

Relevant Files

This DTD defines the rules for configuring any of the OVSN modules. This XML file contains a sample *default* custom Service module. You copy and rename this file when creating your own custom Service module. The contents of the default file is inserted into your portal each time you use the [Add] button to insert your custom Service module.

See “Creating Your Module’s Default Instance” on page 104 for more information.

- OVServiceDoc.dtd & OVSvcCustomDoc.dtd

These DTDs are documented in the *HP OpenView VantagePoint Navigator Concepts and Configuration Guide* that is part of the OpenView Service Navigator documentation set. These DTDs control the XML formatting of data when it is sent from the OVO/OVSN management server to the SIP server. Note that the OVSvcCustomDoc.dtd contains the system entity OVServiceDoc.dtd. See the *HP OpenView VantagePoint Navigator Concepts and Configuration Guide* and read the information about service.dtd.

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/VPNavigator/customView.html

This directory contains the Help topics for OVSN modules. The Help topics are accessed by clicking the [?] button on the module’s title bar. If you want to supply your own customized help files, see “Writing Your Own Online Help” on page 108.

Table 8-1 Custom Service Module Files on the SIP Server

File Name	Windows Location %SIP_HOME%\....	UNIX Location /opt/OV/SIP/....
mgmtStations.dtd	conf\share\stations\	conf/share/stations/
mgmtStations.xml	conf\share\stations\	conf/share/stations/
ovoConfig.dtd	conf\share\stations\	conf/share/stations/

Table 8-1 Custom Service Module Files on the SIP Server (Continued)

File Name	Windows Location %SIP_HOME%\....	UNIX Location /opt/OV/SIP/....
README.html	samples\service\	samples/service/
exchmail.xml	samples\service\	samples/service/
exchmail_html.xsl	samples\service\	samples/service/
*.xsl	conf\styles\	conf/styles/
OVMModuleRegistration.dtd	registration\	registration/
OVRRegSvcCustom.xml	registration\	registration/
OVServiceDoc.dtd	conf\share\modules\service\	conf/share/modules/service/
OVSvcCustomDoc.dtd	conf\share\modules\service\	conf/share/modules/service/
OVSvcModules.dtd	conf\share\views\	conf/share/views/
OVDDefaultSvcCustom.xml	registration\defaults\	registration/defaults/
PortalView.dtd	conf\share\views\	conf/share/views/
<i>PortalView.xml</i>	conf\share\views\	conf/share/views/
customView.html	htdocs\C\help\VPNavigator\	htdocs/C/help/VPNavigator/

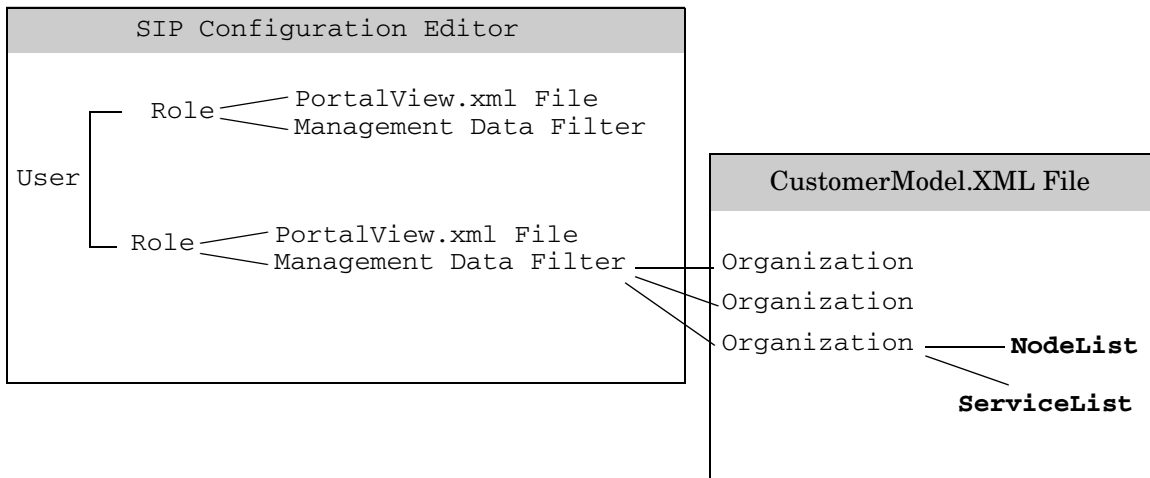
9**Segmenting the OVO/OVSN
Data by Customer**

Integrating OVO/OVSN Data into Your Customer Model

The SIP Customer Model allows you to associate resources (nodes and services) with *Organizations*. These *Organizations* are associated with SIP *Roles* so that data is automatically filtered appropriately when a user displays any of the OVO/OVSN modules. Figure 9-1 illustrates how the SIP Customer Model works. This process is required to enable the OVO/OVSN modules.

Before you proceed, decide for which Organizations 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 create node and service lists of the resources assigned to each of these organizations. Because of the assigned resource lists, each of these organizations could view the same instance of the OVO/OVSN modules, yet see only the data appropriate for them.

Figure 9-1 SIP Customer Model



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

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 10, “Display Filtering for OVO/OVSN,” on page 135. You can combine the SIP customer model and display filtering to get the desired results in the OVO Messages module and OVSN modules.

The OVO Messages module responds to *both* <NodeList> and <ServiceList> elements. The OVO Messages module displays any data that passes *either* the node or service list (the union of the two sets). OVSN modules respond only to <ServiceList> elements:

- **NodeList elements**

In the OVO message database, there is a node field and a source node field. The SIP <NodeList> elements key off of the node field.

- **ServiceList elements**

If the Service Navigator product (OVSN) is installed and configured on your OVO/OVSN management server, the SIP <ServiceList> elements key off of the currently defined services.

You must create lists of nodes and services in XML that conform to the specifications in these DTD files:

Windows:

```
%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 <ServiceList> elements can be defined in one or more XML files or a mix of CGIs, servlets, URLs, and files. These approaches are not mutually exclusive and can be used in combination:

- “Manually Creating NodeList and ServiceList Elements” on page 120
- “Customer-to-Node Mappings” on page 121
- “Customer-to-Service Mappings” on page 125

- “Registering a Customer Model Source” on page 132

SIP provides several tools that simplify the process of defining `<NodeList>` elements by gathering the required data from the NNM object database (a component of OVO). If these tools might be useful in your particular situation, see the *NNM Integration with SIP* manual (`NNM_Integration.pdf`) for more information about the following methods:

- “Dynamically Gathering Customer Views Organization Data”

This CGI program works only if the OpenView Customer Views program is 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”

This CGI program works only if the OpenView Customer Views program is 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.

- “Dynamically Gathering NNM Node and Interface Data”

Use the supplied servlet to dynamically generate lists of nodes 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>` elements are formatted according to the `SimpleCustomerModel.dtd`.

- “Exporting NNM Node and Interface Data to an XML File”

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>` elements formatted according to the `SimpleCustomerModel.dtd`.

You can also create your own program (CGI or servlet) to generate your `<NodeList>` and/or `<ServiceList>` elements 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.

Manually Creating NodeList and ServiceList 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 services specifically for use with the OVO Messages module and OVSN modules. You can create one XML file that contains all the various <NodeList> elements and <ServiceList> elements, or you can create multiple XML files containing the various list elements. Your lists 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="fully-qualified-hostname" />
      <Node name="IP-address" />
    </NodeList>

    <ServiceList name="Your Choice 2" >
      <Service name="as-defined-in-OVSN-database" >
        <Depth>0</Depth>
      </Service>
      <Service name="as-defined-in-OVSN-database">
        <Depth>256</Depth>
      </Service>
    </ServiceList>

  </Organization>

  <Organization name="Accounting">
    <NodeListRef href="Your Choice 1" />
    <ServiceListRef 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. These <NodeList> elements control which messages are displayed within the OVO Messages module.

NOTE

The SIP modules that communicate with OpenView Network Node Manager (NNM) also are affected by this same node list. See the *NNM Integration with SIP* manual (`NNM_Integration.pdf`) for more information:

- Alarms module
- Network Device Health module
- Topology module

The SIP report modules that communicate with OpenView Reporter (OVR) also are affected by this same node list. See the *OVR Integration with SIP* manual (`OVR_Integration.pdf`) for more information:

- OVR:NNM Reports module
- OVR:OVO Reports module
- OVR:OV Performance Reports module

If this causes a problem, you can assign multiple roles to a particular user. Optimize one role's node list for the OVO Messages module, another role's node list for the NNM modules, and another role's node list for the reports modules.

Determining Which Nodes to Add to the SIP Customer Model

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 the following to gather that information.

On the OVO for UNIX management server:

1. As root, run the `/opt/OV/bin/OpC/utils/opcnode` command to get a list of current node values in the OVO database:

```
$ opcnode -list_nodes
```

The following is an example of the output generated by this command for one database node value:

```
List of all Nodes in the ITO database:
=====
Name           = serverName.div.co.com
Label          = serverName
IP-Address     = 192.168.1.1
Network Type   = MACH_S400, NETWORK_IP, CONSOLE_TEMPLATE, COMM_NCS
Machine Type   = MACH_HP11_PA_RISC
Comm Type      = MACH_S800, COMM_DCE_UDP
=====
Operation successfully completed.
```

To list all nodes assigned to a specific group, run:

```
$ opcnode -list_ass_nodes group_name=xx
```

See the *opcnode.1m* manpage for more information.

2. You use this information when creating the <NodeList> elements on the SIP server.

On the OVO for Windows Management Server

To establish communication between OVO and SIP, you followed steps in Chapter 2 (“On Windows” on page 21) to place a SIP-supplied script called *ovow2sip.asp* onto the OVO/W management server. In the following steps, you will, optionally, edit the script to enable remote access to it. You will also use the script to generate a list of nodes in the XML format that complies with the *SimpleCustomerModel.dtd*.

1. *Optional:* To enable remote access to the *ovow2sip.asp* script, use an ASCII editor and edit the *ovow2sip.asp* script to disable the `LOCAL_ADMIN_ACCESS_ONLY` variable. It should look like this:

```
var LOCAL_ADMIN_ACCESS_ONLY = false;
```

2. From a web browser on the OVO/W management server, run the *ovow2sip.asp* script with the following parameter:

- `ovow2sip.asp?query=ListNodes`

3. You use this information when creating the <NodeList> elements on the SIP server.

Adding NodeLists to the SIP Customer Model

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

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

On the SIP server:

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:

```
%SIP_HOME%\conf\share\organizations\
```

UNIX:

```
/opt/OV/SIP/conf/share/organizations/
```

2. Save your new XML file in the `/conf/share/organizations` directory. If you place your XML file in any other location, update the path information when you get to step 8 and step 12.
3. Create a `<NodeList>` element for each group of nodes:

```
<NodeList>
</NodeList>
```

4. *Optional:* Give each node list a name that you can reference later when you are assigning node lists to specific `<Organization>` definitions:

```
<NodeList name="yourChoice" >
</NodeList>
```

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

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

6. *Optional*: add the `type` attribute to each node specification. This field is not used by the OVO Messages or OVSN modules. It is, however, displayed through the Managed Resources module:

```
<Node name="IP-address" type="xx" />
```

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

7. *Optional*: add the `DisplayString` element to each node specification. This field is not used by either the OVO Messages or OVSN modules. So disregard this element.
8. Ensure that the following lines are at the top of each XML file that you create:

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

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\">
```

9. Ensure that the following line is at the end of each XML file that you create:

```
</SimpleCustomerModel>
```

10. Save the XML file. For example, the following shows two node lists that contain all the desired nodes:

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

```
<SimpleCustomerModel>

  <NodeList name="Accounting">
    <Node name="host1.acme.com" />
    <Node name="server1.acme.com" />
    <Node name="router1.acme.com" />
    <Node name="host2.acme.com" />
    <Node name="server2.acme.com" />
    <Node name="router2.acme.com" />
  </NodeList>

  <NodeList name="Manufacturing">
    <Node name="host3.acme.com" />
    <Node name="server3.acme.com" />
    <Node name="router3.acme.com" />
    <Node name="host4.acme.com" />
    <Node name="server4.acme.com" />
    <Node name="router4.acme.com" />
  </NodeList>

</SimpleCustomerModel>
```

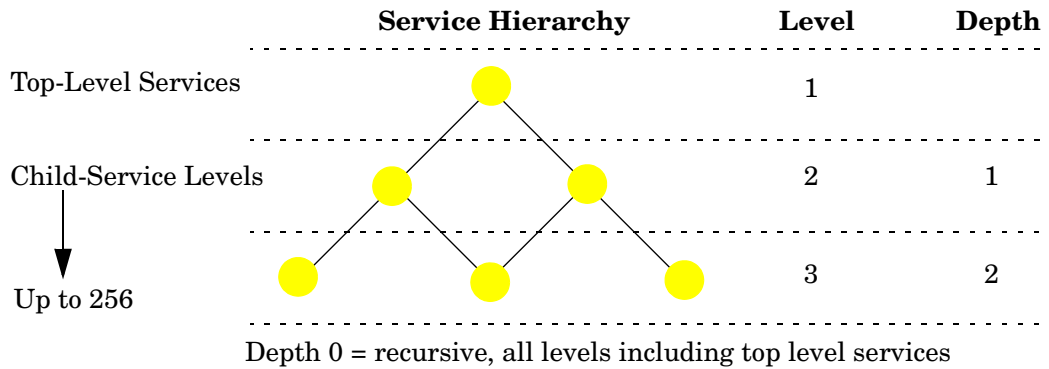
11. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
12. Register the new source of customer model data with SIP. For detailed instructions, see “Registering SIP Customer Model Sources” on page 132.
13. You are now ready to associate 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 120.

Customer-to-Service Mappings

If the OpenView Service Navigator program (OVSN) is installed and configured on your OVO/OVSN management server, you *must* create a <ServiceList> element within each SIP <Organization> in order to enable the OVSN modules.

It is recommended that you include only top-level services in your service lists. For each top-level service, you then specify the allowable depth (up to a maximum of 256 child-service levels).

Figure 9-2 Deciding Upon the Child-Service Depth Setting



The depth is the number of child-service levels within the OpenView Service Navigator hierarchy that are allowed to display in your SIP module. The depth value always indicates depth starting from level 2. For example:

- depth="0" stands for *recursive* and means that your SIP modules are allowed to display all service information.
- depth="1" means that your SIP modules are allowed to display service information from levels 1 and 2.
- depth="7" means your SIP modules are allowed to display service information from levels 1, 2, 3, 4, 5, 6, 7, and 8.

NOTE

The OVR:OVO Reports module that communicates with OpenView Service Navigator (OVSN) also respond to <ServiceList> elements. See the *OVR Integration with SIP* manual (OVR_Integration.pdf) for more information.

If sharing <ServiceList> elements causes a problem, you can assign multiple roles to a particular user. Optimize one role's <ServiceList> element for the OVSN modules and the second role's node <ServiceList> element for the OVR:OVO Reports module.

Determining Which Services to Add to the SIP Customer Model

If you do not already know the names of the services that you want to include in your <ServiceList> elements, do the following to gather that information:

On the OVO for UNIX management server:

As root, run the following commands to generate lists of currently configured services:

- To generate a list of top-level services, as root, run this OVSN command (in /opt/OV/bin/OpC/):

```
opcservice -list
```

The following is an example of the output generated by this command:

```
Service: clusterapps  
Service: email  
Service: geo_orga
```

- To generate a list of all services (top-level and child services), as root, run this OVSN command (in /opt/OV/bin/OpC/):
opcservice -list -recursive | grep Service: > filename

You use this information when creating the <ServiceList> elements on the SIP server.

On the OVO for Windows Management Server

To establish communication between OVO and SIP, you followed steps in Chapter 2 (“On Windows” on page 21) to place a SIP-supplied script called `ovow2sip.asp` onto the OVO/W management server. In the following steps, you will, optionally, edit the script to enable remote access to it. You will also use the script to generate a list of nodes in the XML format that complies with the `SimpleCustomerModel.dtd`.

1. *Optional:* To enable remote access to the `ovow2sip.asp` script, use an ASCII editor and edit the `ovow2sip.asp` script to disable the `LOCAL_ADMIN_ACCESS_ONLY` variable. It should look like this:

```
var LOCAL_ADMIN_ACCESS_ONLY = false;
```

2. From a web browser on the OVO/W management server, run the `ovow2sip.asp` script with one of the following parameters:

- `ovow2sip.asp?query=ListServices`

- `ovow2sip.asp?query=ListServices&includeDisplay=true`
3. You use this information when creating the `<ServiceList>` elements on the SIP server.

Adding Services to the SIP Customer Model

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

```
<ServiceList name="your choice" >
    <Service name="as-defined-in-OVSN-database" />
    <Service name="as-defined-in-OVSN-database" />
</ServiceList>
```

On the SIP server:

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.
2. 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:

`%SIP_HOME%\conf\share\organizations\`

UNIX:

`/opt/OV/SIP/conf/share/organizations/`

3. Save your new XML file in the `/conf/share/organizations` directory. If you place your XML file in any other location, update the path information when you get to step 10 and step 14.
4. Create a `<ServiceList>` element for each group of services:

```
<ServiceList>
</ServiceList>
```

5. *Optional:* Give each service list a name that you can reference later when you are assigning service lists to specific `<Organization>` definitions:

```
<ServiceList name="yourChoice" >
</ServiceList>
```


6. *Required*: enter the name value for each Service to be included in this list. It is recommended that you include only top-level services. This is the service name as defined within the OpenView Service Navigator database. For example:

```
<ServiceList name="yourChoice" >
  <Service name="from OVSN database" />
  <Service name="cluster" />
  <Service name="email_exch" />
  <Service name="geo_orga" />
</ServiceList>
```

7. *Optional*: set the depth attribute for each service specification. Depth controls how many child-service levels are displayed in your SIP modules. If no Depth is specified, the default is zero (0).

```
<Service name="serviceName" >
  <Depth>12</Depth>
</Service>
```

Valid entries:

0 = *recursive*, means that all levels of child services in the service tree pass through.

1 - 256 = a specific number of child-service levels in the service tree pass through.

8. *Optional*: add the type attribute to each service specification. This field is not used by the OVO Messages or OVSN modules. It is, however, displayed through the Managed Resources module:

```
<Service name=" " type="server" />
```

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

9. *Optional*: add the DisplayString element to each service specification. The DisplayString displays in the OVSN modules instead of the label of the service as defined within the OpenView Service Navigator database:

```
<Service name="serviceName" >
  <Depth>0-256</Depth>
  <DisplayString>optional</DisplayString>
</Service>
```

10. Ensure that the following lines are at the top of each XML file that you create:

Segmenting the OVO/OVSN Data by Customer

Manually Creating NodeList and ServiceList Elements

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

<SimpleCustomerModel>
```

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">
```

11. Ensure that the following line is at the end of each XML file that you create:

```
</SimpleCustomerModel>
```

12. Save the XML file. For example, the following shows two service lists that contain all the desired top-level services:

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

    <ServiceList name="Marketing" >
        <Service name="email" />
        <Service name="webServers" />
        <Service name="databaseServers" />
        <Service name="eCommerce" />
    </ServiceList>

    <ServiceList name="Accounting" >
        <Service name="email" />
        <Service name="webServers" />
        <Service name="accountingSystems" />
    </ServiceList>
</SimpleCustomerModel>
```

```
<Service name="auditingSystems" />  
</ServiceList>  
</SimpleCustomerModel>
```

13. After you make modifications to XML files, validate the syntax. See “Validating XML Files” on page 155 for more information.
14. Register the new source of customer model data with SIP. For detailed instructions, see “Registering SIP Customer Model Sources” on page 132.
15. You are now ready to associate these <ServiceList> 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 120.

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

```
http://othermachine/CustomerModel.xml  
http://SIPhostname/ovportal/NNMSimpleCustomerModel
```

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].

10 **Display Filtering for OVO/OVSN**

Introduction to Display Filtering

NOTE

Display filtering is *optional*.

Whereas the Management Data filter determines what is *possible* to see in the OVO Messages module and OVSN modules (Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115), display filters control what is actually visible in particular module instances. Display filters provide a method of restricting the data one last time before the module is visible. Display filters are input in a variety of ways:

- For the **OVO Messages module**, display filtering can be applied at a message category level. Because these display filters are entered at the message category level, they affect all OVO Messages module instances that display that particular message category. You can create multiple sets of message category files to meet different display filtering needs. To create a display filter for a message category, you must directly edit the message category’s XML file. See page 137.
- For the **OVSN modules provided with SIP**, display filtering is applied through the module’s `Edit` window. Each display filter affects only one module instance. See:
 - Service Browser module, page 142.
 - Service Cards module, page 143.
 - Service Graph module, page 144.
 - Service Health module, page 145.
- For a **custom Service module (created by you)**, display filtering is configured through the `ServiceQuery` elements defined within the particular module instance. Because the custom Service modules do not have an `Edit` window, you must directly edit the module’s XML file to create the filter. See page 104.

Filtering Possibilities for the OVO Messages Module

Filters for Specific Message Category Definitions

A display filter is *required* for each individual message category. This display filter actually determines which of the five standard message attributes are used to sort the message category. An empty display filter results in everything that passes the Management Data filter, passing the display filter. The messages displayed in the OVO Messages module result from the *intersection* of the Management Data filter and the display filter. It is possible at runtime for the *intersection* of these lists to be the empty set. In this case, no messages are output for the message category. (See Chapter 9, “Segmenting the OVO/OVSN Data by Customer,” on page 115 for information about the Management Data filter.)

The filtering possibilities for message categories include the following. Only one of these filters can be used per OVO message category definition file:

NOTE

When writing filters, use Perl5 regular expressions. For example:
`.*\.eagle\.wingnuts\.com`

See www.omniscium.com/projects/RegExpTool/ for an interactive tool that helps you write Perl5 expressions.

See www.perl.com or www.perldoc.com for information about Perl5 regular expressions.

-
- `ApplicationSelection`
Passes only messages from specified applications (as defined within OVO). See the `OVOMessageCategory.dtd` file for more information.

For example, the following `ApplicationSelection` allows all applications to pass:

```
<ApplicationSelection />
```

The following example allows application 1, 2, and 3 to pass (application# values can be Perl5 regular expressions):

```
<ApplicationSelection
application="(application1|application2|application3)" />
```

- **NodeSelection**
Passes messages from devices specified by hostname or capability, or can refer to an organization defined within the SIP customer model. See the `filter.dtd` file for more information.

For example, the following `NodeSelection` filter allows all nodes to pass:

```
<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` database, with values such as `isRouter` or `isNode`. However, this filter can search upon any field in the `OVO` database), and
- Any node that is included in `OrganizationOne` or `OrganizationTwo` definition.

```
<NodeSelection>
  <IPHostFilter>
    <IPHost hostname="15\.2\.5\.125" />
    <IPHost hostname="15\.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.

- MessageGroupSelection
Passes only messages from specified OVO message groups (as defined within OVO). See the OVOMessageCategory.dtd file for more information.

For example, the following MessageGroupSelection allows all groups to pass:

```
<MessageGroupSelection />
```

The following example allows groups 1, 2, and 3 to pass (messageGroup# values can be Perl5 regular expressions):

```
<MessageGroupSelection  
messageGroup=" (messageGroup1|messageGroup2|messageGroup3) "  
>
```

- ObjectSelection
Passes only messages from specified OVO objects. See the OVOMessageCategory.dtd file for more information.

For example, the following ObjectSelection allows all objects to pass:

```
<ObjectSelection />
```

The following example allows objects 1, 2, and 3 to pass (object# values can be Perl5 regular expressions):

```
<ObjectSelection object="(object1|object2|object3)" />
```

The following example allows objects that end with "mail" to pass:

```
<ObjectSelection object="(. *mail)" />
```

- ServiceSelection
Passes only messages from the OVSN services specified by name and/or type (as defined within OVO). See the OVOMessageCategory.dtd file for more information.

For example, the following `ServiceSelection` allows all services to pass:

```
<ServiceSelection />
```

The following example allows services 1, 2, and 3 to pass (`service#` values can be Perl5 regular expressions):

```
<ServiceSelection service="(service1|service2|service3)" />
```

The following example allows services that begin with "exch_" to pass:

```
<ServiceSelection service="(exch_*)" />
```

Empty display filters are provided with the default OVO message category definition files, thus allowing *all* messages to pass. To modify these default display filters, you must edit the source XML files (“Editing OVO Message Categories” on page 45). The default OVO message category files are:

- **Windows:** %SIP_HOME%\conf\share\modules\applications.xml
 %SIP_HOME%\conf\share\modules\messageGroups.xml
 %SIP_HOME%\conf\share\modules\nodes.xml
 %SIP_HOME%\conf\share\modules\objects.xml
 %SIP_HOME%\conf\share\modules\services.xml
- **UNIX:**
 /opt/OV/SIP/conf/share/modules/applications.xml
 /opt/OV/SIP/conf/share/modules/messagesGroups.xml
 /opt/OV/SIP/conf/share/modules/nodes.xml
 /opt/OV/SIP/conf/share/modules/objects.xml
 /opt/OV/SIP/conf/share/modules/services.xml

Filtering at the Module Instance Level

At the module instance level, you can designate:

- Server (hostname)
Optional: Only one OVO management server communicates with any OVO Messages module:
 - If you have more than one OVO management server configured in the SIP Configuration Editor, use this attribute to designate from which server this module instance should gather messages. The fully-qualified hostname of one of the OVO management servers as entered into the SIP Configuration Editor.

— If this attribute is not present, the module instance gathers messages from the first OVO management server listed in the SIP Configuration Editor.

See “Establishing Communication Between OVO/OVSN and SIP” on page 20 for more information about the SIP Configuration Editor.

- `MessageCategoryRef href="messageCategory"`
The href values must match the id attribute values in your *OVOMessageCategory.xml* files. Messages for this OVO Messages module instance are gathered from the specified message categories.

Display Filtering the OVSN Modules

Filtering Possibilities for the Service Browser Module

You can easily modify the list of services displayed in one instance of the Service Browser module by using the module's `Edit` window. SIP automatically writes the display filter for you:

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 Service Browser module, click the edit button:



4. Select `Choose from list`. Click `[Help]` if you need more information.

The list includes:

- All services (including all levels of child services) that are currently configured in OVSN on the specified OVO/OVSN management server.
- This list is filtered according to the `Management Data` filter assigned to the current SIP role. Only those services and child service that passed the filter are displayed.

5. Select any service in the `All Services` list and move it to the `Selected Services` list by clicking `[Add]`.

Use shift-click or control-click to move groups of services from side to side.

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

The filter in your *PortalView.xml* file looks similar to this example:

```
<ServiceBrowser details="yes" server="myOVOSrv.div.co.com">
  <ServiceRef name="apps" />
  <ServiceRef name="cluster" />
  <ServiceRef name="email" />
</ServiceBrowser>
```

Filtering Possibilities for the Service Cards Module

You can easily modify the list of services displayed in one instance of the Service Cards module by using the module's Edit window. SIP automatically writes the display filter for you:

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 Service Cards module, click the edit button:



4. Select Choose from list. Click [Help] if you need more information.

The list includes:

- All services (including all levels of child services) that are currently configured in OVSN on the specified OVO/OVSN management server.
- This list is filtered according to the Management Data filter assigned to the current SIP role. Only those services and child service that passed the filter are displayed.

5. Select any service in the All Services list and move it to the Selected Services list by clicking [Add].

Use shift-click or control-click to move groups of services from side to side.

6. To save the changes and return to the main portal page, click [OK].
7. Log off of the SIP portal.

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

The filter in your *PortalView.xml* file looks similar to this example:

```
<ServiceCard details="no" server="myOVOSrv.div.co.com">
  <ServiceRef name="apps"/>
  <ServiceRef name="cluster"/>
  <ServiceRef name="email"/>
</ServiceCard>
```

Filtering Possibilities for the Service Graphs Module

You can easily modify the list of services in the Choose a Service Graph list box within the Service Graphs module by using the module's Edit window. SIP automatically writes the display filter for you:

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 Service Graphs module, click the edit button:



4. Select Choose from list. The list includes:
 - All services (including all levels of child services) that are currently configured in OVSN on the specified OVO/OVSN management server.
 - This list is filtered according to the Management Data filter assigned to the current SIP role. Only those services and child service that passed the filter are displayed.

5. Select any service in the All Services list and move it to the Selected Services list by clicking [Add].

Use shift-click or control-click to move groups of services from side to side. You can also specify a depth for each selected service.

6. Click [Help] if you need more information.
7. To save the changes and return to the main portal page, click [OK].

8. Log off of the SIP portal.
9. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

The filter in your *PortalView.xml* file looks similar to this example:

```
<ServiceGraph server="myOVOSrv.div.co.com">  
    <GraphRef name="planning" />  
    <GraphRef name="finance" />  
</ServiceGraph>
```

Filtering Possibilities for the Service Health Module

You can easily narrow the list of services that are being monitored for the health calculation in a Service Health module by using the module's Edit window. SIP automatically writes the display filter for you:

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 Service Health module, click the edit button:



4. In the Current Service Health Gauges list, select the gauge you wish to configure and click [Modify].
5. In Gauge - Service Selection, select Choose from list. The list includes:
 - All services (including all levels of child services) that are currently configured in OVSN on the specified OVO/OVSN management server.
 - This list is filtered according to the Management Data filter assigned to the current SIP role. Only those services and child service that passed the filter are displayed.
6. Select any service in the Available Services list and move it to the Selected Services list by clicking [Add Service].

Use shift-click or control-click to move groups of services from side to side. Click [Help] if you need more information.

7. To save the changes and return to the main portal page, click [OK] twice.
8. Log off of the SIP portal.
9. Log into the SIP portal as the appropriate user to ensure that you have the desired results.

Each gauge has its own separate filter. The filter in your *PortalView.xml* file looks similar to this example:

```
<ServiceHealth
  details="yes"
  server="myOVOSrv.div.co.com">
  <ServiceSummary
    id="myGauge"
    inverse="no"
    critical="no"
    major="no"
    minor="no"
    warning="no">
    normal="yes"
    recursive="yes"
    scoreName="Normal"
    title="Normal Services"
    <ServiceRef name="email"/>
    <ServiceRef name="apps"/>
    <ServiceRef name="pkgs"/>
    <ServiceRef name="cluster"/>
  </ServiceSummary>
</ServiceHealth>
```

A **Service Status**

Service Status

Table A-1 describes the meaning of each severity color.

Table A-1 **Service Status**

Status	Color	Meaning
Critical	Red	A service-affecting condition has occurred and immediate corrective action is required.
Major	Orange	The severity of the problem is relatively high and normal use of the object is likely to be impeded.
Minor	Yellow	A problem of relatively low severity has occurred, which should not impede normal use of the object.
Warning	Cyan	A potential or impending, service-affecting fault has occurred. Action should be taken to diagnose and correct the problem.
Normal	Green	Message output is expected: for example, a process is starting or completing, or status information is displayed.

B **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” in 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:

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** or use the batch command **%SIP_HOME/bin/restart_tomcat.bat**.

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

Restarting Tomcat

Restarting the Servlet Engine

C 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. By default, the apache web server and SIP run as:

Solaris: user "nobody"

HP-UX: user "www"

The default user names for Solaris and HP-UX can be changed during installation.

At runtime, `umask` is set by tomcat 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: %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.

A

- adding
 - create your own service module, 107
 - OVO Messages module, 36
 - Service Browser module, 55, 107
 - Service Cards module, 65
 - Service Graph module, 75
 - service health gauges, 89
 - Service Health module, 87
- ApplicationSelection, 137

C

- CapabilityFilter, 138
- communication
 - SIP server, 23
- communication path
 - configuring, 20
 - OpenView Operations to SIP, 14
 - OpenView Service Navigator to SIP, 15
- configuring
 - for non-English language, 29
 - OVO Messages module, 38
 - service browser, 57, 58
 - service cards, 67
 - service graph, 77
 - service health gauges, 89
- creating
 - message categories, 47
 - XML node lists, 123
 - XML service lists, 128
 - your own service module, 107
- custom service modules
 - developing stylesheet, 101
 - example stylesheets, 101
 - required ServiceList, 116
 - understanding, 98
 - writing online help, 108
- customer model
 - CGI, 118
 - creating, 116
 - current settings, 25
 - registering, 132
 - requirements, 25
 - writing filters, 116

D

- default module
 - OVDDefaultOVO.xml, 50
 - OVDDefaultSrvBrowser.xml, 61, 112

- OVDDefaultSrvCards.xml, 71
- OVDDefaultSrvGraph.xml, 82
- OVDDefaultSrvHealth.xml, 95
- deleting message categories, 48
- developing a stylesheet
 - custom service modules, 101
- display filtering
 - introduction, 136
- DTDs
 - filters.dtd, 140
 - OVO messages, 50
 - OVServiceDoc.dtd, 112
 - service browser, 61, 111
 - service cards, 71
 - service graph, 82
 - service health, 95

E

- editing
 - XML files, 154
- editing message categories, 45
- editing XML files, 154
- examples
 - custom service module stylesheets, 101

F

- files
 - custom OVSN module, 112
 - OVO messages, 50
 - service browser, 61, 111
 - service cards, 71
 - service graph, 82
 - service health, 95
- filtering
 - NodeList, 25, 117, 120, 121
 - ServiceList, 117, 120, 125
- filtering at module instance
 - GraphRef, 145
 - introduction to display filters, 136
 - ServiceRef, 143, 144, 146
- filtering message categories
 - ApplicationSelection, 137
 - introduction to display filters, 136
 - MessageGroupSelection, 139
 - NodeSelection, 138
 - NodeSelection(CapabilityFilter), 138
 - NodeSelection(IPHostFilter), 138
 - NodeSelection(OrganizationFilter), 138
 - ObjectSelection, 139

Index

ServiceSelection, 139

G

global settings

OVO Messages module, 42

I

installation, 17

configuring communication channels, 20

See SIP Installation Guide

(SIP_Install_Guide.pdf)

IPHostFilter, 138

M

message categories

creating, 47

editing, 45

MessageGroupSelection, 139

messages

understanding, 36

N

NNM Reports module

customer model requirements, 25

NodeSelection, 138

non-English language mode, 29

configuring, 29

O

ObjectSelection, 139

online help

per module instance, 108

OpenView Operations

and Service Information Portal, 12

in SSL environment, 31

OpenView Service Information Portal

in SSL environment, 31

OpenView Service Navigator

and Service Information Portal, 13

OrganizationFilter, 138

OV Performance Reports module

customer model configuration, 25

OVO Messages module, 12

adding, 36

configuring, 38

creating categories, 47

default module, 50

deleting categories, 48

DTDs, 50

editing categories, 45

files, 50

global settings, 42

required NodeList, 116

understanding, 36

XML for categories, 46

XML for module instances, 40

OVO Reports module

customer model configuration, 25

OVOConfig.xml, 51

OVServiceDoc.dtd, 112

OVSN custom module

files, 112

P

permissions

XML files, 154

R

registering

customer model source, 132

restarting

Tomcat, 150

role configuration

SIP, 26

S

server configuration

OVSN management server, 22

SIP, 23

service browser

adding, 55, 58, 107

configuring, 57, 58

default module, 61, 112

DTDs, 61, 111

files, 61, 111

required ServiceList, 116

sorting services, 54

understanding, 54

XML, 59

service cards

adding the module, 65

configuring, 67

default module, 71

DTDs, 71

files, 71

required ServiceList, 116

understanding, 64

- XML, 69
- service graph
 - adding, 75
 - configuring, 77
 - default module, 82
 - DTDs, 82
 - files, 82
 - required ServiceList, 116
 - understanding, 74
 - XML, 79
- service health
 - adding a module, 87
 - adding gauges, 89
 - configuring gauges, 89
 - default module, 95
 - DTDs, 95
 - file, 95
 - required ServiceList, 116
 - understanding gauges, 86
 - XML, 91
- Service Information Portal
 - and OpenView Operations, 12
 - and Service Navigator, 13
- Service Navigator
 - and Service Information Portal, 13
 - modules, 13
 - starting processes, 22
- service status, 148
- ServiceSelection, 139
- severity, 148
- SIP
 - communication, 23
 - installation directory, 23
 - non-English language mode, 29
 - OS requirements, 20
 - OVSN server configuration, 22
 - role configuration, 26
 - server configuration, 23
- sorting
 - services in service browser, 54
- SSL, 31
- starting
 - Service Navigator processes, 22
- status, 148
- stylesheets
 - for custom service modules, 101

T

- Tomcat, restarting, 150

U

- understanding
 - custom service modules, 98
 - messages, 36
 - service browser, 54
 - service cards, 64
 - service graph, 74
 - service health gauges, 86
- UNIX
 - restarting Tomcat, 150

W

- Windows
 - restarting Tomcat, 150
- writing online help
 - per module instance, 108

X

- XML
 - backing up, 154
 - editing files, 154
 - node lists, 123
 - OVO Messages module, 40, 46
 - rules for editing, 154
 - Service Browser module, 59
 - Service Cards module, 69
 - Service Graph module, 79
 - Service Health module, 91
 - service lists, 128
 - validating, 155
- XML files
 - permissions, 154
- XSL
 - See stylesheets.