

HP OpenView Service Information Portal 2.0

Presenting VantagePoint Navigator Data

Windows NT®, Windows® 2000, HP-UX, and Solaris



Manufacturing Part Number: J4797-90005

April 2001

© Copyright 2001 Hewlett-Packard Company.

Legal Notices

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

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

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

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

HEWLETT-PACKARD COMPANY
United States of America

Copyright Notices. ©Copyright 1983-2001 Hewlett-Packard Company, all rights reserved.

Reproduction, adaptation, or translation of this material without prior written permission is prohibited, except as allowed under the copyright laws.

Trademark Notices.

Adobe® is a trademark of Adobe Systems Incorporated.

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

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

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

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

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

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

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

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

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

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

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

UNIX® is a registered trademark of the Open Group.

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

Windows® 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.

Conventions**Contact Information****1. Overview of Presenting VantagePoint Navigator Data Through the Portal**

How VantagePoint Navigator Works with Service Information Portal	18
The VantagePoint Navigator Modules	19
Understanding the Configuration Files	20
OVSvcModules.dtd	21
Rules for Direct Editing of XML Files	28
Backing Up XML Files	28
Validating XML Files	28
Avoiding Loss of Changes	29

2. Presenting Services in the Service Browser

Understanding the Service Browser	32
Viewing Services	32
Searching Services	33
Sorting Services	33
Configuring the Service Browser	35
Adding a Service Browser to a Portal View	35
Through the User Interface	35
Through the Portal View File	36
Removing a Service Browser from a Portal View	38
Through the User Interface	38
Through the Portal View File	38

3. Presenting Service Cards

Understanding Service Cards	42
Configuring Service Cards	44
Adding Service Cards to a Portal View	44
Through the User Interface	44
Through the Portal View File	45
Changing the Display Order of Service Cards	47
Through the User Interface	47
Through the Portal View File	47

Contents

Removing Service Cards from the Service Cards Module	49
Through the User Interface	49
Through the Portal View File	50
4. Presenting Services in the Service Graph	
Understanding the Service Graph	54
Configuring the Service Graph	55
Adding a Service Graph to a Portal View	55
Through the User Interface	55
Through the Portal View File	56
Removing a Service Graph from a Portal View	59
Through the User Interface	59
Through the Portal View File	59
5. Presenting Service Health Data	
Understanding Service Health Gauges	62
Gauge View	63
Details View	63
Configuring Service Health Gauges	64
Calculating Service Health	64
Adding Service Health Gauges to a Portal View	65
Through the Portal View File	65
Changing the Display Order of Service Health Gauges	66
Through the Portal View File	66
Removing Service Health Gauges from a Portal View	67
Through the Portal View File	67
6. Presenting Service Details	
Understanding Service Details	70
Controlling Access to Service Details	72
Enabling/Disabling Access to Service Details	72
Through the User Interface	72
Through the Portal View File	72

7. Presenting Service Reports

Understanding Service Reports	76
Configuring the Service Reports	77
Setting up HP OpenView VantagePoint Reporter	77
Adding Service Reports to a Portal View	77

8. Presenting Custom Service Views

Understanding Custom Service Views	80
Understanding Depth and Level in ServiceQuery	81
Custom Service View Examples	82
Configuring a Custom Service View	83
Adding Custom Service Views to a Portal View	83
Developing Your Own Stylesheet	86
Writing Your Own Online Help	87
To Associate the [?] Button With Your Own Help File	87
Removing a Custom Service View from a Portal View	89
Through the User Interface	89
Through the Portal View File	89
OVSvcCustomDoc.dtd	91

A. Service Status

Service Status	96
----------------------	----

Contents

Figure 2-1. The Service Browser Module	32
Figure 3-1. The Service Cards Module	42
Figure 4-1. The Service Graph Module	54
Figure 5-1. The Service Health Gauges	62
Figure 6-1. Details of a Service	70
Figure 7-1. A Bookmarks Module Used for Service Reports	76
Figure 8-1. Displaying XML Data Through Customized Service Views	80
Figure 8-2. Deciding the Value for Depth	81

Table 1-1. OVSvcModules.dtd Elements and Attributes	24
Table 8-1. OVSvcCustomDoc.dtd Elements and Attributes.	93
Table A-1. Service Status.	96

Tables

Conventions

The following typographical conventions are used in this manual.

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

Contact Information

Technical Support and Training

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

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

Documentation Feedback

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

You can provide feedback about documentation:

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

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

1 Overview of Presenting VantagePoint Navigator Data Through the Portal

How VantagePoint Navigator Works with Service Information Portal

HP OpenView VantagePoint Navigator, an add-on product of HP OpenView VantagePoint Operations, enables you to manage an IT environment while focusing on the IT services you provide. The integration of VantagePoint Navigator into Service Information Portal provides to your customers up-to-date information about their managed services.

For more information about HP OpenView VantagePoint Navigator, see the *HP OpenView VantagePoint Navigator Concepts and Configuration Guide*.

The VantagePoint Navigator Modules

You can choose from and combine in any way the following VantagePoint Navigator modules to display your customers' services:

- **Service Browser**

Table-like view of services, including drill-down to details (including actions, propagation and calculation rules).

- **Service Graph**

Graphical view of the service environment.

- **Service Card**

Business-card like view of a service.

- **Service Health**

Shows aggregated service status in gauge-like view.

- **Service Reports**

Reports generated by HP OpenView VantagePoint Reporter on service status.

- **Custom Service Views**

Your own presentation of services in a VantagePoint Navigator module.

Understanding the Configuration Files

Before you can configure the VantagePoint Navigator modules, you must have VantagePoint Navigator servers configured as described in the *Configuring VantagePoint Navigator* guide (*Configuring_VPNavigator.pdf*).

The VantagePoint Navigator modules are configured in portal view files. The following chapters describe how to modify portal view configuration files to display VantagePoint Navigator data through Service Information Portal.

Configuration files use eXtensible Markup Language (XML). Each type of configuration file has an associated Document Type Definition (DTD) that describes the proper syntax for that file.

VantagePoint Navigator comes with the following DTDs:

- `servers.dtd`
Used by XML files to define VantagePoint Navigator servers in Service Information Portal. See the manual *Configuring VantagePoint Navigator* (*Configuring_VPNavigator.pdf*) for more information.
- `OVSvcModules.dtd`
Included in `PortalView.dtd`. Used by portal view files to configure VantagePoint Navigator modules. See “OVSvcModules.dtd” on page 21 for more information.
- `OVSvcCustomDoc.dtd`
Used by the Custom Service Views module only. See Chapter 8, “Presenting Custom Service Views,” on page 79 for details.
- `OVSvcServiceDoc.dtd`
Used by the Custom Service Views module only. This DTD is documented in the *HP OpenView VantagePoint Navigator Concepts and Configuration Guide*. See also Chapter 8, “Presenting Custom Service Views,” on page 79.

OVSvcModules.dtd

This section contains a detailed description of the XML syntax of the `OVSvcModules.dtd`. Note that the `OVSvcModules.dtd` is a system entity of the `PortalView.dtd`.

The `OVSvcModules.dtd` is located in the following directory:

Windows NT/2000:

```
<install_dir>\SIP\conf\share\modules\service
```

UNIX: /etc/opt/OV/share/SIP/conf/share/modules/service

The DTD shows the proper grammar for the service module configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- OVSvcModules.dtd -->
<!-- Copyright (c) 2001 Hewlett-Packard Company -->
<!ELEMENT ServiceModules ((ServiceCard | ServiceGraph |
ServiceBrowser | ServiceCustom | ServiceHealth)*)>
<!ELEMENT ServiceCard (ServiceRef*)>
<!ATTLIST ServiceCard
id CDATA #IMPLIED
server CDATA #REQUIRED
details (1 | 0 | YES | yes | NO | no) "no"
>
<!ELEMENT ServiceGraph (GraphRef*)>
<!ATTLIST ServiceGraph
id CDATA #IMPLIED
server CDATA #REQUIRED
>
<!ELEMENT ServiceBrowser (ServiceRef*)>
```

Overview of Presenting VantagePoint Navigator Data Through the Portal **OVSvcModules.dtd**

```
<!ATTLIST ServiceBrowser
id CDATA #IMPLIED
server CDATA #REQUIRED
details (1 | 0 | YES | yes | NO | no) "no"
>
<!ELEMENT ServiceHealth (ServiceSummary*)>
<!ATTLIST ServiceHealth
id CDATA #IMPLIED
server CDATA #REQUIRED
details (1 | 0 | YES | yes | NO | no) "no"
>
<!ELEMENT ServiceCustom (ServiceQuery*)>
<!ATTLIST ServiceCustom
id CDATA #IMPLIED
name CDATA #REQUIRED
server CDATA #REQUIRED
>
<!ELEMENT ServiceSummary (ServiceRef*)>
<!ATTLIST ServiceSummary
id CDATA #REQUIRED
title CDATA #REQUIRED
scoreName CDATA #REQUIRED
normal (1 | 0 | YES | yes | NO | no) "yes"
warning (1 | 0 | YES | yes | NO | no) "yes"
minor (1 | 0 | YES | yes | NO | no) "yes"
major (1 | 0 | YES | yes | NO | no) "yes"
critical (1 | 0 | YES | yes | NO | no) "yes"
recursive (1 | 0 | YES | yes | NO | no) "no"
```

```
inverse (1 | 0 | YES | yes | NO | no) "no"  
>  
<!ELEMENT ServiceQuery EMPTY>  
<!ATTLIST ServiceQuery  
name CDATA #REQUIRED  
depth CDATA #IMPLIED  
level CDATA #IMPLIED  
>  
<!ELEMENT ServiceRef EMPTY>  
<!ATTLIST ServiceRef  
name CDATA #REQUIRED  
>  
<!ELEMENT GraphRef EMPTY>  
<!ATTLIST GraphRef  
name CDATA #REQUIRED  
depth CDATA #IMPLIED  
width CDATA #IMPLIED  
height CDATA #IMPLIED  
>
```

Table 1-1 on page 24 lists the elements and attributes of the OVSvcModules.dtd.

Table 1-1 OVSvcModules.dtd Elements and Attributes

Elements	Attributes	Description
ServiceCard	id	Optional. Unique identifier of the Service Card module. <string>
	server	Required. Name of server as defined in the file servers.xml. <string>
	details	Optional. Specifies whether or not a user is allowed to access service details. <"yes" "no"> Default is "no".
ServiceGraph	id	Optional. Unique identifier of the Service Graph module. <string>
	server	Required. Name of server as defined in the file servers.xml. <string>
ServiceBrowser	id	Optional. Unique identifier of the Service Browser module. <string>
	server	Required. Name of server as defined in the file servers.xml. <string>
	details	Optional. Specifies whether or not a user is allowed to access service details. <"yes" "no"> Default is "no".
ServiceHealth	id	Optional. Unique identifier of the Service Health module. <string>
	server	Required. Name of server as defined in the file servers.xml. <string>
	details	Optional. Specifies whether or not a user is allowed to access service details. <"yes" "no"> Default is "no".

Table 1-1 OVSvcModules.dtd Elements and Attributes

Elements	Attributes	Description
ServiceCustom	id	Optional. Unique identifier for a Custom Service View. <string>
	name	Required. Name of the Custom Service View module. This name is referenced by the XSL stylesheet to format the XML data in HTML. <string>
	server	Required. Name of server as defined in the file servers.xml. <string>
ServiceSummary	id	Required. Unique identifier for a Service Summary. <string>
	title	Required. Name of the Service Health Gauge displayed just above the gauge. <string>
	scoreName	Required. Name for calculated health in percent. Displayed to the right of the gauge. <string>
	normal	Required. Severity normal. <"yes" "no"> Default is "yes".
	warning	Required. Severity warning. <"yes" "no"> Default is "yes".
	minor	Required. Severity minor. <"yes" "no"> Default is "yes".
	major	Required. Severity major. <"yes" "no"> Default is "yes".
	critical	Required. Severity critical. <"yes" "no"> Default is "yes".

Table 1-1 OVSvcModules.dtd Elements and Attributes

Elements	Attributes	Description
ServiceSummary	recursive	Optional. If set to yes, all services in the customer model data source will be used for the purpose of status calculation. If set to no, only the toplevel services listed in the customer model data source will be used. <"yes" "no"> Default is "yes".
	inverse	Optional. Inverts the calculated percentage. For example, if the calculated value is 25%, this attribute inverts it to 75%. <"yes" "no"> Default is "no".
ServiceQuery	name	Required. Name of the service as defined in VantagePoint Navigator. <string>
	depth	Optional. Number of levels in the service hierarchy to be displayed to a customer. <positive integer>
	level	Optional. Specific level in the service hierarchy to be displayed to a customer. <positive integer>
ServiceRef	name	Required. Name of the service as defined in VantagePoint Navigator. <string>

Table 1-1 OVSvcModules.dtd Elements and Attributes

Elements	Attributes	Description
GraphRef	name	Required. Name of the service as defined in VantagePoint Navigator. <string>
	depth	Optional. Number of levels of the service hierarchy that will be displayed by the Service Graph. By default, the toplevel services defined in the customer model data source will be displayed if no value for depth is defined. <positive integer>
	width	Optional. Width of the Service Graph image in pixels. Default is 600 pixels.
	height	Optional. Height of the Service Graph image in pixels. Default is 500 pixels.

Rules for Direct Editing of XML Files

- Make a backup before modifying 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.

Validating XML Files

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 NT/2000: %SIP_HOME\bin
UNIX: /opt/OV/SIP/bin

The correct usage of the `xmlvalidate` command is:

```
xmlvalidate <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’ has been misspelled.

As an alternative to `xmlvalidate`, you can find an XML validation tool for Windows NT/2000 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.

Overview of Presenting VantagePoint Navigator Data Through the Portal
OVSvcModules.dtd

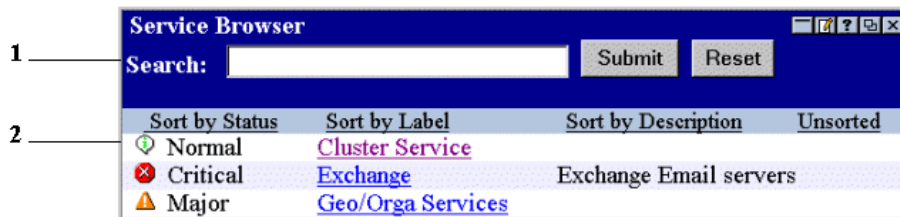
2**Presenting Services in the
Service Browser**

Understanding the Service Browser

The Service Browser shows a list of the toplevel services specified in the customer model data source. From here, you can access details about each service or drill down to any of the lower-level services. To quickly locate and display a service, use the search feature. The order of services can be easily changed by applying one of the four sort functions. The number and selection of services in the Service Browser depends on the configuration of the customer model data source.

Figure 2-1 on page 32 shows an example of the Service Browser module.

Figure 2-1 **The Service Browser Module**



1. Section dedicated to searching and sorting services.
2. Section including a list of services.

See the following topics for more information:

- Viewing Services
- Searching Services
- Sorting Services

Viewing Services

The second section of the Service Browser has the following main columns:

- **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 96 describes the meaning of each color and gives a detailed explanation of each status.

Clicking on the status icon in this column displays the details of that service. Access to service details can be configured in the *PortalView.xml* file. By default, user access to service details is disabled. See “Understanding Service Details” on page 70.

- **Sort by Label column**

The second column from the left displays the label of the service. Clicking on the underlined label shows a drill-down view of the Service Browser, displaying the children services of this service

- **Sort by Description column**

The third column from the left displays a description of the service.

“Sorting Services” on page 33 explains how services can be sorted in the Service Browser.

Searching Services

Service Search helps locating a service in Service Information Portal. Service Search is easy to use: simply type in a word(s) and click on the [Submit] button.

The search tool will search all service labels. If a label doesn’t exist for a service, the service name will be searched. Search terms in lowercase will match words in any case. Search terms can be complete words, syllables, or single letters.

The result of the Service Search is a Service Browser like window with a list of all services whose label or name matches the search term.

Sorting Services

By default, the services appear unsorted. If required, services can be sorted according to the following criteria:

- **Sort by Status**

Click on the [Sort by Status] button to display services in the following order:

Understanding the Service Browser

- critical services at the top
 - major services
 - minor services
 - warning services
 - normal services at the bottom
 - **Sort by Label**
Click on the [Sort by Label] button to sort services alphabetically by their label.
 - **Sort by Description**
Click on the [Sort by Description] button to sort services alphabetically by their description.
- To sort the files in the original order, click the [Unsorted] button.

Configuring the Service Browser

The Service Browser module shows a list of services in a table-like view. In addition, it lets you search the service hierarchy for a particular service.

The first step in configuring the Service Browser module is to ensure that VantagePoint Navigator has been configured to work with Service Information Portal. See the *Configuring VantagePoint Navigator* (*Configuring_VPNavigator.pdf*) guide for more information. After that, you can add a Service Browser to be displayed in a portal view.

- Adding a Service Browser to a Portal View
- Removing a Service Browser from a Portal View

Adding a Service Browser to a Portal View

Two ways exist for adding a Service Browser to be displayed in a portal view.

- Through the User Interface
- Through the Portal View File

However, it's much easier through the user interface than opening and manually editing the XML file.

Through the User Interface

1. Login as the customer whose portal you want to change.
2. Scroll to the bottom of the page. You'll notice drop-down list boxes with [Add] buttons next to them.
3. Select `Service Browser` from the list and click [Add].

A Service Browser is added to the portal page. The new module automatically displays the services of the VantagePoint Navigator server listed first in the `servers.xml` file. Further configuration is necessary.

4. On the title bar of the Service Browser module, click [Edit].
5. On the `Service Browser - Edit` page, select a VantagePoint Navigator management server from the drop-down list box.

Configuring the Service Browser

6. If you want to allow access to Service Details, check the `Allow details view drill-down` check box.
7. Choose which services you wish to as toplevel services in the Service Browser module:
 - **Toplevel only**

Select `All toplevel` from customer model, if you want to display the toplevel services specified in the customer model data source.
 - **Selected services**

Select `Choose` from list and select a service from the `All services` list. Click `[Add]`.

Repeat this step until the `Selected services` list contains all the services you want to list in the Service Browser.
8. To save and return to the main portal page, click `[OK]`.

Through the Portal View File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

1. If a portal view file already exists, open it with an ASCII or XML editor and proceed to Step 2.

If a portal view file does not yet exist, see the *SIP 2.0 Administrator Guide* (`Administrator_Guide.pdf`) and follow the procedure for creating portal views.

2. Create a new module instance and copy the following attribute into the `ModuleInstance` element:

```
classid="com.hp.ov.portal.modules.service.SvcListServlet"
```

The `ModuleInstance id` must be unique among all module instances in the portal view file. For information on the `ModuleInstance` element, see the *SIP 2.0 Administrator Guide*

(Administrator_Guide.pdf).

3. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags.

```
<ServiceBrowser details="no" server="bbn_server1">
  <ServiceRef name="email"/>
  <ServiceRef name="email_exch"/>
  <ServiceRef name="geo_orga"/>
</ServiceBrowser>
```

4. Replace the value of the `server` attribute with your own value. This is the logical name of the VantagePoint Navigator server as defined in `servers.xml`. See the *Configuring VantagePoint Navigator* for more information (Configuring_VPNavigator.pdf).

Replace the values of the `name` attribute with your own values. These are the service names as defined in VantagePoint Navigator. Note that the services listed here must also be configured in the customer model data source.

See also “OVSvcModules.dtd” on page 21 for more information about the correct XML syntax.

5. Save the XML file. In a browser, log in as the appropriate customer and ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP. The correct usage of this command is:
`xmlvalidate <xml_filename>.xml.`

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.

Make sure the following has been added to your `PATH` variable:

Windows NT/2000: %SIP_HOME%\bin
UNIX: /opt/OV/SIP/bin

Presenting Services in the Service Browser

Configuring the Service Browser

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

Removing a Service Browser from a Portal View

The following ways exist for removing Service Browsers from a portal view:

- Through the User Interface
- Through the Portal View File

However, it's much easier through the user interface than opening and manually editing the XML file.

Through the User Interface

Modules can be removed through the user interface, either through the `Options` window or by clicking the `Close [X]` button in the module's title bar.

When using the `Options` window, the module is removed entirely from the `PortalView.xml` file. When using the `Close [X]` button, the `display` attribute in the `PortalView.xml` file is set to "no". If you want to redisplay this module, open the `PortalView.xml` file and change the `display` attribute to "yes".

Through the Portal View File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

1. In an ASCII or XML editor, open the `PortalView.xml` file.
2. Search the string `<ModuleInstance`

`classid="com.hp.ov.portal.modules.service.SvcListServlet"` until you locate the Service Browser that you wish to remove.

3. Do one of the following:

- Change the `display` attribute in the `<ModuleInstance>` to `display="no"`. The `<ModuleInstance>` section is still present in the XML file.
 - Delete everything from `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcListServlet">` through `</ModuleInstance>` to remove the selected Service Browser from the portal view file.
4. Save the XML file. In a browser, log in as the appropriate customer to ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP. The correct usage of this command is:

```
xmlvalidate <xml_filename>.xml.
```

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.

Make sure the following has been added to your PATH variable:

Windows NT/2000: %SIP_HOME%\bin

UNIX: /opt/OV/SIP/bin

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

Presenting Services in the Service Browser
Configuring the Service Browser

Understanding Service Cards

Service Cards show at a glance the most important information about a service. The number and selection of Service Cards depends on the Service Card module configuration. Figure 3-1 on page 42 shows an example of the Service Cards module.

Figure 3-1 **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 on it, a Service Browser opens, displaying all children of the service. See “Understanding the Service Browser” on page 32.

2. Icon of the service.

The color of the icon indicates the current status of the service. See Table A-1 on page 96 for a description of the meaning of each color.

If you click on the icon, the details of the service are displayed. See “Understanding Service Details” on page 70.

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.

Configuring Service Cards

The Service Cards module lets you add individual services in a business card like format to the portal.

The first step in configuring the Service Cards module is to ensure that VantagePoint Navigator has been configured to work with Service Information Portal. See the *Configuring VantagePoint Navigator* guide for more information ([Configuring_VPNavigator.pdf](#)). After that, you can add Service Cards to be displayed in a portal view.

- Adding Service Cards to a Portal View
- Changing the Display Order of Service Cards
- Removing Service Cards from the Service Cards Module

Adding Service Cards to a Portal View

Two ways exist for adding Service Cards to be displayed in a portal view.

- Through the User Interface
- Through the Portal View File

However, it's much easier through the user interface than opening and manually editing the XML file.

Through the User Interface

1. Login as the customer whose portal you want to change.
2. Scroll to the bottom of the page. You'll notice drop-down list boxes with [Add] buttons next to them.
3. Select `Service Cards` from the list and click [Add].

The Service Cards module is added to the portal page. The new module automatically displays the services of the VantagePoint Navigator server listed first in the `server.xml` file. Further configuration is necessary.

4. On the title bar of the Service Cards module, click [Edit].
5. On the `Service Cards - Edit` page, select a VantagePoint Navigator management server from the drop-down list box.

- Depending on which server you select, the number and selection of services in the `All services` list box below will change.
- If you want to allow access to Service Details, check the `Allow details view drill-down` check box.
 - Choose which services you wish to display as Service Cards in the Service Cards module:
 - Toplevel only**
Select `All toplevel` from `customer model`, if you want to display the toplevel services specified in the customer model data source.
 - Selected services**
Select `Choose from list` and select a service from the `All services` list. Click `[Add]`.

Repeat this step until the `Selected services` list contains all the services you want to display as Service Cards.
 - To save and return to the main portal page, click `[OK]`.

Through the Portal View File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

- If a portal view file already exists, open it with an ASCII or XML editor and proceed to Step 2.

If a portal view file does not yet exist, see the *SIP 2.0 Administrator Guide* (`Administrator_Guide.pdf`) and follow the procedure for creating portal views.

- Create a new module instance and copy the following attribute into the `ModuleInstance` element:

```
classid="com.hp.ov.portal.modules.service.SvcCardServlet"
```

The `ModuleInstance` `id` must be unique among all module

Presenting Service Cards

Configuring Service Cards

instances in the portal view file. For information on the `ModuleInstance` element, see the *SIP 2.0 Administrator Guide* (`Administrator_Guide.pdf`).

3. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags.

```
<ServiceCard details="no" server="bbn_server1">
  <ServiceRef name="email" />
  <ServiceRef name="geo_orga" />
  <ServiceRef name="cluster" />
</ServiceCard>
```

4. If you want to include more Service Cards, copy and paste the `ServiceRef` element as required. If you don't reference any services with the `<ServiceRef>` element, all toplevel services specified in the customer model data source will be used.
5. Replace the value of the `server` attribute with your own value. This is the logical *name* of the VantagePoint Navigator server as defined in `servers.xml`. See the *Configuring VantagePoint Navigator* for more information (`Configuring_VPNavigator.pdf`).

Replace the values of the `name` attribute with your own values. These are the service names as defined in VantagePoint Navigator. Note that the services listed here must also be configured in the customer model data source.

See also "OVSvcModules.dtd" on page 21 for more information about the correct XML syntax.

6. Save the XML file. In a browser, log in as the appropriate customer and ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP. The correct usage of this command is:
`xmlvalidate <xml_filename>.xml.`

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.

Make sure the following has been added to your PATH variable:

Windows NT/2000: %SIP_HOME%\bin

UNIX: /opt/OV/SIP/bin

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

Changing the Display Order of Service Cards

Two ways exist for changing the display order of services in the Service Cards module:

- Through the User Interface
- Through the Portal View File

However, it's much easier through the user interface than opening and manually editing the XML file.

Through the User Interface

1. On the title bar of the Service Cards module, click [Edit].
2. On the Service Cards - Edit page, select Choose from list.
3. Rearrange the order of the Selected services by selecting a service and clicking the [Up] or [Down] button.

Repeat this step until the Selected services list displays the services in the order you prefer.

4. To save and return to the main portal page, click [OK].

Through the Portal View File

Make a backup of XML files before you customize them. If you edit the XML file and get incorrect XML syntax, you may want the ability to

Presenting Service Cards

Configuring Service Cards

revert to the previous version of the file.

1. In an ASCII or XML editor, open the *PortalView.xml* configuration file.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

2. Search the string `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcCardServ let"` until you locate the Service Card(s) that you wish to move.
3. For the Service Card you wish to move, cut everything from `<ServiceRef name=` through to `</ServiceRef>` and paste it into the desired location between the `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcCardServ let"` and the `</ModuleInstance>` tags.
4. Repeat step 3 until the *PortalView.xml* file contains the Service Cards in the order in which you want to display them to this customer.
5. Save the XML file. In a browser, log in as the appropriate customer to ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP. The correct usage of this command is:

```
xmlvalidate <xml_filename>.xml.
```

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.

Make sure the following has been added to your PATH variable:

Windows NT/2000: `%SIP_HOME%\bin`

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

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

Removing Service Cards from the Service Cards Module

The following ways exist for removing Service Cards from the Service Cards Module:

- Through the User Interface
- Through the Portal View File

However, it's much easier through the user interface than opening and manually editing the XML file.

Through the User Interface

1. On the title bar of the Service Cards module, click [Edit].
2. On the Service Cards - Edit page, select Choose from list.
3. Select a service in the Selected services list and click [Remove].

Repeat this step until the Selected services list displays only the services you want.

4. To save and return to the main portal page, click [OK].

Modules can be removed through the user interface, either through the Options window or by clicking the Close [X] button in the module's title bar.

When using the Options window, the module is removed entirely from the `PortalView.xml` file. When using the Close [X] button, the display attribute in the `PortalView.xml` file is set to "no". If you want to redisplay this module, open the `PortalView.xml` file and change the display attribute to "yes".

Presenting Service Cards

Configuring Service Cards

Through the Portal View File

Make a backup of XML files before you customize them. 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* configuration file.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

2. Search the string `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcCardServlet"` until you locate the Service Card(s) that you wish to remove.
3. Do one of the following:
 - If you want to remove *all* Service Cards from the customer's configuration, do one of the following:
 - Change the `display` attribute in the `<ModuleInstance>` to `display="no"`. The `<ModuleInstance>` section is still present in the XML file.
 - Delete everything from `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcCardServlet"` through `</ModuleInstance>` to remove all Service Cards from the portal view file.
 - If you want to remove selected Service Cards from the customer's configuration, delete everything from `<ServiceRef name=` through to `</ServiceRef>`.
4. Save the XML file. In a browser, log in as the appropriate customer to ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML

parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP.

The correct usage of this command is:

```
xmlvalidate <xml_filename>.xml.
```

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.

Make sure the following has been added to your PATH variable:

Windows NT/2000: %SIP_HOME%\bin

UNIX: /opt/OV/SIP/bin

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

Presenting Service Cards
Configuring Service Cards

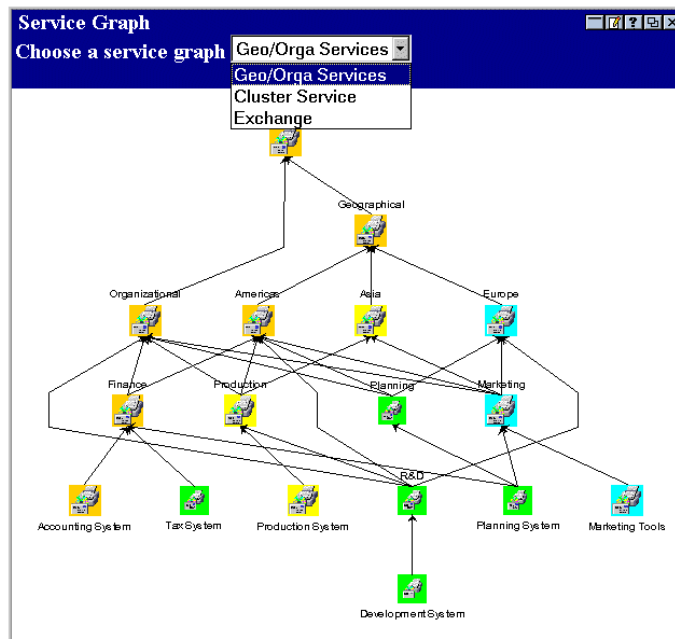
4 Presenting Services in the Service Graph

Understanding the Service Graph

The Service Graph module displays the services in a graphical view. Depending on how many services are configured for the Service Graph module, you have the choice between several Service Graphs. To display a Service Graph, choose a Service Graph from the list box at the top of the Service Graph module.

See Figure 4-1 on page 54 for an example of the Service Graph module.

Figure 4-1 The Service Graph Module



Configuring the Service Graph

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

The first step in configuring the Service Graph module is to ensure that VantagePoint Navigator has been configured to work with Service Information Portal. See the *Configuring VantagePoint Navigator* guide for more information ([Configuring_VPNavigator.pdf](#)). After that, you can add a Service Graph to be displayed in a portal view.

- Adding a Service Graph to a Portal View
- Removing a Service Graph from a Portal View

Adding a Service Graph to a Portal View

Two ways exist for adding a Service Graph to be displayed in a portal view.

- Through the User Interface
- Through the Portal View File

However, it's much easier through the user interface than opening and manually editing the XML file.

Through the User Interface

1. Login as the customer whose portal you want to change.
2. Scroll to the bottom of the page. You'll notice drop-down list boxes with [Add] buttons next to them.
3. Select `Service Graph` from the list and click [Add].

A Service Graph module is added to the portal page. The new module automatically displays the services of the VantagePoint Navigator server listed first in the `servers.xml` file. Further configuration is necessary.

4. On the title bar of the Service Graph module, click [Edit].
5. On the `Service Graph - Edit` page, select a VantagePoint Navigator management server from the drop-down list box.

Presenting Services in the Service Graph

Configuring the Service Graph

6. Choose which services you wish to display as Service Graphs in the Service Graphs module:
 - **Toplevel only**

Select `All toplevel` from `customer model`, if you want to display the toplevel services specified in the customer model data source.
 - **Selected services**

Select `Choose` from `list` and select a service from the `All services list`. Click `[Add]`.

Repeat this step until the `Selected services list` contains all the services you want to display as Service Graphs.
7. To save and return to the main portal page, click `[OK]`.

TIP

The default size of the resulting Service Graph can be changed by directly editing the portal view file. See the *Configuring VantagePoint Navigator* guide for more information (`Configuring_VPNavigator.pdf`).

Through the Portal View File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

1. If a portal view file already exists, open it with an ASCII or XML editor and proceed to Step 2.

If a portal view file does not yet exist, see the *SIP 2.0 Administrator Guide* (`Administrator_Guide.pdf`) and follow the procedure for creating portal views.

2. Create a new module instance and copy the following attribute into the `ModuleInstance` element:

```
classid="com.hp.ov.portal.modules.service.SvcCardServlet"
```


The `ModuleInstance` `id` must be unique among all module instances in the portal view file. For information on the `ModuleInstance` element, see the *SIP 2.0 Administrator Guide* (`Administrator_Guide.pdf`).

3. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags.

```
<ServiceGraph server="bbn_server1">
  <GraphRef name="email" depth="3"
    width="600" height="500" />
</ServiceGraph>
```

4. If you want to include more Service Graphs, copy and paste the `GraphRef` element as required. If you don't reference any services with the `<GraphRef>` element, all toplevel services specified in the customer model data source will be used.
5. Replace the value of the `server` attribute with your own value. This is the logical *name* of the VantagePoint Navigator server as defined in `servers.xml`. See the *Configuring VantagePoint Navigator* for more information (`Configuring_VPNavigator.pdf`).
6. Replace all other values with your own values.

The services listed under the `name` attribute must also be configured in the customer model data source. The `depth`, `width`, and `height` attributes can only be set in the portal view file and not via the portal user interface.

See also "OVSvcModules.dtd" on page 21 for more information about the correct XML syntax.

7. Save the XML file. In a browser, log in as the appropriate customer and ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP. The correct usage of this command is:
`xmlvalidate <xml_filename>.xml.`

Presenting Services in the Service Graph

Configuring the Service Graph

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.

Make sure the following has been added to your PATH variable:

Windows NT/2000: %SIP_HOME%\bin

UNIX: /opt/OV/SIP/bin

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

Removing a Service Graph from a Portal View

The following ways exist for removing Service Graphs from a portal view:

- Through the User Interface
- Through the Portal View File

However, it's much easier through the user interface than opening and manually editing the XML file.

Through the User Interface

1. On the title bar of the Service Graph module, click [Edit].
2. On the Service Graph - Edit page, select Choose from list.
3. Select a service in the Selected services list and click [Remove].

Repeat this step until the Selected services list displays only the services you want.

4. To save and return to the main portal page, click [OK].

Modules can be removed through the user interface, either through the Options window or by clicking the Close [X] button in the module's title bar.

When using the Options window, the module is removed entirely from the *PortalView.xml* file. When using the Close [X] button, the display attribute in the *PortalView.xml* file is set to "no". If you want to redisplay this module, open the *PortalView.xml* file and change the display attribute to "yes".

Through the Portal View File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: <install_dir>\SIP\conf\share\views

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

1. In an ASCII or XML editor, open the *PortalView.xml* configuration file.

Presenting Services in the Service Graph

Configuring the Service Graph

2. Search the string `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcGraphServlet">` until you locate the Service Graph that you wish to remove.
3. Do one of the following:
 - Change the display attribute in the `<ModuleInstance>` to `display="no"`. The `<ModuleInstance>` section is still present in the XML file.
 - Delete everything from `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcGraphServlet">` through `</ModuleInstance>` to remove the selected Service Graph from the portal view file.
4. Save the XML file. In a browser, log in as the appropriate customer to ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP. The correct usage of this command is:

```
xmlvalidate <xml_filename>.xml.
```

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.

Make sure the following has been added to your PATH variable:

Windows NT/2000: %SIP_HOME%\bin

UNIX: /opt/OV/SIP/bin

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

5**Presenting Service Health Data**

Understanding Service Health Gauges

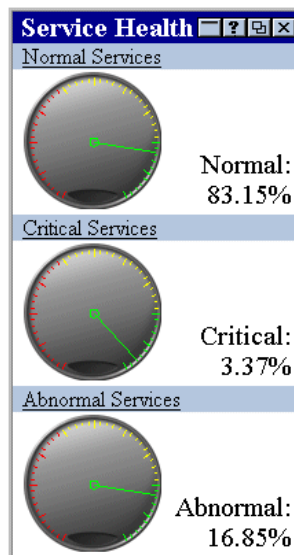
Service Health Gauges show the availability of all or selected services in percent. Service health is scored as a value from 0-100.

Two types of views are available for Service Health: Gauges and Details. Gauges are displayed when the tab first displays in your portal, and indicate the overall health rating for selected or all services that are being monitored by that particular gauge. Details are viewed by clicking on a gauge or a health title link.

Figure 5-1 on page 62 shows an example of the Service Health module. Depending on how you have configured the module, you may see one or more gauges, and the gauges may present different data. See “Calculating Service Health” on page 64 for more information about how to set up Service Health Gauges.

Figure 5-1

The Service Health Gauges



Gauge View

Service Health Gauges show health metrics that you can define and configure. In Figure 5-1 on page 62, the first gauge shows the percentage of all services that have the severity normal. The value is 90% which means that your service environment is very healthy.

The colors on the gauge indicate the health of your managed services: the red area on the left shows percent values between 0 and 40, the yellow area in the middle shows values between 40 and 80, and the green area on the right shows values between 80 and 100.

Details View

All health gauges provide drill-down capability for two levels of health detail. To get to the first level of health details, click on the gauge or the health title link above the gauge. A second level of detail can be reached from the first level page by clicking on the severity link in the *Severity* column.

The first column of a detail page—*Severity*—displays a list of all severities.

The second column—*numbers*—contains the raw data values, that is the number of services of each severity.

The third column—*percentage*—displays the health score for each severity. The score is a percent value.

Below that, total numbers are given for the services that have contributed to the calculation and those that have not.

The second level of health details shows a *Service Browser* which lists only those services that have contributed with their severity to the health calculation. For *Service Health Gauges*, the *Service Browser* shows a flat list of services that have contributed to the calculation with their severity. See also *Understanding the Service Browser*.

Configuring Service Health Gauges

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

The first step in configuring the Service Health Gauges is to ensure that VantagePoint Navigator has been configured to work with Service Information Portal. See the *Configuring VantagePoint Navigator* guide for more information ([Configuring_VPNavigator.pdf](#)). After that, you can add Service Health Gauges to be displayed in a portal view.

- Adding Service Health Gauges to a Portal View
- Changing the Display Order of Service Health Gauges
- Removing Service Health Gauges from a Portal View

Calculating Service Health

The first step in calculating service health is to decide which severities should go into the service health calculation. For example, you may decide that you want to show the percentage of all critical and major services compared to all services in the customer model data source. In this case you would select the critical and major services when configuring a Service Health Gauge. A resulting value of, for example, 5% indicates that your service environment is healthy because only 5% of all services are in the state critical or major.

However, the health gauges are designed in such a way that 5% would be indicated with a pointer pointing into the red area of the health gauge because the red area shows percent values between 0 and 40. To show that your environment is healthy and within the green area of the health gauge, you could invert the gauge (using the XML attribute `inverse`). The pointer would then point to 95% (within the green area of the gauge) but the score would still be displayed as 5%.

Alternatively, instead of calculating the percentage of all critical and major services, you could calculate the percentage of all minor, warning, and normal services.

TIP

The total number of all services is configurable. You could either use all toplevel services defined in the customer model data source, or you could

specifically select services (using the XML element `ServiceRef`).

In addition, you can use the XML attribute `recursive` to include either all services from the customer model data source (not only the toplevel services), or, if having selected specific services, all subservices of the selected services.

Adding Service Health Gauges to a Portal View

Through the Portal View File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

1. If a portal view file already exists, open it with an ASCII or XML editor and proceed to Step 2.

If a portal view file does not yet exist, see the *SIP 2.0 Administrator Guide* (`Administrator_Guide.pdf`) and follow the procedure for creating portal views.

2. Create a new module instance and copy the following attribute into the `ModuleInstance` element:

```
classid="com.hp.ov.portal.modules.service.SvcHealthServlet"
```

The `ModuleInstance id` must be unique among all module instances in the portal view file. For information on the `ModuleInstance` element, see the *SIP 2.0 Administrator Guide* (`Administrator_Guide.pdf`).

3. Open the `OVDDefaultSvcHealth.xml` file, and copy everything from `<ServiceHealth>` through to `</ServiceHealth>` between the `ModuleInstance` starting and closing tags.

`OVDDefaultSvcHealth.xml` is available in the following directory:

Windows NT/2000: `<install_dir>\SIP\registration\defaults`

UNIX: `/etc/opt/OV/SIP/registration/defaults`

Configuring Service Health Gauges

4. If you want to include more Service Health Gauges, copy and paste the `ServiceSummary` element as required.
5. Replace the value of the `server` attribute with your own value. This is the logical name of the VantagePoint Navigator server as defined in `servers.xml`. See the *Configuring VantagePoint Navigator* guide for more information (`Configuring_VPNavigator.pdf`).

Replace the values of the other attributes and elements with your own values. See “Calculating Service Health” on page 64 for a detailed explanation of how to set up Service Health Gauges.

See also “OVSvcModules.dtd” on page 21 for more information about the correct XML syntax.

6. Save the XML file. In a browser, log in as the appropriate customer and ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

Changing the Display Order of Service Health Gauges

Through the Portal View File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

1. In an ASCII or XML editor, open the `PortalView.xml` configuration file.
2. Search the string `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcHealthService t"` until you locate the Service Health Gauge(s) that you wish to move.
3. For the Service Health Gauge you wish to move, cut everything from `<ServiceSummary>` through to `</ServiceSummary>` and paste it into the desired location between the `<ServiceHealth>` and the `</ServiceHealth>` tags.

4. Repeat step 3 until the *PortalView.xml* file contains the Service Health Gauges in the order in which you want to display them to this customer.
5. Save the XML file. In a browser, log in as the appropriate customer to ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

Removing Service Health Gauges from a Portal View

Through the Portal View File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

1. In an ASCII or XML editor, open the *PortalView.xml* configuration file.
2. Search the string `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcHealthServlet"` until you locate the Service Health Gauge(s) that you wish to remove.
3. Do one of the following:
 - If you want to remove *all* Service Health Gauges from the customer's configuration, do one of the following:
 - Change the `display` attribute in the `<ModuleInstance>` to `display="no"`. The `<ModuleInstance>` section is still present in the XML file.
 - Delete everything from `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcHealthServlet"` through `</ModuleInstance>` to remove all Service Health Gauges from the portal view file.

If you want to remove *selected* Service Health Gauges from the

Configuring Service Health Gauges

customer's configuration, delete everything from
<ServiceSummary> through to </ServiceSummary>.

4. Save the XML file. In a browser, log in as the appropriate customer to ensure that the desired behavior has been established.

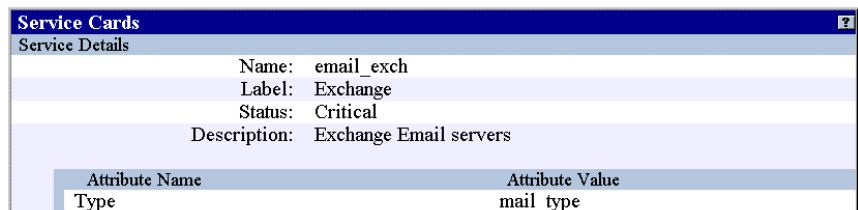
Changes to portal view files take effect when you display or refresh the portal view.

Understanding Service Details

The Service Details drill-down view provides detailed information about a service, for example, any rules or actions associated with a service. For more information about each detail, see the *HP OpenView VantagePoint Navigator Concepts and Configuration Guide*.

Individual user access to service details can be configured in the *PortalView.xml* file. By default, user access to service details is disabled. See Figure 6-1 on page 70 for an example of how Service Details may look like.

Figure 6-1 Details of a Service



The screenshot shows a window titled "Service Cards" with a "Service Details" tab. The details are as follows:

Name:	email_exch
Label:	Exchange
Status:	Critical
Description:	Exchange Email servers

Attribute Name	Attribute Value
Type	mail_type

If configured, you can access Service Details in either of the following ways:

- **Via the Service Browser**

Click on the status icon in the first column of the Service Browser.

See “Understanding the Service Browser” on page 32 for more information about the Service Browser.

- **Via Service Cards**

Click on the service icon in the Service Cards module.

See “Understanding Service Cards” on page 42 for more information about Service Cards.

- **Via Service Health Gauges**

Click on the underlined name of a Service Health Gauge, and then on the status icon in the first column of the resulting Service Health Details page. A Service Browser opens which lets you access service details by clicking on the status icon in the first column.

See “Understanding Service Health Gauges” on page 62 for more information about Service Health Gauges.

Controlling Access to Service Details

By default access to service details is disabled for all customers. You can change the setting in the *PortalView.xml* file for each of the following modules:

- Service Browser
- Service Cards
- Service Health Gauges

Service Details contain very detailed information about a service and may therefore not be immediately understood by your customers. This is why access to Service Details has been disabled by default. If you want to allow your customers to access Service Details, you should also provide explanatory text for each detail. More information about what each detail means is available in the *HP OpenView VantagePoint Navigator Concepts and Configuration Guide*.

Enabling/Disabling Access to Service Details

The following ways exist for enabling/disabling access to Service Details:

- Through the User Interface
- Through the Portal View File

However, it's much easier through the user interface than opening and manually editing the XML file.

Through the User Interface

1. On the title bar of the module for which you want to enable/disable access to Service Details, click [Edit].
2. On the edit page, check the Allow details view drill-down check box to enable access to Service Details. To disable the access, uncheck the box.
3. To save and return to the main portal page, click [OK].

Through the Portal View File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: <install_dir>\SIP\conf\share\views

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

1. If a portal view file already exists, open it with an ASCII or XML editor and proceed to Step 2.

If a portal view file does not yet exist, see the *SIP 2.0 Administrator Guide* (*Administrator_Guide.pdf*) and follow the procedure for creating portal views.

2. Depending on the module for which you want to disable access to service details, search the following string:

Service Browser: <ServiceBrowser

Service Cards: <ServiceCard

Service Health Gauges: <ServiceHealth

3. To *disable* access to service details, change the details attribute in the starting tag of the module to details="no".

To *enable* access to service details, change the details attribute in the starting tag of the module to details="yes".

4. Save the XML file. In a browser, log in as the appropriate customer to ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP. The correct usage of this command is:

```
xmlvalidate <xml_filename>.xml.
```

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.

Presenting Service Details
Controlling Access to Service Details

Make sure the following has been added to your PATH variable:

Windows NT/2000: %SIP_HOME%\bin

UNIX: /opt/OV/SIP/bin

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

Understanding Service Reports

Service reports contain historical, aggregated data, diagrams and detailed tables about the status, availability, and performance of managed services. They are generated by HP OpenView VantagePoint Reporter, the HP OpenView management reporting tool. The type and content of the available reports depends on what has been configured for VantagePoint Navigator or VantagePoint Operations.

Reports can be viewed by clicking on the underlined report name. The result (usually an HTML page) is displayed in a new browser window.

Note that reports are not updated dynamically. The date and time when a report is generated depends on the configuration of HP OpenView VantagePoint Reporter.

See Figure 7-1 on page 76 for an example of how a Bookmarks module can be used to display Service Reports.

Figure 7-1 **A Bookmarks Module Used for Service Reports**



Configuring the Service Reports

Configuration is done in two steps:

1. Configure VantagePoint Reporter. See “Setting up HP OpenView VantagePoint Reporter” on page 77.
2. Add a Service Reports module to Service Information Portal. See “Adding Service Reports to a Portal View” on page 77.

Setting up HP OpenView VantagePoint Reporter

For Service Information Portal to be able to access the service reports via HTTP, you must tell VantagePoint Reporter to generate the reports into a web server directory. This can be any URL.

In addition, you may want to configure a generation interval or other settings. See the VantagePoint Reporter documentation for more information.

Adding Service Reports to a Portal View

You configure Service Reports by adding a new Bookmarks module to the portal view. See the *SIP 2.0 Administrator Guide* ([Administrator_Guide.pdf](#)) for more information about setting up the Bookmarks module.

TIP

Use the `title` attribute to choose an appropriate name for the Service Reports module.

You must be careful when setting up a Bookmarks module for Service Reports: ensure that your customers can access only those reports that are intended for them. Otherwise a situation may occur where a customer can generate a report about another customer’s services.

If customer access to reports is not an issue, you can also use the integration of VantagePoint Reporter into Service Information Portal. This integration is shipped with Service Information Portal but must be configured before it can be used.

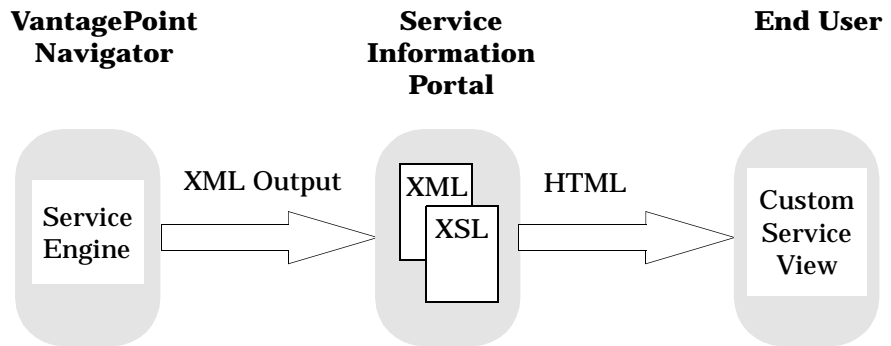
Presenting Service Reports
Configuring the Service Reports

8 Presenting Custom Service Views

Understanding Custom Service Views

Custom Service Views enable you to present service data using your own format, layout, and style without having to program your own module. All you need to do is develop a stylesheet that formats the XML output of the VantagePoint Navigator service engine. See Figure 8-1 on page 80 for an overview of the process.

Figure 8-1 **Displaying XML Data Through Customized Service Views**



The XML output produced by the VantagePoint Navigator service engine is based on the `OVSERVICEDOC.dtd` which is a system entity of `OVSVCUSTOMDOC.dtd`. Your stylesheet must be developed according to the rules of `OVSVCUSTOMDOC.dtd`.

In addition to the stylesheet, you should also write online help about your custom service view. Service Information Portal ships a default HTML text that is displayed to your customers when they click the [?] button in the title bar of your Custom Service View module. You should replace this text with your own, customized version. See “Writing Your Own Online Help” on page 87 for more information.

Understanding Depth and Level in ServiceQuery

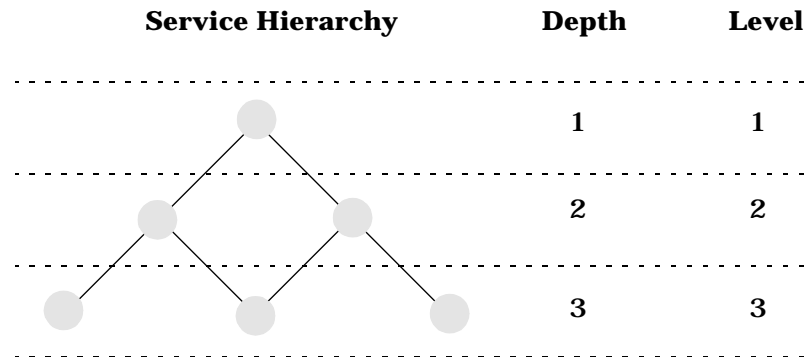
Part of the configuration task of adding a custom service view to be displayed to a customer, is the definition of the `depth` and `level` attributes of `ServiceQuery`. You should be aware of the following implications: `depth` describes the level in the service hierarchy that is visible to the customer, `level` specifies a specific level from which services will be visible to the customer. `depth="0"` stands for recursive and means that all levels are visible.

For example, if `depth="3"` and `level="2"`, only levels 2 and 3 are visible. Level 1 is not visible. See also Figure 8-2 on page 81

NOTE

The services specified by `depth` and `level` must also be defined in the customer model data source. Otherwise your Custom Service View will query a service that is not included in the customer model data source and will return empty XML data. See also “Configuring the Customer Model Data Source” on page `HIDDEN`.

Figure 8-2 Deciding the Value for Depth



Depth 0 = recursive = all levels including toplevel

Custom Service View Examples

Service Information Portal ships some example files that help you get started with the Custom Service View module. The example files are located in the following directory:

Window NT/2000: `<install_dir>\SIP\samples\service\`

UNIX: `/opt/OV/SIP/samples/service/`

The following files are available:

`README.txt` Readme file explaining how to use the examples.

`exchmail.xml` VantagePoint Navigator configuration file.

`exchmail_html.xsl` XSL stylesheet.

This is just one example of how the Custom Service View module could be used. Other possible use scenarios include a grouping of the services by geography, grouping by function, or visualizations with the use of images.

Configuring a Custom Service View

Configuration is done in four steps:

1. Ensure that VantagePoint Navigator has been configured to work with Service Information Portal. See the *Configuring VantagePoint Navigator* guide for more information (*Configuring_VPNavigator.pdf*).
2. Configure the Custom Service View module. See “Adding Custom Service Views to a Portal View” on page 83.
3. Develop an XSL stylesheet. See “Developing Your Own Stylesheet” on page 86.
4. Write the online help. See “Writing Your Own Online Help” on page 87.

Adding Custom Service Views to a Portal View

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

1. If a portal view file (*PortalView.xml*) already exists, open it with an ASCII or XML editor and proceed to Step 2.

If a portal view file does not yet exist, see the *SIP 2.0 Administrator Guide* (*Administrator_Guide.pdf*) and follow the procedure for creating portal views.

2. Create a new module instance and copy the following attribute into the `ModuleInstance` element:

```
classid="com.hp.ov.portal.modules.service.SvcCustomServlet"
```

The `ModuleInstance` `id` must be unique among all module instances in the portal view file. For information on the `ModuleInstance` element, see the *SIP 2.0 Administrator Guide*

Presenting Custom Service Views

Configuring a Custom Service View

(Administrator_Guide.pdf).

3. Copy the following text into the XML file, between the `ModuleInstance` starting and closing tags.

```
<ServiceCustom name="svcbrowser" server="bbn_server1">  
  <ServiceQuery name="true" depth="0" level="1" />  
  <ServiceQuery name="email" depth="0" level="1" />  
</ServiceCustom>
```

4. Replace the value of the attribute with your own values:

<code>name</code>	The name you enter here references the XSL stylesheet. See also “Developing Your Own Stylesheet” on page 86.
<code>server</code>	This is the logical name of the VantagePoint Navigator server as defined in <code>servers.xml</code> . See the <i>Configuring VantagePoint Navigator</i> for more information (Configuring_VPNavigator.pdf).
<code>name</code>	These are the service names as defined in VantagePoint Navigator.
<code>depth</code>	This value specifies the number of levels in the service hierarchy that are displayed to the customer.
<code>level</code>	This value specifies a specific level in the service hierarchy that will be displayed.

See also “OVSvcModules.dtd” on page 21 for more information about the correct XML syntax.

5. Save the XML file. In a browser, log in as the appropriate customer and ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP. The correct usage of this command is:
`xmlvalidate <xml_filename>.xml.`

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.

Make sure the following has been added to your PATH variable:

Windows NT/2000: %SIP_HOME%\bin

UNIX: /opt/OV/SIP/bin

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

Developing Your Own Stylesheet

Your stylesheet must use XSL (eXtensible Stylesheet Language) and must transform the XML data into HTML data. It must be based on the `OVSvcCustomDoc.dtd`. See “`OVSvcCustomDoc.dtd`” on page 91 for more information about the XML syntax.

After you have developed it, save the 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` tag in the `PortalView.xml` file.

TIP

It is important that the names correspond because this is how Service Information Portal associates your stylesheet with your Custom Service View.

Stylesheets are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\styles`

UNIX: `/etc/opt/OV/SIP/conf/styles`

NOTE

The HTML output of your stylesheet must only contain HTML tags that are valid between the starting and closing tags of `<body>`. Service Information Portal automatically provides the framing `<html>`, `<head>`, and `<body>` tags.

Writing Your Own Online Help

Service Information Portal ships a default HTML file that is displayed when your customer clicks the [?] button in the title bar of your Custom Service View module. You should replace this default file with your own help information to assist your customers in using the module.

The help file is in HTML format and should only use HTML syntax that is supported by all web browsers. Your file can be located anywhere in the `htdocs/C/help` directory of Service Information Portal. However, it is recommended that you store it in the following directory:

Windows NT/2000: `<install_dir>\SIP\htdocs\C\help\VPNavigator`

UNIX: `/opt/OV/SIP\htdocs/C/help/VPNavigator`

To Associate the [?] Button With Your Own Help File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: `<install_dir>\SIP\conf\share\views`

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

1. In an ASCII or XML editor, open the `PortalView.xml` configuration file.
2. Search the string `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcCustomServlet"` until you locate the Custom Service View with which you want to associate the help file.
3. Insert the attribute `help` into the `<ModuleInstance>`:

```
help="/OVSipDocs/C/help/VPNavigator/topic.html"
```

Replace `topic.html` with the name of your help file. The attribute `help` allows you to override the default help URL defined in the module registration file. See the *SIP 2.0 Administrator Guide* (`Administrator_Guide.pdf`) for more information about the `ModuleInstance` element.

4. Save the XML file. In a browser, log in as the appropriate customer to ensure that the desired behavior has been established.

Presenting Custom Service Views

Configuring a Custom Service View

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP. The correct usage of this command is:

```
xmlvalidate <xml_filename>.xml.
```

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.

Make sure the following has been added to your PATH variable:

Windows NT/2000: %SIP_HOME%\bin

UNIX: /opt/OV/SIP/bin

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

Removing a Custom Service View from a Portal View

The following ways exist for removing Custom Service Views from a portal view:

- Through the User Interface
- Through the Portal View File

However, it's much easier through the user interface than opening and manually editing the XML file.

Through the User Interface

Modules can be removed through the user interface, either through the Options window or by clicking the Close [X] button in the module's title bar.

When using the Options window, the module is removed entirely from the *PortalView.xml* file. When using the Close [X] button, the display attribute in the *PortalView.xml* file is set to "no". If you want to redisplay this module, open the *PortalView.xml* file and change the display attribute to "yes".

Through the Portal View File

Make a backup of XML files before you customize them. 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.

Portal view files are stored in the following directory:

Windows NT/2000: <install_dir>\SIP\conf\share\views

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

1. In an ASCII or XML editor, open the *PortalView.xml* configuration file.
2. Search the string <ModuleInstance classid="com.hp.ov.portal.modules.service.SvcCustomServlet" until you locate the Custom Service View that you wish to remove.
3. Do one of the following:
 - Change the display attribute in the <ModuleInstance> to display="no". The <ModuleInstance> section is still present

Presenting Custom Service Views

Configuring a Custom Service View

in the XML file.

- Delete everything from `<ModuleInstance classid="com.hp.ov.portal.modules.service.SvcCustomServlet" >` through `</ModuleInstance>` to remove the selected Custom Service View from the portal view file.
4. Save the XML file. In a browser, log in as the appropriate customer to ensure that the desired behavior has been established.

Changes to portal view files take effect when you display or refresh the portal view.

NOTE

After you make modifications to XML files, validate the syntax. Provided with SIP is the command `xmlvalidate`, which checks whether the XML file is both well-formed and valid. `xmlvalidate` uses the same XML parser as SIP, so if the file passes `xmlvalidate`, it will work with SIP. The correct usage of this command is:

```
xmlvalidate <xml_filename>.xml.
```

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.

Make sure the following has been added to your PATH variable:

Windows NT/2000: %SIP_HOME%\bin

UNIX: /opt/OV/SIP/bin

If the output of the `xmlvalidate` command indicates a problem but does not fully describe it, you can open the XML file in Internet Explorer which sometimes provides a clearer description of the problem.

As an alternative, you can find a commercial XML validation tool for Windows NT/2000 at www.xmlspy.com.

OVSvcCustomDoc.dtd

This section contains a detailed description of the XML syntax of the OVSvcCustomDoc.dtd. **Note that the OVSvcCustomDoc.dtd contains the system entity OVServiceDoc.dtd.** See the *HP OpenView VantagePoint Navigator Concepts and Configuration Guide* for more information about this DTD (service.dtd).

The OVSvcCustomDoc.dtd is located in the following directory:

Windows NT/2000:

```
<install_dir>\SIP\conf\share\modules\service
```

UNIX: /etc/opt/OV/SIP/conf/share/modules/service

The DTD shows the proper grammar for the service module configuration:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- OVSvcCustomDoc.dtd -->
<!-- Copyright (c) 2001 Hewlett-Packard Company -->
<!ENTITY % ServicesDTD SYSTEM "OVServiceDoc.dtd">
%ServicesDTD;

<!ELEMENT ServiceDocument (Header, Services)>
<!ELEMENT Header (Sort?, ModuleName?, ParamList, ParamGraph,
ParamSearch, GraphImage, DrillDown, Details,
ServiceListSelected?, ServiceList*)>
<!ELEMENT ModuleName EMPTY>
<!ATTLIST ModuleName
name CDATA #REQUIRED
>
<!ELEMENT Sort EMPTY>
<!ATTLIST Sort
type CDATA #REQUIRED
>
```

Presenting Custom Service Views

Configuring a Custom Service View

```
<!ELEMENT ParamList EMPTY>
<!ATTLIST ParamList
service CDATA #REQUIRED
>
<!ELEMENT ParamGraph EMPTY>
<!ATTLIST ParamGraph
service CDATA #REQUIRED
depth CDATA #REQUIRED
level CDATA #REQUIRED
>
<!ELEMENT ParamSearch EMPTY>
<!ATTLIST ParamSearch
search (status | description | label | unsorted) #REQUIRED
>
<!ELEMENT GraphImage EMPTY>
<!ATTLIST GraphImage
imageid CDATA #REQUIRED
>
<!ELEMENT ServiceListSelected EMPTY>
<!ATTLIST ServiceListSelected
name CDATA #REQUIRED
label CDATA #REQUIRED
>
<!ELEMENT ServiceList EMPTY>
<!ATTLIST ServiceList
name CDATA #REQUIRED
label CDATA #REQUIRED
>
```

<!ELEMENT DrillDown EMPTY>

<!ELEMENT Details EMPTY>

Table 8-1 on page 93 lists the elements and attributes of the OVSvcCustomDoc.dtd.

Table 8-1 OVSvcCustomDoc.dtd Elements and Attributes

Elements	Attributes	Description
ServiceDocument		Required.
Header		Required.
Sort	type	For internal use only.
ModuleName	name	For internal use only.
ParamList	service	For internal use only.
ParamGraph	service	For internal use only.
	depth	For internal use only.
	level	For internal use only.
ParamSearch	search	For internal use only.
GraphImage	imageid	For internal use only.
DrillDown		For internal use only.
Details		For internal use only.
ServiceListSelected	name	For internal use only.
	label	For internal use only.
ServiceList		For internal use only.
Services		Required. See the <i>HP OpenView VantagePoint Navigator Concepts and Configuration Guide</i> for more information.

Presenting Custom Service Views
Configuring a Custom Service View

A Service Status

Service Status

Table A-1 on page 96 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.

A

adding

- custom service views, 83
- service browser, 35
- service cards, 44
- service graph, 55
- service health gauges, 65
- service reports, 77

C

calculating

- service health, 64

changing order

- of service cards, 47
- of service health gauges, 66

configuration files

- understanding, 20

configuring

- custom service views, 83
- service browser, 35
- service cards, 44
- service graph, 55
- service health gauges, 64
- service reports, 77

conventions

- typographical, 13

custom service views

- adding, 83
- configuring, 83
- developing stylesheet, 86
- examples, 82
- presenting, 79
- removing, 89
- understanding, 80
- writing online help, 87

D

details

- See service details.

developing stylesheet

- custom service views, 86

disabling

- access to service details, 72

display order

- changing for service cards, 47

display order gauges

- changing for service health, 66

DTDs

- OVSServiceDoc.dtd, 91

OVSvcCustomDoc.dtd, 91

OVSvcModules.dtd, 21

VantagePoint Navigator, 20

E

editing

- XML files, 28

enabling

- access to service details, 72

examples

- custom service views, 82

O

online help

- for custom service views, 87

OVSServiceDoc.dtd, 91

OVSvcCustomDoc.dtd, 91

OVSvcModules.dtd, 21

P

presenting

- custom service views, 79
- service browser, 31
- service cards, 41
- service details, 69
- service graph, 53
- service health gauges, 61
- service search, 75
- VantagePoint Navigator data in SIP 2.0, 17

R

removing

- custom service views, 89
- service browser, 38
- service cards, 49
- service graph, 59
- service health gauges, 67

S

searching

- services in service browser, 33

service browser

- adding, 35
- configuring, 35
- presenting, 31
- removing, 38
- searching services, 33
- sorting services, 33

Index

- understanding, 32
- viewing services, 32
- service cards
 - adding, 44
 - changing display order, 47
 - configuring, 44
 - presenting, 41
 - removing, 49
 - understanding, 42
- service details
 - disabling access, 72
 - enabling access, 72
 - presenting, 69
 - understanding, 70
- service graph
 - adding, 55
 - configuring, 55
 - presenting, 53
 - removing, 59
 - understanding, 54
- service health
 - calculating, 64
- service health gauges
 - adding, 65
 - changing display order, 66
 - configuring, 64
 - presenting, 61
 - removing, 67
 - understanding, 62
- Service Information Portal and VantagePoint Navigator, 18
- service reports
 - adding, 77
 - configuring, 77
 - setting up VantagePoint Reporter, 77
 - understanding, 76
- service search
 - presenting, 75
- service status, 96
- setting up
 - VantagePoint Reporter, 77
- severity, 96
- sorting
 - services in service browser, 33
- status, 96
- stylesheets
 - for custom service views, 86

U

- understanding

- configuration files, 20
- custom service views, 80
- service browser, 32
- service cards, 42
- service details, 70
- service graph, 54
- service health gauges, 62
- service reports, 76

V

- VantagePoint Navigator and Service Information Portal, 18
 - DTDs, 20
 - modules available in SIP 2.0, 19
 - presenting data in SIP 2.0, 17
- VantagePoint Reporter
 - setting up for service reports, 77
- viewing
 - services in service browser, 32

W

- writing online help
 - custom service views, 87

X

- XML
 - backing up, 28
 - editing files, 28
 - rules for editing, 28
 - validating, 28
- XSL
 - See stylesheets.